CITY UNIVERSITY LONDON

School of Mathematics, Computer Science and Engineering

Autonomous Quality of Service Management and Policing in Unmanaged Local Area Networks

PhD Thesis

Supervisor :	Dr. Veselin Rakočević
Second Supervisor :	Professor Muttukrishnan Rajarajan
Local Supervisor:	Prof. Dr. rer. nat. Rudolf Jäger

by

Dipl.-Ing. (FH) Christopher Sebastian Köhnen

April, 2016

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACE	ADAPTIVE Communication Environment
ACK	ACKnowledgement
AIT	Address Information Table
API	Application Programming Interface
ARP	Address Resolution Protocol
BSSID	Basic Service Set IDentification
CBMPAR	Class-Based Mean Prediction Accuracy Ratio
CBPHR	Class-Based Prediction Hit Rate
CBQoS	Class-Based QoS range
CBR	Constant Bit Rate
CB	Class-Based
CDN	Content Delivery Network
CDP	Cisco's Cisco Discovery Protocol
CLMPAR	ClassLess Mean Prediction Accuracy Ratio
CLPHR	ClassLess Prediction Hit Rate

CLQoS	ClassLess QoS Range
CL	ClassLess
CoS	Class of Service
CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DASH	Dynamic Adaptive Streaming over HTTP
DD-WRT	DresDen-WirelessRouTer
DFA	Deterministic Finite Automata
DFN	Deutsches Forschungs-Netzwerk (German Science Network)
DF	Delay Factor
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DIFFUSE	Distributed Firewall and Flow-shaper Using Statistical Evidence
DNS	Dynamic Name System
DPI	Deep Packet Inspection
DSCP	Differentiated Services Code Point
ESI	Error Source Identifier
eSPID	Enhanced Statistical Protocol IDentification
ETSI	European Telecommunications Standards Institute

FB	Formula Based
FE	Functional Entity
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
FTT	Flexible Time-Triggered
GIST	General Internet Signalling Transport
GMPLS	Generalized Multi-Protocol Label Switching
GRM	Global Resource Manager
HB	History Based
HLS	HTTP Live Streaming
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICCE	International Conference on Consumer Electronics
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFG	Inter Frame Gap
IntServ	Integrated Services
IPTV	Internet Protocol TeleVision

IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IP	Internet Protocol
IRC	Internet Relay Chat
ITU-T	ITU - Technical Work Group
ITU	International Telecommunications Union
JCNC	Journal of Computer Networks and Communications
KLD	Kullback-Leibler Divergence
LAN	Local Area Network
LI	Local Indication
LLDP-DU	LLDP Data Units
LLDP-MED	LLDP Media Endpoint Discovery
LLDP	Link Layer Discovery Protocol
LLTD	Link Layer Topology Discovery
LRM	Local Resource Manager
LSP	Label Switched Paths
MAC	Media Access Control
MA	Moving Average
MBAC	Measurement Based Admission Control
MDI	Media Delivery Index

mDNS	Multicast Dynamic Naming System
ME	Mean Estimation
MIB	Management Information Base
MLR	Media Loss Rate
MMEDIA	Second International Conference on Advances in Multimedia
MMS	Microsoft Media Server Protocol
MOS	Mean Opinion Score
MPAR	Mean Prediction Accuracy Ratio
MPEG	Moving Picture Experts Group
MPHR	Mean Prediction Hit Rate
MPLS	Multiprotocol Label Switching
MRM	Message Routing Method
MR	Media Rate
MSAP	Media Service Access Point
MTU	Maximum Transmission Unit
NAS	Network Attached Storage
NAT	Network Address Translation
NMS	Network Management System
NSIS QoS-NSLP	NSIS Signaling Layer Protocol for Quality-of-Service Signaling

NSIS	Next Steps In Signaling
NSLP	NSIS Signaling Layer Protocol
NTLP	NSIS Transport Layer Protocol
ODP	On-Demand QoS Path framework
OER	Over-Estimation Rate
OSGi	Open Services Gateway initiative
OS	Operating System
P.NAMS	Parametric Non-intrusive Assessment of audiovisual Media Streaming quality
PBAC	Parameter-Based Admission Control
PBX	Private Branch Exchange
PCM	Pulse Code Modulation
PESQ	Perceptual Evaluation of Speech Quality
PHR	Prediction Hit Rate
PoE	Power over Ethernet
POP3	Post Office Protocol version 3
pTopo	Physical Topology
QM	QoSiLAN Manager
QoE	Quality of Experience
QoSiLAN	Quality of Service in unmanaged Local Area Networks

QoS	Quality of Service
QSLP-LAN	QoS Signalling Layer Protocol for Local Area Networks
QSpec	QoS Specification
RACF	Resource and Admission Control Function
RACS	Resource and Admission Control Sub-System
RBS	Receive Buffer Size
RLS	Recursive Last Square
RMSE	Root Mean Square Error
RPP-IPv4	Reservation Path Parameter for IPv4
RRI	Request Identification Information
RR	Receiver Report
RSN	Reservation Sequence Number
RSVP-TE	RSVP Traffic Engineering
RSVP	Resource Reservation Protocol
RTCP	Real-Time Control Protocol
RTP	Real-time Transport Protocol
RTT	Round Trip Time
SCBP	Statistical Class Based Bandwidth Prediction
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol

SNMP	Simple Network Management Protocol
SPID	Statistical Protocol IDentification
SSH	Secure SHell
SVG	Scalable Vector Graphics
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TC	Traffic Control
TDMA	Time Division Multiple Access
TFM	Traffic Flow Management
TFTP	Trivial File Transfer Protocol
THM	Technische Hochschule Mittelhessen (University of Applied Sciences Mittelhessen), Germany
TIA	Telecommunications Industry Association
TLS	Transport Layer Security
TLVs	Type Length Values
TLV	Type Length Value
TMOD-1	Traffic Model
ToS	Type of Service
TP	True Positive
TS	Transport Stream
TV	Television

UDP	User Datagram Protocol
UPnP-QoS	Universal Plug'n Play Quality of Service Architecture
UPnP	Universal Plug'n Play
VBR	Variable Bit Rate
VB	Virtual Buffer
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
WLAN	Wireless LAN
WMA	Windows Media Audio
WMV	Windows Media Video
WRED	Weighted Random Early Detection
XML	Extensible Markup Language

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors Dr. Veselin Rakočević, Prof. Dr. rer. nat. Rudolf Jäger and Professor Muttukrishnan Rajarajan for the continuous support of my PhD study and related research, for their motivation and patience. My sincere thanks also goes to the staff at the University of Applied Sciences Mittelhessen (THM) and the fellow labmates from the Laboratory for Telecommunications, mainly Dr. Christian Überall, Dr. Christian Köbel, Dr. Florian Adamsky and Nils Hellhund for their stimulating discussions and friendship and to the countless students who supported me in different manners. I'm especially grateful to Prof. Dr. Joachim Habermann, Prof. Dr.-Ing. Karl-Friedrich Klein and Prof. Dr. rer. nat. Rudolf Jäger for inspiring, encouraging and supporting me all through the way. Last but not least a special thanks goes to my beloved wife and children and all the family. Words can not express how grateful I am for all of the sacrifices that you've made on my behalf.

dedicated in memories to my father

Declaration

I declare to grant powers of discretion to the University Librarian to allow the thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

Quality of service in local area networks is becoming more and more important, since bandwidth intensive applications are increasing in modern households, but the infrastructure is often limited. State of the art research and standardisation knows of QoS technologies targeting QoS in consumer networks, but these often require deployment on all devices, are limited to certain access technology or lack autonomous configuration support. This thesis presents a novel approach to Quality of Service in unmanaged Local Area Networks, called the QoSiLAN framework. It does not rely on network infrastructure support, but on host cooperation. It identifies traffic with a QoS demand and predicts the required resources to enable per-link bandwidth reservations in an autonomous manner. In contrast to traditional approaches, the bandwidth reservation is not realised explicitly by infrastructure support, but implicitly by host cooperation. This works by involving cooperating hosts, which limit their bandwidth output to not over-provision links with active QoS reservations in the network, while a full device coverage is not required essentially. The resource management and admission control is coordinated by a dedicated QoSiLAN Manager host, which also maintains a detailed link layer network topology map to make sophisticated resource policing admissions on link basis. To enable the QoSiLAN framework, this Thesis contributes the framework as well as new knowledge to the enabler technologies for traffic identification, resource prediction, topology mapping, policing and admission control as well as a dedicated QoS signalling communication protocol.

Chapter 1

Introduction

During the last decade, the use of multimedia services has increased dramatically in home and private networks [Bri09; Eur15]. The Internet Protocol TeleVision (IPTV) and Voice over Internet Protocol (VoIP) services have a demand for high bandwidth capacity and very strong Quality of Service (QoS) needs [Goo02; IPT06]. To guarantee these, common QoS strategies like Integrated Services (IntServ) [BCS94], using the Resource Reservation Protocol (RSVP) [BZ97], or Differentiated Services (DiffServ) [Nic+98; Bla+98; Gro02] have to be supported by the network, as well as by the end-devices. Since in most home or private networks low cost hardware is used, a support for these protocols cannot be assumed. Beside this, the Internet audio and video services tunnel their media streams using the Hypertext Transfer Protocol (HTTP) [RS99] and are therefore not easily distinguishable from the common Internet browsing traffic. This results in less technology acceptance by the users, since lacking QoS leads to a lower Quality of Experience (QoE) levels [Kil08]. Especially for IPTV providers, the network plane inside the households is unpredictable, as Internet providers can only influence the QoS level until the transfer point to the house. In addition to that, the network is a shared medium, in

contrast to the traditional Television (TV) cable, which leads to new challenges for IPTV services to achieve the accustomed QoE level. Local Area Network (LAN) management in unmanaged networks is mostly limited to Internet Protocol (IP) auto-configuration and service discovery protocols like Dynamic Host Configuration Protocol (DHCP) [Dro97], Universal Plug'n Play (UPnP) [BPW13] and Multicast Dynamic Naming System (mDNS)/Bonjour [CK13]. In addition, consumer oriented Internet gateway devices support QoS only at a pass-through level, but are not able to manage the traffic within the LAN. Available QoS solutions like the UPnP QoS Architecture [For06] require implementations on all hosts and network devices and explicit support from all applications, which does not reflect the real world scenario. These aspects lead to QoS problems in consumer LANs, because of the lack of autonomous QoS LAN management capabilities.

This work proposes novel solutions to address QoS challenges in unmanaged consumer networks with the goal to increase the QoE level. The proposed solution is called Quality of Service in unmanaged Local Area Networks (QoSiLAN) and aims to increase the QoS level in unmanaged LANs without relying on QoS support by the network or applications and does not require deployment in all network devices. The host-cooperative approach employs and integrates multiple different technologies from several disciplines to fulfil its tasks. This interdisciplinary approach integrates technologies from the fields of network discovery, traffic identification by machine learning, network protocol engineering, bandwidth prediction and policing and admission control. The initial QoSiLAN framework concept was presented in Athens/Glyfada, Greece at the Second International Conference on Advances in Multimedia (MMEDIA) in June 2010 [Koe+10a], first. Later, a complete overview about the whole solution was published in the Journal of Computer Networks and Communications (JCNC) in 2015 [Koe+15].

The main contribution of this work lies in the novel solution to an unsolved problem in unmanaged LANs. Its layer 3 approach allows it to work across access technologies for fixed and wireless clients, as well as for simplex and duplex access mediums, like PowerLAN and Ethernet, which can be found in consumer and office LANs very often. The unique combination of improved technology usages and algorithms form the proposed QoSiLAN framework, whose proper functional operation and interaction are the core of this work. Further ideas of employing the QoSiLAN framework in wider usage scenarios were discussed at the International Conference on Consumer Electronics (ICCE) in 2011 [Ada+11], but are out of scope of this thesis.

In the past, the problem of resource-/bandwidth-reservation was mostly solved by establishing QoS aware network infrastructure hardware within the LAN. For Differentiated Services (DiffServ), the routers and switches in a network need to implement prioritisation capabilities in order to obey the Type of Service (ToS) or Differentiated Services Code Point (DSCP) - marks in Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) headers or Institute of Electrical and Electronics Engineers (IEEE) 802.3p labels in Ethernet headers [Nic+98; Bla+98]. The Differentiated Services (DiffServ) enable prioritised packet forwarding and scheduling. A much stronger infrastructure relation applies for the most common integrated services solutions for QoS in IP and Multiprotocol Label Switching (MPLS) networks, the RSVP [Wro97]. There, all routers along the data path have to implement the RSVP stack to support bandwidth management in order to ensure network properties on the reservation path, like available bandwidth, jitter and delay. Also, QoS solutions, developed for consumer LANs rely on infrastructure support, like the UPnP QoS [For06] protocol does.

Other solutions, like the LLTD protocol [Cor09] are tools to acquire real-time QoS information without network support, but still require application- and system-support on both end-points. All these technologies require complex implementations on the infrastructure devices, but most low cost and consumer devices do not support any of these QoS technologies. In addition, these known and well researched QoS solutions require manual configuration or support by the applications, which cannot be assumed for most consumer applications, especially, when using Internet services. Other approaches, like Dynamic Adaptive Streaming over HTTP (DASH) [ISO14] aim at lowering the occupied streaming bandwidth in case of resource conflicts, which affects the QoE significantly and does not solve the QoS problem neither inside the LAN nor for the Internet connection. This shows that there is a need for a new solution to the QoS problem in unmanaged LANs, which does not rely on infrastructure and application support, but works autonomously in a smart way. In particular, the self-organisation of user devices is important to consumer targeted solutions, since complex configuration and application decisions cannot be handled by end-users because of the required expert knowledge. The ability to discover network bottlenecks in unmanaged networks is provided by the LLTD [Cor09] protocol, but the decision to handle it is left to the application. In addition, consumer LANs often show a hybrid technology mixture, where different access technology come to application. The set-up is often chaotic and cannot be assumed to be configured optimally. Therefore, a solution is required, which discovers bottlenecks and resolves network issues automatically and provides application independence.

1.1 Contribution

The main contribution of the research presented in this thesis is the novel Quality of Service (QoS) framework, which enables self-organised bandwidth resource reservations for QoS in unmanaged, hybrid local area networks on a per link basis, without support from the network infrastructure. This host based QoS framework is supported by various key technologies, which have been researched and improved to fit them for the QoSiLAN framework. Therefore, contributions are also made in the fields of flow identification using the Enhanced Statistical Protocol IDentification (eSPID), resource prediction using the Statistical Class Based Bandwidth Prediction (SCBP), the new QoS Signalling Layer Protocol for Local Area Networks (QSLP-LAN), the Link Layer Topology Discovery (LLTD) and the Policing and Admission Control for LAN Management. The contributions are:

- Enhanced Statistical Protocol IDentification (eSPID): To enable self-organised and application independent QoS, a daemon must analyse the outgoing traffic from the hosts to identify and classify streams with QoS demand. The contribution is an enhancement to the light-weight traffic flow identification algorithm Statistical Protocol IDentification (SPID), the measure selection and the optimisation of algorithm parameters. The evaluations were performed in a test-bed using recorded real-world flows. The eSPID algorithm was presented in Zurich, Switzerland at the International Conference on Consumer Electronics (ICCE) 2010 [Koe+10b], first.

- Statistical Class Based Bandwidth Prediction (SCBP): In order to reserve resources, the amount of needed bandwidth must be known. In a self-organised system a smart algorithm must estimate the needed bandwidth resources to enable appropriate resource reservation. The

contribution is the design and protocol of the lightweight flow bandwidth prediction algorithm using classifications. It's design was derived from experimental observation and its multi-dimensional optimisation and validation was evaluated using a test-bed with real Internet traffic sources.

- Next Steps In Signaling (NSIS) QoS Signalling Layer Protocol for Local Area Networks (QSLP-LAN): The novel QoSiLAN framework's QoS communication schema requires for an adapted QoS signalling behaviour and an aligned set of properties in order to accomplish the complex task of cooperative, host-based QoS management in LANs. The contribution is the design and protocol of a QoS protocol, based on the stack of the latest Internet Engineering Task Force (IETF) recommendations called NSIS. In addition, the analysis contains a assessment of the complexity and applicability of the NSIS protocol stack. Evaluations were performed in a test-bed with real-world recorded traffic flows.

- Link Layer Topology Discovery (LLTD): The link layer inter-connection of hosts, switches, routers and access points in the network, as well as their link capacity and access technology properties must be known to the QoSiLAN framework, in order to manage the resources in an economic, link based manner. Therefore, the known LLTD Protocol was re-implemented to enable self-organised topology and resources discovery. The contribution is the novel application of the LAN discovery protocol as data source for QoS resource policing and management. The analysis of the protocol was performed within a self-designed Media Access Control (MAC)-layer simulator and test-bed integrations.

- Policing and Admission Control for LAN Management: The QoSiLAN framework with its various measurement and statistical investigated

input parameters requires an individual policing and admission control configuration to manage the resource in the LAN efficiently. The contribution is the design and protocol of the resource reservation admission and policing procedures to enable network-link based resource reservations using collaborative traffic shaping without infrastructure support. The proof of the concept was analysed using a test-bed integration.

As this work is an interdisciplinary approach to QoS in LANs using a complex integration of multiple key technologies, the focus of this work lies more in the functional operation and the proof of the proposed concept, than in the excessive optimisation of selected parameters.

1.2 Thesis Structure

This thesis is structured as follows. Chapter 2 prepares the reader with the background knowledge for all key technologies used in order to develop the interdisciplinary approach. In addition, different QoS solutions for LAN management are discussed. Chapter 3 combines the presentation of the research solution with the evaluations, as well as the approaches, environments and scientific methodology used to achieve the results and evaluations, to keep a central thread between the presented multiple parts of the solution, the contributions and their evaluation. Chapter 4 summarises the results and discusses the achievements. In addition, the potential for future research and possible approaches are proposed.

Chapter 2

Quality of Service in Local Area Networks

The idea for the Quality of Service in unmanaged Local Area Networks (QoSiLAN) framework was inspired by practical observation that most unmanaged networks lack appropriate QoS support, as pointed out in section 1. I consider unmanaged networks to be those LANs that consist of wired and wireless links, regardless of the access technology, that lack the functionality or are not configured for QoS features. The infrastructure devices in unmanaged LANs are not QoS aware, typically. If they are, the QoS features can be considered as not configured properly. In most cases, even if some QoS features are supported and enabled, the functionality is limited to the Internet line management or it can only be configured manually. Even the Microsoft Corporation dropped the RSVP support from their Windows operating system in 2001 [Cor13]. There are already Internet routers for consumers in the market, which perform self-organised traffic flow identification and classification using prioritised packet scheduling [QUA]. This means that at best QoS is only available for traffic passing the gateway and no QoS guarantees are possible as compared to the Integrated Services (IntServ) QoS framework. Wider literature investigations presented

in section 2.3 reveal that this is still an unrepresented research field and that no fully self-organized, host-cooperative approach has been proposed yet. In the following, multiple QoS frameworks and approaches as well as the related work for the required modules to support the proposed QoSiLAN framework are discussed. As presented in this chapter the QoS strategies for most managed traffic situations are known and well researched. But since the IntServ-based QoS implementations can still only be found in professional network equipment or rarely on consumer devices, QoS has not reached the end-users' in-house installations fully, yet. In the remainder of this chapter, the most common QoS frameworks and protocols are presented.

2.1 QoS Measurement

Scientific measurement of QoS requires a definitive and unified definition. The European Telecommunications Standards Institute (ETSI) and the International Telecommunications Union (ITU) defined QoS as the quality perceived by the end user, called the E-Model, resulting in a value R, which represents the user satisfaction [G1008]. Another voice quality testing method was defined by the ITU - Technical Work Group (ITU-T) using the Perceptual Evaluation of Speech Quality (PESQ), which is applicable not only to speech codecs but also to end-to-end measurements. It measures the effects of one-way speech distortion and noise on speech quality. The IETF network working group provided a more concrete definition, where QoS is defined as the set of service requirements imposed on the network, while transporting a traffic flow [Cra+98]. They also defined a framework [Pax+98] for IP performance metrics and measurements. Below the most relevant metrics that constitute the set of service requirements are presented. These metrics are used to assess the perceived QoS. Research efforts have been invested in formulating a relation among the QoS

metrics to determine a final measure for the overall QoS and convert it into a subjective value [CR01; Con02]. For the application of audio/video transmission, the ITU-T defined a recommendation for Parametric Non-intrusive Assessment of audiovisual Media Streaming quality (P.NAMS) [PNA12]. It provides an overview of two recommended objective parametric quality assessment models that predict the impact of observed IP network impairments on quality experienced by the end user in multimedia mobile streaming and IPTV applications over transport formats. It aims on in-service quality monitoring for specific IP-based audiovisual services and benchmarking of different service implementations.

2.1.1 Delay and Latency

The delay or latency of packet delivery is one of the critical parameters for real-time traffic, e.g. as caused by VoIP using the Real-time Transport Protocol (RTP) for audio and video streaming. It's measurement is already integrated within the Real-Time Control Protocol (RTCP), which periodically sends Receiver and Sender Reports including delay information, in parallel to RTP transmission. The delay or latency can also be measured using time stamps and synchronised clocks or using a ping echo tests. The echo tests are not suitable for a reliable test of the end-to-end delay between two nodes, since it measures the round trip delay. Due to asymmetric links and routes in the Internet, packet delay may vary because of different link speed and different routes in both directions. Also the processing delays may vary for sending and receiving on both nodes.

2.1.2 Jitter

Jitter is the variation in the end-to-end transit delay of the packets sent from the sender to the receiver. It can easily be calculated at the receiver.

As shown in (2.1), the jitter value J_{new} is calculated as an average value over 16 values, as recommended by the IETF [Sch+03]. The inter-arrival jitter is the variation in the relative transmit time for a pair of packets and the gain parameter $\frac{1}{16}$ gives a good noise reduction ratio while maintaining a reasonable rate of convergence.

$$J_{new} = J_{old} + \frac{(|rcvd_i - rcvd_{i-1} - send_i - send_{i-1}|) - J_{old}}{16} \tag{2.1}$$

Jitter can be caused by network congestion, timing drift or route changes. High jitter values can have serious impact on real-time traffic, like VoIP, since high jitter values cause the receiver's de-jitter buffer to enlarge, which causes high delays. High jitter values can also have negative impact on Transmission Control Protocol (TCP) connections, since TCP defines time-out values for packet retransmission.

2.1.3 Packet Loss

Packet loss mainly occurs due to buffer overflow and transmission unreliability through wireless channels. If more packets arrive at a certain network node than its buffer can handle, all incoming packets are dropped until the buffer has capacity again. In wireless networks interference and inadequate signal strength can cause packet loss. Although most wireless link layer protocols support retransmission mechanisms, lasting connection problems result in packet loss. Packet loss can be detected by the receiver due to missing sequence numbers or time stamp values in real-time traffic. As defined by RTCP [Sch+03], the packet loss rate is calculated as shown in (2.2), where N is the number of packets and S is the sequence number.

$$loss_{rate} = \frac{N_{expected} - N_{received}}{S_{highest} - S_{initial}} \tag{2.2}$$

2.1.4 Throughput

Throughput is usually measured as the number of bytes that are exchanged between communicating pairs. The maximal throughput of a line can be calculated in the case of TCP as shown in (2.3).

$$T_{max} = \frac{RBS_{max}}{RTT} \tag{2.3}$$

The maximal Receive Buffer Size (RBS) is usually 64Kbyte e.g., forTCP and the Round Trip Time (RTT) can be measured using echo tests. The RTT does not only depend on the link speed, but but also on the processing times of all nodes along the communication path. The throughput is always limited by the lowest link bandwidth in the path. The mean throughput can be estimated by continuously calculating the mean of the received amount of data over the transmission time, as shown in (2.4).

$$T_{mean} = \frac{\sum bits_{received}}{t - t_0} \tag{2.4}$$

The unit most often used to express networking throughput is bits per second (bps). This term is often expressed in thousands, millions or billions as kbps, Mbps or Gbps. It almost always uses the decimal, not binary, versions of the kilo, mega or giga multipliers [Koz05].

2.2 Nature of Internet Traffic

The nature of the Internet traffic depends on the application and service. Signalling protocols e.g., for routing and auto-configuration, typically cause periodic bulk traffic with a small amount of data, whereas data traffic for file transfer causes unpredictable bulk traffic with possibly high throughput peaks. In contrast to bulk traffic, streaming traffic appears more or less continuously. One can identify Constant Bit Rate (CBR) and Variable Bit Rate (VBR) streaming. Most QoS schedulers aim to manage VBR streams by using intelligent buffering to produce streams, which result in constant bit-rate streams, which are easier to manage from the QoS's view.

2.2.1 VoIP

The Voice over Internet Protocol (VoIP) is the transmission of telephone calls using the Internet Protocol (IP). The most common VoIP protocol for call management over the IP is the Session Initiation Protocol (SIP) [Ros+02]. The voice data transmission is usually handled by the RTP [Gro+96; Sch+03]. As for the transport protocol Transmission Control Protocol (TCP) can be used as well as User Datagram Protocol (UDP).

2.2.1.1 Characteristics

The VoIP traffic is characterised by a constant bit-rate between 4.2Kbps and 64Kbps [Goo02], depending on the codec. The frame duration differs from 20 ms to 0.125 ms. The most common Codec is the G.711 [Boo88]. The G.711 specifies a Pulse Code Modulation (PCM) codec, using 8000 samples per second using 8 bit resolution. Since the transit delay of packets is an important speech quality factor in VoIP, small packet sizes are desirable. This results in a bit-rate of 64 Kbit/s. Since RTP/RTCP [Sch+03] is used, a

5 % additional overhead throughput demand can be assumed. VoIP services use RTP/RTCP in most cases. There, the Private Branch Exchange (PBX) Server is free to choose TCP or UDP as the transport protocol. H.323 [IPT09] and SIP signalling use TCP in most inter-domain communication cases.

2.2.1.2 Measurement

Considering that RTCP is used for the delivery of VoIP traffic, QoS measurements are part of the system. Every client is supposed to send a Receiver Report (RR) periodically to the sender to report the reception quality. The RR contains information about the reception quality e.g., delay, jitter and throughput. In addition, the IETF proposed a framework for Traffic Flow Management (TFM) [BMR99]. The TFM provides a general framework for describing network traffic flows, presents an architecture for traffic flow measurement and reporting, discusses how this relates to an overall network traffic flow architecture and indicates how it can be used within the Internet. The TFM framework defines meter and manager entities. The meter nodes transparently analyse the traffic on per flow basis, by sorting packets by characteristics e.g., transport protocol and port. The manager is responsible for configuring and controlling one or more meters. A meter can be a measurement device or be implemented in routers, switches or client systems. Beside the technical parameters, subjective parameters are important to assess the quality of audio and video transmission. The ITU-T recommends a Mean Opinion Score (MOS) value [nt96], which is defined as the arithmetical mean value of several standardised measurement methods. They evaluate the subjective impression of audio and video quality. The MOS value scores range from one, the worst, to five, the best, individual quality impression.

2.2.2 IPTV

The Internet Protocol TeleVision (IPTV) is the transmission of moving images using the Internet Protocol (IP) as transport medium. The video is encoded digitally and transmitted using streaming protocols like RTP [Gro+96; Sch+03], Microsoft Media Server Protocol (MMS) [Cor15b] or Moving Picture Experts Group (MPEG)-Transport Stream (TS) [ISO15]. Emerging adaptive streaming protocols are MPEG-DASH [ISO14] or HTTP Live Streaming (HLS) [PM15], which are able to adapt the streaming bitrate according to the available bandwidth. IPTV can use either TCP or UDP as the transport layer protocol.

2.2.2.1 Characteristics

The IPTV traffic characteristic is totally different to the characteristics of VoIP traffic. In contrast to common thinking, IPTV traffic cannot be regarded as high bit rate VoIP traffic, as pointed out in [Tec07]. Instead, IPTV traffic has reached the maximum Ethernet packet size and has, depending on the encoder, not a continuous packet pattern, but often a bursty one with varying inter packet gaps. The typical throughput demand for IPTV is about 3.75 Mbps [WC06]. In addition, IPTV stream sessions last longer. The authors from [Tec07] state that most router and switch hardware was never tested for such requirements, which often causes difficulties to locate problems. This is independent from the bandwidth capacity, but is related to the implemented queueing behaviour and buffer sizes. High quality IPTV streams exceed with two packets, which have a size of 1200 Bytes each, derived from the the common Maximum Transmission Unit (MTU) in LANs, the buffer size of a common router, which is about 2000 Bytes. This causes packet loss, if additional IPTV streams are present on the same link, even if the bandwidth capacity is about 10 Gbps.

2.2.2.2 Measurement

To represent the quality characteristics of IPTV the IETF proposed a Media Delivery Index (MDI) [WC06]. It defines a Delay Factor (DF) and a Media Loss Rate (MLR) to express the video transmit quality. The DF is estimated as shown in (2.5),

$$DF = \frac{VB(max) - VB(min)}{MR} \tag{2.5}$$

by dividing the variation of the Virtual Buffer (VB) sizes by the nominal Media Rate (MR). The Media Loss Rate is the count of lost or out-of-order flow packets over a selected time interval, where the flow packets are packets which may carry one or more media frames. This is why deep packet inspection is needed. The MDI is defined as $\frac{DF}{MLR}$. Although the MDI measurement requires deep packet inspection because of its UDP nature, MDI values give a very clear picture of potential queue usage and performance by showing flow pattern changes per flow in real time. Therefore, the MDI also helps in separating encoding problems from network problems.

2.3 Approaches for QoS in Local Area Networks

This section discusses various standardisation and state-of-the-art research approaches to manage Quality of Service (QoS) in Local Area Networks (LANs).

2.3.1 Differentiated Services

The Differentiated Services (DiffServ) is a QoS method to prioritise IP packets, specified by the IETF [Nic+98; Bla+98; Gro02]. A conventional IP network cannot distinguish between applications the packets belong to. Therefore, identifiers are introduced in the IP header fields to announce the importance of packets to routers, switches and hosts. If a network element supports DiffServ, it decides a privileged forwarding to the receiver, based on the used identifier. As identifier, the Type of Service (ToS) byte within the IPv4 header or the Class Field in the IPv6's header is used. To delimit the ToS and Class Field from former definitions it is called Differentiated Services Code Point (DSCP). The DSCP uses the `class sectors` and the `drop precedence` fields, with three bits each, to define the packet handling and scheduling priorities. Using these parameters, forwarding rules can be defined, which are handled by QoS policies within the network routers. A DSCP value of e.g., `000 000` defines a best effort handling, whereas a DSCP value of e.g., `101 110` defines an expedited forwarding. Like Bless et al. [BNW03] have shown, best effort classified traffic can harm the expedited traffic in terms of packet loss rate and throughput. Therefore, they propose a lower effort per-domain behaviour for DiffServ handling. They show that a lower than best effort classification of bulk traffic, in conjunction with a Weighted Random Early Detection (WRED) queueing enhances the packet loss rate probability and throughput performance of the higher

priority traffic dramatically [BNW03]. In parallel, the lower effort bulk traffic remains in acceptable performance ranges.

The DiffServ architecture is not designed to reserve resources or to give delivery guaranties, but to enhance the performance of labelled flows in contrast to the default best effort packet delivery behaviour in IP networks. The advantage is the path and flow independence and the omitted signalling effort. Newer research [Cho+03; BNW03] revealed, that higher prioritisation is not sufficient to guarantee expedited QoS. In contrast, lower effort packet scheduling for common traffic is recommended to achieve the desired QoS level using the DiffServ QoS framework.

2.3.2 Integrated Services (IntServ)

The Integrated Services (IntServ) architecture [BCS94] is an extension to the Internet Protocol (IP) to support resource reservations for real-time and non real-time QoS applications, since the IP was originally designed for best effort data packet delivery. The IntServ defines a packet scheduler to enable sophisticated packet forwarding, a classifier to differentiate the packets and an admission control function to determine if a requested QoS parameter can be granted without impacting on other QoS requests. The IntServ supports two different QoS methods. First, the guaranteed QoS method to ensure bandwidth capacity, delay, jitter and drop parameters for a QoS request and second, the controlled load method. The original reference implementations for IntServ was the RSVP, which will be described in section 2.3.2.1. Lately the IETF proposed a new harmonised signalling protocol framework, the NSIS framework [Han+05], which added the NSIS Signaling Layer Protocol for Quality-of-Service Signaling (NSIS QoS-NSLP) Signalling [J M09], as described in section 3.5.

2.3.2.1 Resource Reservation Protocol (RSVP)

The Resource Reservation Protocol (RSVP) [Bra+97] was designed to operate in the IntServ architecture [BCS94]. The Traffic Engineering extension to RSVP, the RSVP Traffic Engineering (RSVP-TE), is a commonly used protocol in provider and core network routers using the Multiprotocol Label Switching (MPLS) to reserve QoS characteristics on a simplex path for a specified customer [Awd+01]. All routers along the data path reserve network resources for the requested flow. The RSVP operates as an application protocol on top of IPv4 or IPv6 and can use the UDP as well as the TCP transport. It supports unicast as well as multicast operation. The RSVP itself does not transport data. It only works as a signalling protocol for resource reservations and is transparent to router not supporting the RSVP. The RSVP uses soft states to manage reservation states in routers and hosts for providing support for dynamic route changes. This causes periodically sent `Path` and `Resv` messages to refresh the reservation sessions and states, as shown in Figure 2.1. States are deleted, if no matching

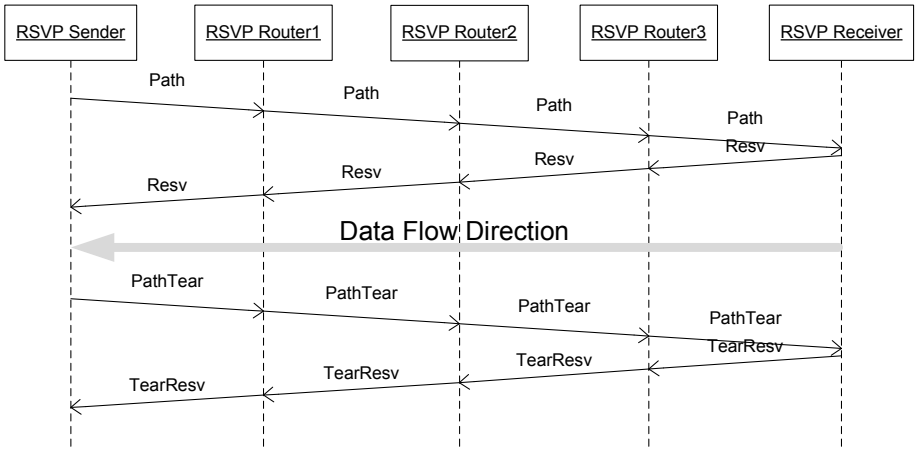


FIGURE 2.1: RSVP Signalling Diagram and Data Path

refresh message arrives before the expiration of a clean-up time-out interval. States can also be deleted by sending tear down messages. To create a new RSVP session, a host creates a session identification consisting of the

destination address, the protocol id and if multiple sessions are supported, also the destination port. During the session setup this identification is

1 = Path	The sender sends the RSVP path message. It is used to determine a possible path to the receiver, see Figure 2.1. Along the path, the RSVP supporting routers record themselves as RSVP Hop Object in the RSVP Path Message's payload to report their existence to the receiver.
2 = Resv	Along the recorded path, the receiver sends a RSVP reservation message back to the sender of the preceding RSVP Path Message, see Figure 2.1. It contains the flow specifications for the reservation.
3 = PathErr	RSVP path error messages report errors in processing Path messages. They travel upstream towards senders and are routed hop-by-hop using the path states. At each hop, the IP destination address is the unicast address of a previous hop. PathErr messages do not modify the state of any node through which they pass; they are only reported to the sender application.
4 = ResvErr	The RSVP reservation error messages report errors in processing Resv messages, or they may report the spontaneous disruption of a reservation e.g., by administrative pre-emption. The destination address is always the next hop's address.
5 = PathTear	The RSVP path tear down message is sent from the sender to the receiver to delete the reservation state in the routers along the path, as shown in Figure 2.1.
6 = ResvTear	The RSVP reservation tear down message is sent from the receiver to the sender to delete the reservation state in the routers along the path, as shown in Figure 2.1.
7 = ResvConf	The RSVP reservation confirmation messages are sent from senders to receivers to confirm reservation requests. The ResvConf messages are only sent, if a RESV_CONFIRM object was present in the corresponding Resv message.

TABLE 2.1: RSVP Message Type Field Definition

communicated along the data path. The common header of RSVP messages is shown in Figure 2.2. It has a length of 8 bytes and starts with

a 4 bit version field, which is always set to 1. The 4 bit flags field is not assigned yet. The 8 bit message type field can have values from 1 to 7 with the purposes as listed in Table 2.1. The RSVP checksum field uses the one's complement of the one's complement sum of the message, with the checksum field replaced by zero. The Send_TTL value limits the number of hops a RSVP message can survive. The RSVP length field specifies the overall length of the RSVP message, including the payload. The payload

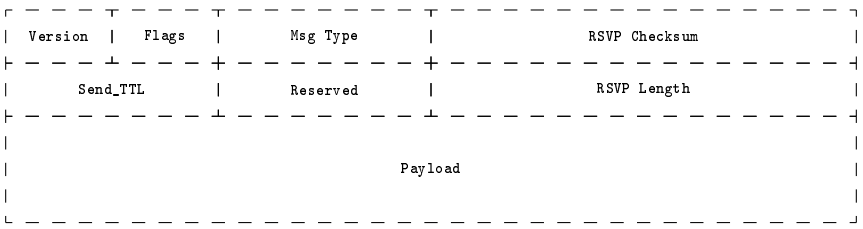


FIGURE 2.2: RSVP Common Header

of a RSVP message carries RSVP the objects. RSVP objects consist of a 32 bit header and one or more 32 words payload, as shown in Figure 2.3 The RSVP object header's length field specifies the length of the whole

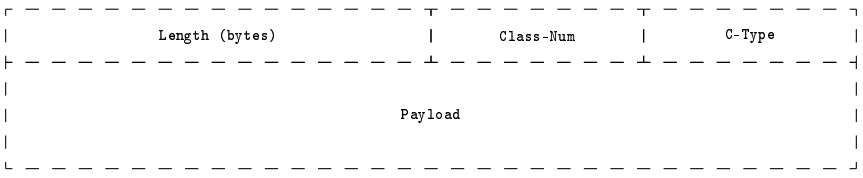


FIGURE 2.3: RSVP Option Format

object, measured in bytes. The Class-Num field identifies the object class as defined in RFC2205 [Bra+97] appendix A. E.g., one important class is the RSVP_HOP class, which uses the Class-Num value `3`. Each router attaches this object to each path massage it forwards. It carries the IP address of the router's sending interface and an interface handle to distinguish its different logical outgoing interfaces. The values of the C-Type field are defined within the Class-Num definitions e.g., the RSVP_HOP class defines the value of `1` to the IPv4 address information and the C-Type value of `2` to indicate the IPv6 address information.

The RSVP was developed to meet the needs for Internet real-time applications such as video conferencing and streaming and to support peering provider's business models based on selling network resources. However, its deployment in LANs has never fulfilled the expectations [MF05]. End-user Internet applications and consumer operating systems usually do not support RSVP. Instead, an extension, the RSVP-Traffic Engineering (RSVP-TE), defined in [Awd+01] and [FAV08] is widely used by network and backbone providers in their MPLS networks to provide QoS parameter promises like bandwidth, latency and packet loss to their customers. The RSVP-TE generally allows the establishment of MPLS Label Switched Paths (LSP), taking into consideration network constraint parameters such as the available bandwidth and the explicit number of hops.

2.3.2.2 Localized RSVP

The Localized RSVP [Man+05] was designed to enable hosts to request resources from their host network, if the correspondent sending node lacks RSVP support. It utilises a local RSVP proxy server, usually also an Internet gateway, to act on behalf of the correspondent node. This RSVP proxy scans for the Local Indication (LI) bit (`bit 0x8`) to differentiate reservations that are internal to the access network and those, which are external with participation of a correspondent node. Hence, the RSVP signalling is not forwarded to the next hop, but instead answered directly by the RSVP proxy application. The protocol also adds a second bit to the RSVP header, the Expedited Refresh (ER) bit (`bit 0x4`), to indicate that a Path message is sent as a refresh to a broken path and must be forwarded immediately. This is necessary to allow for mobile receiving nodes to change the local access router and to refresh and repair the reservation path right after a hand-over has occurred. It also introduces two new message types. The Path Request (type 8) is used to request a Path message from the local RSVP proxy. The Path Request Tear (type 9) was introduced to

allow the receiver to tear down a downstream reservation. Due to the newly introduced bits and message types all RSVP routers within the local access network need to support the Localized RSVP extension, not only the proxy.

The Localized RSVP is designed for fixed and mobile access providers to offer RSVP services to end-hosts, even if the correspondent node does not support the RSVP. This increases the use and applicability of RSVP dramatically, but since RSVP featured end-user applications are rare, a wide distribution of Localized RSVP will most likely fail to appear as the classical RSVP did. It is not expected to reach an appropriate relevance. In addition, Localized RSVP does not make sense in LANs with QoS unaware LAN switches, since the reservation of resources would be network wide and not only for the desired path. To efficiently reserve resources RSVP aware switches and router are mandatory. These reasons were a significant source of inspiration for the design of the QoSiLAN framework to find a proper QoS solution for unmanaged LANs.

2.3.2.3 Next Steps in Signalling (NSIS)

The Next Steps In Signaling (NSIS) protocol framework [Han+05] aims to provide a flexibly designed protocol stack to optimise and to harmonise the various existing signalling protocols. The different existing protocols are all designed for special signalling situations and use their own transport mechanisms for signalling. The NSIS framework provides a two layer approach, as presented in Figure 2.4. It is based on the common TCP/IP stack, and optional Transport Layer Security (TLS) [DA99; DR06; RM12] layer for encryption. The signalling transport and host discovery is managed using the NSIS Transport Layer Protocol (NTLP) which is usually implemented using the General Internet Signalling Transport (GIST) protocol, which hosts the NSIS Signaling Layer Protocol (NSLP). The signalling layer, defined by the NSIS Signaling Layer Protocol (NSLP) framework performs

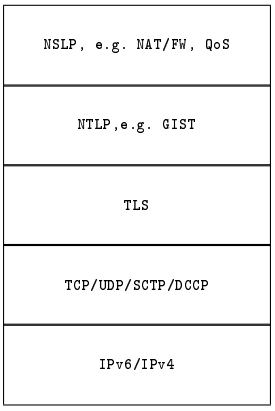


FIGURE 2.4: NSIS Protocol Stack

the actual signalling communication. The NSIS's design was mainly built on consolidated findings and experiences made during the research of the RSVP. Therefore, in the first development stage a QoS approach as a showcase was followed, which aimed at providing the functionality of the RSVP, to demonstrate the technical superiority of the new protocol design in contrast to the traditional QoS signalling protocols, especially the RSVP. The NSIS working group's long term goal is to harmonise the signalling protocols by using a standardised design and separate the signalling from the transport. The RSVP replacement in NSIS uses the General Internet Signalling Transport (GIST) for transport and the QoS-NSLP for signalling.

2.3.2.3.1 General Internet Signalling Transport (GIST)

The General Internet Signalling Transport (GIST) [SH10] protocol utilises common transport and security protocols and builds a message transport layer on top for providing a common service layer to signalling protocols and applications. The GIST protocol does not perform signalling itself, but configures and prepares underlying protocols for enabling duplex message

flow to upper protocols. The GIST protocol does not replace current transport protocols and does not influence paths or routing. It also handles only its own states and the configuration of lower layer protocols, but not the states of higher layer signalling protocols. The main purposes of the GIST protocol are:

- NSIS node discovery
- message routing between NSIS nodes
- signalling message transport

Therefore, the GIST protocol provides a re-usable base for other signalling applications. The GIST protocol supports different routing methods: Predictive Routing, to signal a path the data flow may use in the future and Network Address Translation (NAT) address reservations to support NAT traversal for NSIS nodes behind NAT firewalls. Most GIST functionality is transparent to these routing mechanisms. Hence, the GIST protocol encapsulates this behaviour in a Message Routing Method (MRM). The GIST protocol knows of five different message types. All NSIS signalling applications have to use these messages and to encapsulate their data as payload. The message types are **Query**, **Response**, **Confirm**, **Data** and **Error**. The Query, Response and Confirm messages implement the three way handshake GIST uses to set up a routing state and to perform messaging associations. The Data message is for signalling data transport of higher layer signalling protocols. The Error message is for error reporting between NSIS nodes. For all message types the use of the common header is mandatory as first header. The header has a fixed format, as presented in Figure 2.5. The Version field defines the GIST protocol version, which is currently always set to `1`. The GIST hops field defines the number of hops a message lives, before it is discarded. The Message Length gives the total number of 32-bit words of the payload's length, excluding the

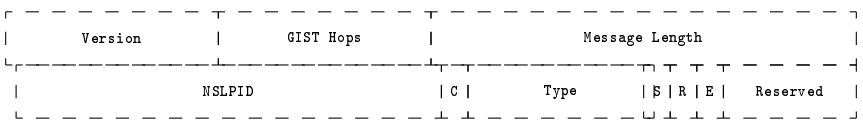


FIGURE 2.5: GIST Common Header

Common Header. The NSLPID is defined by the Internet Assigned Numbers Authority (IANA) organisation and provides the payload's NSLP type identification. The C-flag sets whether the message has to be interpreted by NSIS nodes in the absence of routing information. The Type field defines to which of the six GIST types it belongs. The S-flag is set, if the IP source address is the same as the originator of this message. The R-flag defines if a reply message is requested. The E-flag indicates that the message was explicitly routed.

The GIST's payload must consist of multiple Type Length Value (TLV) entries in a fixed format, as shown in Figure 2.6. The A- and B-flags within

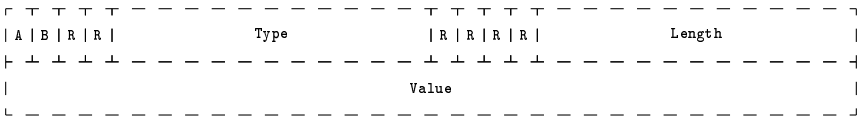


FIGURE 2.6: GIST General Object Format

the GIST's General Object Format header are reserved for extensibility. They define, if a message carrying TLVs must be rejected, if a TLV is not understood, if the TLV must be silently ignored or if the TLV must be forwarded anyway. The R-flags are reserved bits. The values of the Type field are specified by the IANA organisation for identifying the type of object. The Length field defines the length of the Value field, which is variable, measured in 32-bit words.

2.3.2.3.2 NSLP for Quality-of-Service Signalling (QoS-NSLP)

In cooperation with the GIST protocol, the NSLP for Quality of Service Signalling (QoS-NSLP) [MKM10] mainly provides similar functionality as RSVP. It supports different QoS frameworks, but in contrast to RSVP, multicast is not supported, yet. the reason for skipping multicast support is to eliminate complexity. The purpose of QoS-NSLP is to maintain the soft states and QoS reservations along the data path. It supports both sender and receiver initiated reservation, as well as reservations between arbitrary nodes. Hence, it follows the IntServ framework like RSVP. The QoS-NSLP supports four message types:

- **RESERVE**
  This message type is used to create, refresh, modify and remove reservation states in QoS-NSLP nodes. The format of this message is presented in Table 2.2.

```
RESERVE  =  COMMON_HEADER
            RSN [RII] [REFRESH_PERIOD] [BOUND_SESSION_ID]
            [[SESSION_ID_LSIT] [RSN_LIST]] [MSG_LIST]
            [INFO_SPEC] [[PACKET_CLASSIFIER] QSPEC]
```

TABLE 2.2: NSLP RESERVE Message Format

- **QUERY**
  This message type is used to request information about the data path. The format of this message is presented in Table 2.3.

```
QUERY  =  COMMON_HEADER
          [RII] [BOUND_SESSION_ID]
          [PACKET_CLASSIFIER] [INFO_SPEC] QSPEC
```

TABLE 2.3: NSLP QUERY Message Format

- **RESPONSE**
  This message type is used to notify the sender of RESERVE or QUERY messages about the results and provides information about the current states, features and errors. The format of this message is presented in Table 2.4.

```
RESPONSE  =  COMMON_HEADER
             [RII / RSN] INFO_SPEC [[SESSION_ID_LIST]
             [RSN_LIST]] [QSPEC]
```

TABLE 2.4: NSLP RESPONSE Message Format

- **NOTIFY**
  This message type is similar to RESPONSE messages, but does not refer to any other message and can be sent asynchronously. Typically, this message is related to error reporting. The format of this message is presented in Table 2.5.

```
NOTIFY  =  COMMON_HEADER
           INFO_SPEC [QSPEC]
```

TABLE 2.5: NSLP NOTIFY Message Format

The QoS-NSLP header fields are defined as follows:

- All QoS-NSLP messages have to start with a NSLP Common Header, which differs from the GIST Common Header. The NSLP Common Header is presented in Figure 2.7. The Message Type field associates

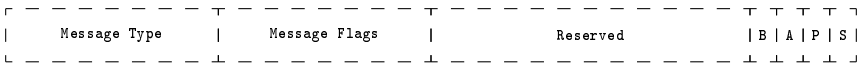


FIGURE 2.7: NSLP Common Header Format

the message to one of the four QoS-NSLP message types. The Message Flags field is message specific. This means, that each message type defines its own message flag according to its needs. E.g., the RESERVE message defines two flags. A tear flag to delete existing reservation states and a replace flag to overwrite existing reservations. The four flags at the end of this message are called Generic Flags and are important to all QoS-NSLP nodes. The S-flag (Scope) indicates that the message shall not be forwarded after the next hop. The P-flag (Proxy) indicated that a message shall not be forwarded by proxies. The A-flag (ACK-REQ) requests to acknowledge the message. The B-flag (BREAK) indicates that there are routers along the path lacking QoS-NSLP support.

- The header objects are defined using the GIST's TLV format as presented in 2.6.
  - Request Identification Information (RRI) - Request Identification Information: This is an identifier, which must be unique within the context of a SESSION_ID and must be re-generated on every new response. It is also used to match RESPONSEs to QUERY and RESERVE messages.
  - Reservation Sequence Number (RSN) - Reservation Sequence Number: This is an incrementing sequence number to define the ordering of state modifications.
  - REFRESH_PERIOD - Refresh Period: This is a time-out value, measured in milliseconds, to define the lifetime of soft states.
  - BOUND_SESSION_ID - Bound Session ID: This is a binding code that indicates the nature of a binding, e.g., bi-directionality or end-to-end connection.
  - PACKET_CLASSIFIER - Packet Classifier: This is a classifier to set priority values and flow labels, similar to DiffServ.

- INFO_SPEC - Information Object and Error Codes: This object contains error information, like the error code, the error class, the error source and error specific information.
- SESSION_ID - Session ID and
- SESSION_ID_LIST - Session ID List: These objects contain the Session ID, used in summary refresh and summary tear messages to identify the soft states and flows.
- RSN - Reservation Sequence Number and
- RSN_LIST - Reservation Sequence Number List: These objects contain a reservation sequence number to identify soft states within a session.
- BOUND_MESSAGE_ID - Bound Message ID: This object contains a message binding type, which identifies the message binding relation. It contains also a message id to relate to other messages.
- QSPEC - QoS Specification: This object contains the QoS information, which can be of different nature, depending on the used QoS model. The encoding and format is defined in [G A09].

The NSIS framework provides an extendible basis to implement new signalling protocols. It reaches its goals by extendability and flexibility. The current specifications to QoS signalling only reflect the QoS model as the RSVP does. In order to support host cooperative QoS signalling without the need for message forwarding, but with additional parameters required as the QoSiLAN framework does, further adjustments and extensions will be needed, as carried out in chapter 3.

2.3.2.4 Conclusion

The IntServ framework enables reliable resource reservation for IP based networks. Common protocols like the RSVP and its various extensions aim to provide the signalling for the most common network situations. The new NSIS protocol family defines a generic way for future signalling protocols, redefining the RSVP in the first place. Successful implementations of these protocols mostly relies on being supported in all: the end-systems, the applications and the infrastructure hardware as well. This causes less distribution of these protocols, since support of application software is rare and native end-system operating system support is not available for customers, commonly. In consequence, only access-network and backbone providers use the RSVP-TE because of monetary needs, since they need to separate/isolate customer streams and to ensure the quality parameters the customers paid for.

2.3.3 Cross Layer Approaches to Quality of Service in Local Area Networks

This section deals with research solutions to enable QoS in LANs, which employ or modify lower layer information.

Modifications of the Ethernet's media access control mechanism, Carrier Sense Multiple Access/Collision Detection (CSMA/CD), used by the 100 Mbit Ethernet standard, are proposed by Endemann *et al.* [EKJ05]. They propose to modify the length of the Time Division Multiple Access (TDMA)'s Inter Frame Gap (IFG), dependent on the packet priority, as shown in Figure 2.8. High priority packets are sent using the minimal IFG, defined by the Ethernet standard and packets with lower priority are sent using longer IFGs. The main advantage of this solution is being a media access (layer 2) approach, which enables QoS support directly in the medium and

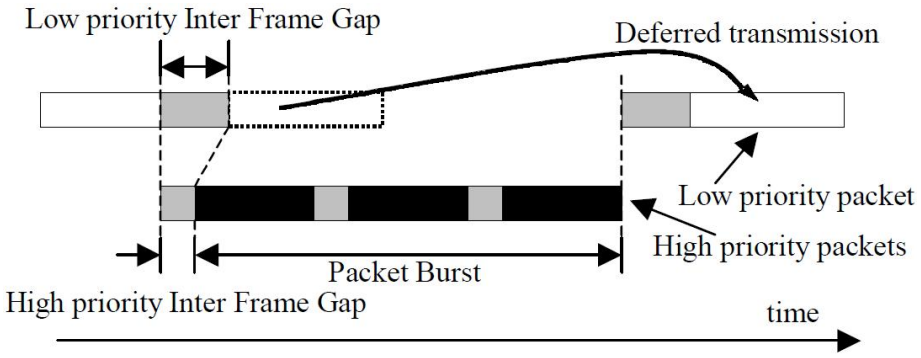


FIGURE 2.8: FTT Client Node Architecture [EKJ05]

promises the best results, when synchronisation is assured. Although this approach is advantageous, its practical relevance is limited, as dedicated network interface firmware and network driver modifications are required to support the solution. Unless manufacturers offer these, the applicability of this solution for consumers is unlikely. In addition, home LANs are typical equipped with heterogeneous access technologies, where this approach is not applicable.

Pedreiras *et al.* [Ped+05] propose a new data link layer protocol. They state that the Ethernet protocol was neither designed for real time data transmission, nor for QoS guarantees and is missing essential features, like real-time support. They propose the Flexible Time-Triggered (FTT) protocol. The FTT protocol is a new flexible real-time communication protocol that supports dynamic QoS management on Ethernet based systems. As presented in Figure 2.9, the FTT protocol works on top of the Ethernet layer and adds a data transmission control layer. For non real-time traffic the common TCP/IP stack is used, whereas for real-time traffic the TCP/IP stack is bypassed, but managed by a real-time Application Programming Interface (API) on application layer level. This produces transparency to standard non real-time application, but requires user applications to support the real-time application API from the FTT protocol. To fulfil its goals of real-time transmission and dynamic QoS management, the FTT protocol

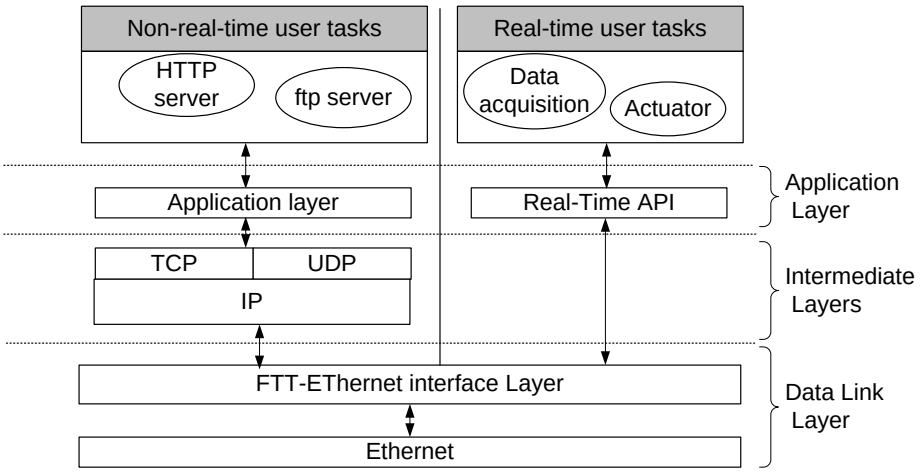


FIGURE 2.9: FTT Protocol Client Dual Stack Node Architecture [Ped+05]

utilises centralised scheduling and a master with multi-slave transmission control. One of the main disadvantages of the FTT protocol is that it requires operating systems which support real-time kernels. This disqualifies FTT from implementation on end-user equipment like personal computers or mobile devices as manufacturers choose not to install real-time kernels on typical end-user equipment . In addition, heterogeneous network topologies cannot be covered by this solution.

Within the research project HOMEPLANE [Dor], an approach addressing QoS in wireless home networks was developed. Hundt et. al. [Hun+07] propose to introduce a home profile for future Wireless LAN (WLAN) standards. This should include a modified interleaver and additionally, a shortened guard interval of 200 ns to be defined. To minimise radio interference they also propose dynamic frequency hopping. These modifications are part of a cross-layer concept, as shown in Figure 2.10. From a Management Information Base (MIB) collected physical layer data is acquired and interpreted by a resource manager, which controls channel parameters of the physical as well as the link layer. The proposed modifications were

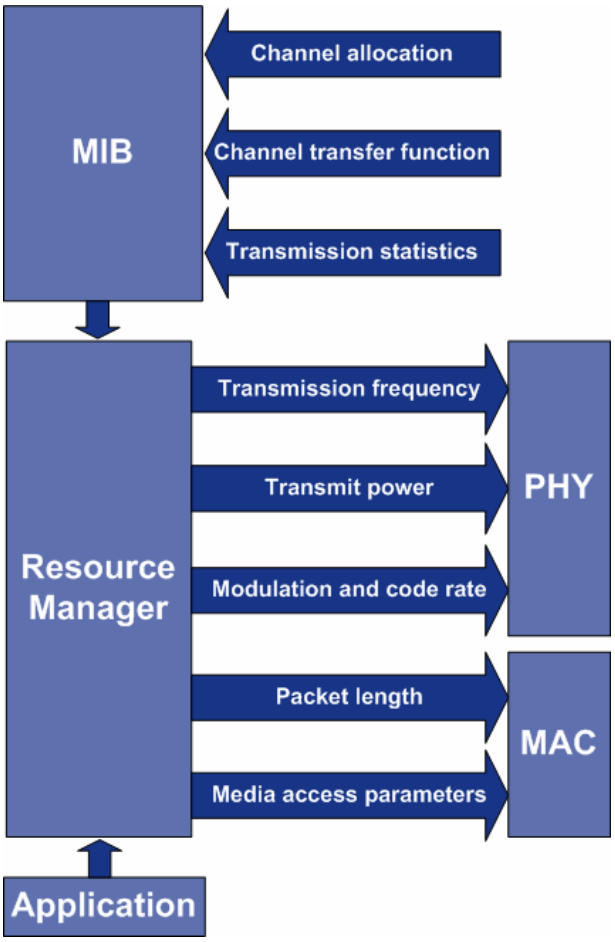


FIGURE 2.10: The HOMEPLANE Cross-layer concept for joint link adaptation [Hun+07]

evaluated using an IEEE 802.11g test-bed. They plan to propose these features into the IEEE 802.11 standards track.

As pointed out, cross layer approaches are able to provide to most advanced QoS level, since they rely on and have access to real-time information across the layers, even to the physical layer. This enables data-mining for most efficient resource management. However, as long as standard bodies and manufacturers do not feature a single solution to have a large distribution

in the market, interoperability is not feasible. Especially, interoperability across different access technologies is a major problem. Therefore, the QoSiLAN framework cannot rely on cross layer support, since interoperability across access technologies was a major design goal.

2.3.4 Application Layer Approaches to Quality of Service in Local Area Networks

This section deals with application layer approaches to QoS in LANs from standardisation as well as from state of the art research.

An approach, most similar to the QoSiLAN framework, has been proposed by Louvel et al. [LPB13; Lou+11; Lou+10], who propose a network resource management framework for multimedia applications distributed in heterogeneous home networks. In this solution to QoS for multimedia applications in LANs, a central management entity, called the Global Resource Manager (GRM) is used as a resource coordinator. On the local devices, the required components are bundled in a Local Resource Manager (LRM). The LRM provides a resource estimation method, implemented using the Bienaymé-Chebyshev inequality algorithm [Gha05] and a scheduling tool for traffic prioritisation, the Linux iproute2 tool's `tc` command [Too15]. The GRM measures the available bandwidth on the links using the iperf tool and coordinates the resources. As main differentiation criteria, Louvel's proposal does not take into account the network topology and limits the approach to a star topology with heterogeneous network interfaces and different devices attached. As the GRM, as a central entity is able to manage all resources, a dedicated QoS protocol is not needed and obviously out of scope of the approach. This limits its practical applicability in home networks dramatically. In the defined topology it benefits from the non-intrusive and adaptable resource management approach, since the

end-devices do not need to be modified essentially to achieve the desired QoS level.

The UPnP Forum proposes a UPnP QoS architecture [For06] to enable QoS services in LANs, consisting of a single IP sub-network. This architecture defines policing and admission control for prioritised, parametrised and hybrid QoS control for individual links, as well as path property discovery for them. Three services are required to implement this functionality, as presented in Figure 2.11. The *QoSPolicyHolder* service gathers path in-

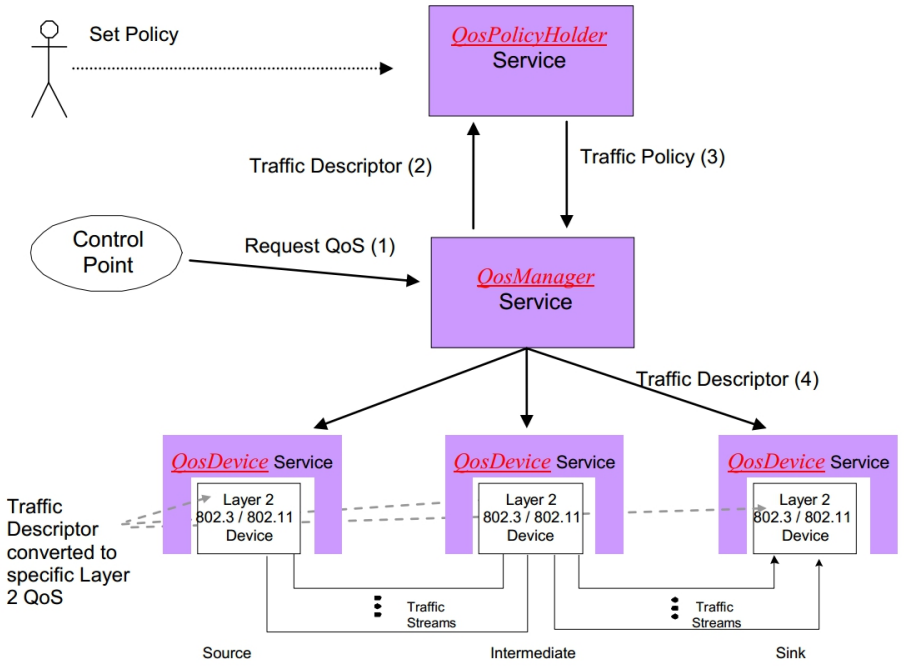


FIGURE 2.11: UPnP-QoS Architecture Overview [For06]

formation and provides appropriate policies for the traffic, described by a *TrafficDescriptor* structure. The *QosManager* service, invoked by an application and implemented within a UPnP Control Point, requests the required resources from the *QoSPolicyHolder* service. The *QosDevice* service is responsible for establishing the QoS for a new traffic stream. To support end-to-end prioritised QoS, the *QosDevice* service needs to be implemented

on all network devices along the data path. For links not supporting the parametrised QoS on the path, prioritised QoS is selected, resulting in a hybrid QoS operation. Network segments not supporting the UPnP QoS architecture result in a QoS establishment failure for the whole end-to-end path. This is an aspect my proposed QoSiLAN framework overcomes, since it does not rely on network support and it can operate, even if not all devices fully support the system. In addition, the UPnP framework requires not only implementation of its services on all devices, but also support by the applications generating the traffic.

This problem was addressed by Laulajainen and Hirvonen [LH09], who propose background services running on traffic causing devices, performing fast application flow detection based on statistical analysis of the first four packet sizes of a stream, which provides a basic identification functionality. This measure is also a small part of the eSPID, described for the QoSiLAN framework in section 3.3. Suraci et al. state, that the Universal Plug'n Play Quality of Service Architecture (UPnP-QoS) architecture works well in the case of moderate traffic loads, but may fail whenever the network becomes overloaded [Sur+10]. They demonstrate their admission control and drop solutions using test-bed evaluations. For the admission control algorithm they rely on the bandwidth information provided by the UPnP framework. If a network segment on the data path has not sufficient residual bandwidth left, the admission is rejected, otherwise it is admitted. The drop strategy decision algorithm is realised using a binary tree to estimate the lowest cost for packet dropping. They determine the cost by the importance/priority of the flow. The QoSiLAN framework, presented in chapter 2.3, also defines an admission control algorithm, which also supports none QoS aware network segments, in contrast to their solution. A dedicated drop strategy is implicitly given by the traffic shaping policy the QoSiLAN framework requests from the operating system. Suraci et al. go one step deeper in the scheduling of packets with their proposal for application QoS management and session control in a heterogeneous home network using inter-MAC layer

support [CAS+10]. They propose an architectural and procedural definition of the home context using the UPnP-QoS and SIP frameworks. Within this architecture, called OMEGA, they introduce a convergence layer between the IP and MAC layer to manage all traffic using the information provided by the UPnP-QoS framework. This architecture requires implementations on all network devices and also uses a centralised coordinator within the network's gateway to manage the resources.

Chen et. al. [CCC07] propose a DiffServ focused scheme for QoS management in heterogeneous home networks. It also adopts to the UPnP-QoS specification by adding monitoring and resource management functionality to the framework. They monitor real-time network traffic to adaptively control the bandwidth and are able to reduce jitter latency and packet loss significantly. The focus of this particular work is set on the last mile from the service provider to the home network, which is out of scope of the QoSiLAN framework. Since the QoSiLAN framework may control the Internet gateway, it also manages these resources and controls the Internet line.

Furthermore, Chan et al. use the Resource Management in DiffServ (RMD) [Wes+03] architecture from Open Services Gateway initiative (OSGi) [All15] to interface to the Per Hop Reservation (PHR) and Per Domain Reservation (PDR) protocols to manage the network traffic not on a per flow basis, but on a link basis. In this way they also cover wired and wireless QoS concerns into admission control using the proposed adaptive QoS mechanism. They prove their results by evaluations in an heterogeneous test-bed.

Lee et al. [LMK07] propose an enhanced UPnP-QoS architecture to support network-adaptive media streaming in home networks. They state, that the initial UPnP-QoS architecture does not provide methods for dynamic network monitoring. Thus, they propose to enhance it by adding a dynamic network monitoring and adaptation scheme. Although the UPnP-QoS 2.0 specification [For14] introduced the GetRotameterInformation method

to retrieve network status information an other important features, it still lacks the QoS-based adaptation method and the capability of guaranteeing streaming quality over time-varying networks and flows. They propose to enhance the UPnP QoS Device with a dedicated Status Monitor component and the UPnP-QoS Manager with a QoS Adapter. Their purpose is to acquire continuous network status information for dynamic QoS management. They use this enhanced functionality to adapt the video streaming quality dynamically. This approach is different from my proposed QoSiLAN framework, since it does not aim to prevent congestion, but only to react to network performance degradation, which leads to lower video quality and therefore probably lower QoE.
Brewka et al. [Bre+11] propose an enhancement to UPnP-QoS for automatic QoS provisioning, which is a missing feature within the UPnP-QoS stack, but is also included in my proposed QoSiLAN framework. They describe the problem of auto-classification of the traffic from non-UPnP-QoS devices present in UPnP-QoS enabled networks. A limitation of their proposal is the assumption that the home gateway and all network devices are UPnP-QoS aware and support their enhancements. Only end-devices are allowed to be none UPnP-QoS compliant. An advantage is the integration of the provider network using Generalized Multi-Protocol Label Switching (GMPLS) transmission, which provides better QoS enforcement possibilities for the Internet link. Their simulation show similar results as ours with the same setup-time issues, which are inherited from their similar reactive traffic identification and classification approach.

The International Telecommunications Union (ITU) proposes an architectural framework of a home network that supports multimedia services within the recommendation H.622 [ITU08]. The ITU - Technical Work Group (ITU-T) identifies two different roles that home networks fulfil and name them as primary and secondary domains. For the primary domain the home network is considered as an extension of the access network from the provider point of view. For the secondary domain, they consider the

home network as an Intra-LAN transmission medium for data distribution among home devices from the user point of view. As an extension of the access network, they state that providers expect it to behave in a similar way to their access network with the same functional QoS services with security and management entities that can be found typically in provider networks. In the role of inter-connecting home devices these features may not be needed. For QoS they also define two different QoS frameworks – class-based QoS and session-based QoS. The session-based QoS is recommended to be realized by the UPnP solution [ITU07] and class-based QoS using the Home Gateway solution [Ini06] and the Multi-Service Delivery Framework [AGG04]. They emphasise the features of these frameworks, like for class-based QoS the reduced complexity, scalability and priority-based mechanism. For session based mechanisms they criticise that some network devices may be unaware of the signalling protocol because network devices need a complicated mechanism and that additional session set-up time is introduced by the resource reservation process. Interestingly, they also consider the NSIS Signaling Layer Protocol for Quality-of-Service Signaling (NSIS QoS-NSLP) [MKM10] and Universal Plug'n Play Quality of Service Architecture (UPnP-QoS) [For06] as emerging new QoS technologies which need further consideration. This is exactly what also the QoSiLAN framework does by further developing the ideas from NSIS and UPnP to enable autonomous session-based QoS for unmanaged networks. In this way the QoSiLAN framework complies with the H.622 recommendation for the primary as well as the secondary domain and fills the gaps in the identified drawbacks of existing and referenced solutions.

There are also QoS approaches for other layers like the routing layer. Haikal et al. propose a distributed QoS adaptive routing engine architecture based on OSPFxQoS [HBA14; Apo+99]. This is a Open Shortest Path First (OSPF) link-state routing protocol extension, which works independently of the QoS architecture. This kind of routing-level QoS architectures work well for large scale hierarchical, routed networks, but does not provide a

solution to unmanaged local networks using a single subnet, which is the target environment for the QoSiLAN framework.

The Data Distribution Service for Real-time systems (DDS) [Gro07] is a middleware architecture for devices, services and QoS management for data centric communication in highly dynamic distributed networks. It follows the publish-subscriber communication model and is able to provide QoS in any environment, where users, devices and services are potentially mobile. Al-Roubaiey and Alkhiaty provide an architecture for a QoS-aware DDS middleware in an ubiquitous environment [RA14]. Their proposed solution, as well as the DDS specification does not provide technical solutions, but only high level descriptions of solution principles and are therefore not directly comparable to the QoSiLAN framework.

Yiakoumis et al. propose to slice the home network in order to enable the service provider to control and share home network resources [Yia+11]. They introduce a slicing layer which manages the LAN resources in slices. This enables traffic isolation, bandwidth isolation and independent resource control. A service provider can request to control a slice exclusively, which requires a trust relationship between the user and the service provider. For this approach, controlling network elements like switches and access points is essential. Finally, in the work, presented by Yiakoumis et al., OpenFlow [Fou15a] enabled switches and wireless access points were used. This is a main differentiation criteria between the slicing approach and the QoSiLAN approach, which does not rely on LAN infrastructure support and emphasises leaving the control of the network to the smart distributed control of the QoSiLAN framework elements. Thus only a trust relationship with the QoSiLAN Manager within the LAN is required.

2.4 Network Topology Discovery

Network Topology Discovery is a key technology for QoS in unmanaged LANs if link based resource management is required. Without detailed knowledge about the LAN infrastructure and the inter-connection of switches and hosts, an efficient resource management is not feasible.

2.4.1 Link Layer Discovery Protocol (LLDP)

The Link Layer Discovery Protocol (LLDP) is a one-way communication protocol, based on the IEEE 802.X standards [IEE05]. It makes use of the Management Information Base (MIB) suggested by the IETF [Ros90], the Physical Topology (pTopo) MIB [BJ00]. This pTopo MIB allows for managing topology information on agents. Since the IETF didn't specify a protocol for discovering the information for the pTopo MIB, the IEEE did so. They developed the Link Layer Discovery Protocol (LLDP) [IEE05], a further development of the Cisco's Cisco Discovery Protocol (CDP) [CIS14]. Although the LLDP is a further development of the CDP, it is not compatible with it, due to its complete redesign.

The LLDP works using its own MIB. There, it stores data about local agents, discovered neighbour agents and LLDP configuration data. The LLDP sends packets, so called LLDP-DU, to all physical interfaces of a device. Any connection point in a topology is unique and therefore identified as Media Service Access Point (MSAP). Using this unique allocation of the MSAPs, a Network Management System (NMS) can build the network topology. A MSAP consists by minimum of a chassis-id, a device unique port-id and a Time To Live value. As shown in Figure 2.13, the LLDP sends its packets periodically, using one-way communication with the dedicated Ethernet multicast MAC address `01-80-C2-00-00-0E`. It works at the link layer using the dedicated Ethernet Type `0x88CC`. All information

are encapsulated in TLVs inside the LLDP Data Unitss (LLDP-DUs), see Figure 2.12. The LLDP packets are not acknowledged. The sending and the

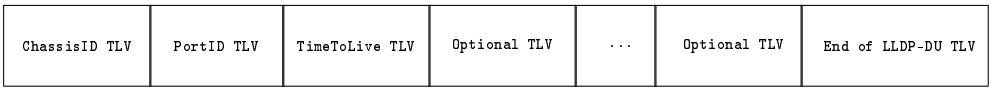


FIGURE 2.12: LLDP-DU format

receiving functionality work completely independently. Hence, the LLDP enables network devices to exchange information about themselves and their neighbour devices. The NMSs use the Simple Network Management Protocol (SNMP) to query for the topology information. An enhancement

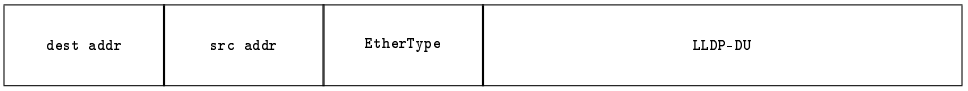


FIGURE 2.13: Ethernet MSDU format

to the LLDP is the LLDP Media Endpoint Discovery (LLDP-MED). Is was published in 2006 by the Telecommunications Industry Association (TIA) [TIA06]. It enables end-point devices to send and receive LLDP-MED packets and adds additional TLVs to the MIB, such as LAN policies for DiffServ, Virtual Local Area Network (VLAN) and layer 2 priority and Power over Ethernet (PoE) device information.

The LLDP is an efficient protocol for exchanging topology information. It allows NMSs to read the topology by using the SNMP. It is manufacturer independent and defined in open standards, in contrast to the CDP. But since the implementation effort is high and the need for a layer 3 capability in network devices, it is not likely that it will fan out in consumer products soon. Professional network infrastructure equipment and high-end SIP phones support this protocol usually, which enables for these professional environments better QoS and device management support, advanced routing intelligence and high-end administration capability. Therefore, the

LLDP is not the right choice for consumer oriented solutions, but can be advantageous if support is present. Hence, the LLDP does not fit perfectly into the QoSiLAN concept, which shall not depend on network infrastructure support.

2.4.2 Layer 3 Discovery

The Layer 3 discovery does not reveal the physical topology of a network, but rather its logical topology. Methods include reverse Dynamic Name System (DNS) queries, traceroute, pathchar discovery [Dow99] and ping scanning. Using these common tools sub-networks and domain affiliations can be discovered. Algorithms, which combine several of these methods have been developed and investigated [LLC98; SSK98; Bej+03]. These methods are intended for large scale networks with many subnets and multiple domains.

In small LANs, layer 3 discovery methods often do not reveal any useful information, since most hosts run very restrictive firewalls, blocking even Internet Control Message Protocol (ICMP) ping requests, on their LAN interfaces, nowadays. That is why these tools are often not able to discover the whole number of stations and many end-devices remain hidden. Additionally, in LANs consisting of only one IPv4 subnet, they do not reveal topology information at all. Therefore, layer 3 discovery is a useful addition to physical discovery to discover more details about a host, if available, but is not sufficient for small LAN topology discovery.

2.5 Traffic Identification and Classification

For traffic identification and classification, there exist two common general strategies. One is to configure static rules and the other is to use machine learning algorithms. According to Nguyen and Armitage [NA08], common metrics to measure the quality of traffic identification and classification precision include: False Positives, False Negatives, True Positives and True Negatives. A good traffic classifier aims to minimise the False Negatives and False Positives. The metrics are defined as follows:

- *False Negatives* (FN):
  Percentage of members of a class incorrectly classified as not belonging to the class.
- *False Positives* (FP):
  Percentage of members of other classes incorrectly classified as belonging to a class.
- *True Positives* (TP):
  Percentage of members of a class correctly classified as belonging to the class (equivalent to $100\ \% - FN$).
- *True Negatives* (TN):
  Percentage of members of other classes correctly classified as not belonging to a class (equivalent to $100\ \% - FP$).

Another commonly used metric is *accuracy*. *Accuracy* is usually defined as the percentage of correctly classified instances among the total number of instances. The machine learning literature often utilises two additional metrics known as *Recall* and *Precision*. The *Recall* is the percentage of members of a class correctly classified as belonging to the class. *Precision* is the percentage of those instances that truly have the right class, among

all those classified as the class. If all metrics are considered to range from 0 as worst to 100 % as best, it can be seen that *Recall* is equivalent to the True Positive (TP). The metrics *Recall* and *Precision* are originated from the fields of information retrieval [MRS08]. The *Recall* metric (R) is the result from (2.6).

$$R = \frac{TP}{TP + FN} \tag{2.6}$$

There, the TP results are divided by the sum of the TP and False Negative (FN) results. The R parameter expresses the actual hit rate for the achieved results. The *Precision* (P), defined in (2.7) is the fraction of the TP over the sum of the TP and False Positive (FP) results.

$$P = \frac{TP}{TP + FP} \tag{2.7}$$

The F-Measure (F) is based on the van Rijsbergen's effectiveness measure [RK79]. The F-Measure (F), defined in (2.8), combines the precision and the recall results to get a harmonic mean, which expresses effectiveness of the applied algorithm for the given results.

$$F = 2 \cdot \frac{R \cdot P}{R + P} \tag{2.8}$$

2.5.1 Classification by Static Rules

The traffic classification by static rules is the classical approach of Internet traffic analysis. In most cases it relies on the IP address and port mapping. For predefined data sets this method is capable of distinguishing different applications by analysing the TCP and UDP header parameters. This has been very successful in the past. But since current peer-to-peer protocols and various application protocols use dynamic ports and many Internet applications use the ports reserved for the HTTP and File Transfer Protocol (FTP) protocol, to enhance connectivity and firewall traversal,

this method becomes more and more ineffective [Kar+04]. For these cases, payload analysis is the only static way to distinguish these different protocols and applications, as investigated by Haffner et al. [Haf+05]. They achieve reproducible identification by searching keywords or standard character patterns and signatures used by the searched protocols. This is often called Layer7 inspection or Deep Packet Inspection (DPI). A lot of clear text protocols like the HTTP or the SIP define standard strings, which must occur at defined offset locations in the packets. One can easily parse the packets searching for these keywords. Standard character pattern and signatures can be found in binary protocols, but also in UTF-16 or Unicode based protocols, where padding bytes for the alphabet characters are common. The most frequent method to identify protocols by static rules is matching expressive pattern specifications, like regular expressions [Fri02]. This is a very well researched field, where high performance DPI is realised. Sailesh Kumar et al. [Kum+06] optimised the Deterministic Finite Automata (DFA) representations of regular expressions using the Delayed Input DFA (D^2FA). According to their evaluations, it reduces space requirements of the representations by 95%. Using this, they designed a DPI hardware architecture, which achieves real-time inspection of multi-gigabit data rates. Fang Yu et al. propose to group regular expressions in order to reduce memory usage and to increase matching efficiency. They propose to group the DFA representations by similarities like expression wild-card type or wild-card position and other conceptions. Using this approach they increased the DPI performance by factor 12 to 42.

The classification by static rules approach works very well in most cases, but searching the whole payload of packets for application signature patterns boosts the computational costs dramatically, which can be reduced by using more efficient approaches. Also, since these strategies are used in commercial applications [Sys05] for filtering purposes, special peer-to-peer protocols have started to use obfuscating methods such as payload encryption and plain-text ciphering to escape from identification, which reduces

the relevance and applicability dramatically.

2.5.2 Classification by Machine Learning Algorithms

The machine learning algorithms are powerful algorithms for traffic classification. These algorithms have been developed for applications in the fields of biology, finance, computer science and others and have the ability to distinguish protocols and applications by their communication characteristics, like the connection duration or average packet size. According to Aroussi and Mellouk [AM14], three types of learning can be identified: unsupervised, supervised and semi-supervised learning algorithms. They evaluated, that supervised and semi-supervised learning are best suited to QoS related modelling.

2.5.2.1 Unsupervised Learning

Algorithms are categorized as unsupervised learning when the observations present only the input values. Their purpose is to find similarities between these values and to group similar data into clusters autonomously [AM14]. The cluster analysis algorithms presented by Erman et al. [EAM06] are some of the most common methods for identifying classes amongst groups of objects. They are convenient for classification purposes, and are called *unsupervised learning* algorithms. This means that no training data is needed and new protocols are automatically identified. Examples for clustering algorithms include the K-Means [JD88], the DBSCAN [Est+96] and the AutoClass [Fay+96] algorithms. Although these algorithms can self-organise to building clusters, manual labelling and assigning to classes is still required. Therefore, the applicability in IP traffic identification applications is limited, since a new cluster cannot be assigned to new flows. Nguyen and Armitage [NA08] compared the three clustering algorithms

K-Means [JD88], DBSCAN [Est+96] and AutoClass [Fay+96] using both public traces and self-collected traces. Their results show that the AutoClass algorithm performs best with an overall accuracy of 92.4 %, with K-Means giving accuracy of 79 % and the DBSCAN 75.6 %.

2.5.2.2 Supervised Learning

Algorithms are categorised as supervised learning when the observations are given in the form of input-output pairs. Their purpose is to learn a function explaining the relationship between the inputs and outputs [AM14]. An example for this is the Naïve Bayes algorithm, described by Moore et al. [MZ05]. It is known as a *supervised learning* algorithm, which uses training data, with categories derived from packet-content, whereas the analysis only relies on header-derived discriminators. This makes it very efficient and precise as evaluated by Jiang et al. [Jia+07]. It works more accurately than the unsupervised learning algorithms, but is not capable of self organised learning. Roughan et al. [Rou+04] evaluated the Nearest Neighbours (NN), Linear Discriminate Analysis (LDA) and Quadratic Discriminant Analysis (QDA) machine learning algorithms to map different network applications to predetermined QoS traffic classes. They state the slow spread of QoS-use is not the lack of interest or need, but rather, the absence of suitable mapping techniques that can aid operators in classifying the network traffic mix among the different QoS classes. They refer to this as the Class of Service (CoS) mapping problem, and hypothesize that solving this would go a long way in making the use of QoS more accessible to operators. The Class of Service (CoS) mapping is performed in a three stage process: statistics collection, classification and rule creation. For classification they defined four classes: Interactive, Bulk data transfer, Streaming and Transactional. As measures they selected features by packet level, flow level, connection level, intra-flow/connection and multi-flow based properties. These classes are assigned using a Bayes classifier. Their evaluation

demonstrates relatively low error rates for their approach. Do Le Quoc et al. [Quo+15] use a Support Vector Machine (SVM) algorithm, one of the most popular supervised learning algorithms for both regression and classification problems. A SVM model [CV95] is a representation of the samples as points in a geometric space, where classes are defined as areas with clear separation. They applied it for their scalable network traffic classification solution. The SVM was selected, since it can scale approximately cubically with the number of observations in a large training dataset, as evaluated by Tsag et al. [TKC05]. Le Quoc's motivation for using SVM came from the finding, that recent DPI approaches cannot analyse encrypted or compressed traffic, which is common for their big-data cloud application.

2.5.2.3 Semi-Supervised Learning

Algorithms are categorised as semi-supervised learning when the observations are in the form of input-output pairs, but the outputs values are not known in a large number of observations. The purpose is to use the known observations to improve the recognition precision [AM14]. Usually, classification of network flows by application works using flow statistics, as presented by Erman et al. [Erm+07]. It consists of two components, the learner and the classifier. The learner discerns a mapping between flows and applications using a fully labelled training data set. Since the creation of fully labelled data sets is quite laborious, unlabelled data sets can be used. The learnt flows and applications are then used to obtain a classifier. The result is a classifier with the highest accuracy, depending on the training data set's size. Another approach for semi-supervised learning is the Statistical Protocol IDentification (SPID) algorithm, which was originally developed by Hjelmvik and John [HJ09]. It is a flexible and lightweight statistical approach to identify applications and protocols for single flows. It also uses labelled training data to learn protocols for their identification. The actual identification is based on multiple statistical measures, which

assess the protocol behaviour as well as the payload characteristic. The learned measures are stored in a database and compared to the current analysis using the Kullback-Leibler Divergence (KLD) (3.1).

Zander and Armitage [Ngu+12] presented a practical test-bed, implementing a semi-supervised learning algorithm using the Distributed Firewall and Flow-shaper Using Statistical Evidence (DIFFUSE) algorithm [WZ12]. The DIFFUSE algorithm enables the de-coupling of classification and flow prioritisation on different network nodes to distribute the computation power by duty e.g., one node for classification and one node for application and communication. The DIFFUSE algorithm performs classification decisions based on statistical flow properties, so called features and a classification model that has been trained on example traffic to recognise these features, like the eSPID algorithm for the proposed QoSiLAN framework does. The DIFFUSE is able to produce a generalised classification model for a trained traffic type. In that way the authors were able to identify first-person-shooter traffic from different applications or games reliably and distinguish it from traffic originated by other applications or games. Mainly, the approach relies on packet length and packet count statistics for identifying the flows. Their evaluations show that the authors still struggle to port the implementation on embedded devices with low computation power and memory, since the algorithm and its implementation require a lot of resources when facing high traffic throughput rates.

The area of machine learning for traffic identification and classification was presented with various approaches in the field of unsupervised, supervised and semi-supervised learning. The characteristics of semi-supervised learning algorithms were identified to fit best for the QoSiLAN framework's requirements. Especially the SPID algorithm was selected for further research as presented in section 3.3.

2.5.3 Conclusion

The field of traffic monitoring is a wide and complex research area. Whereas classification methods relying on static rules are simple to implement, machine learning algorithms work more accurate and are more flexible. Since highest accuracy is demanded to enable self-organized traffic identification on a flow basis, the semi-supervised learning algorithm SPID was selected and enhanced for the QoSiLAN solution. For the implementation of the algorithm further adjustments were made to enhance the results. The solution is described in detail in section 3.3

2.6 Bandwidth Prediction

In the past, the scientific community addressed the problem of traffic prediction mainly to continuous traffic flows, Internet backbone traffic or even more specifically on video codec level. The algorithms, designed for encoding bandwidth prediction, use algorithms to exploit the nature of MPEG video to allocate the bandwidth by scene like Sivaradje and Dananjayan did [SD02]. The algorithms used for Internet backbone prediction, address scenarios, where multiple streams run through one link and predictions aim to provide forecasts for the multiplex - the sum of streams, as Papagiannaki et al. propose [Pap+05]. In publications addressing streaming media per flow prediction, mainly traditional streaming protocols like RTP were investigated. An often addressed problem is traffic prediction on a large scale for Internet backbones [FM10], which aims on statistical predictions like number of streams, amount of bandwidth and occurrence probability.

The QoSiLAN solution requires single stream predictions in a local network scenario. Some general applicable algorithms, like the Recursive Last Square (RLS) algorithm in an application of traffic prediction [CLG95] and machine learning algorithms like SVM [Fen+06] were investigated. It was found that these algorithms are on one hand designed to predict the traffic on a short term basis and do not perform very well in an inert system like the QoSiLAN with forecasting intervals of 60 s. In addition, these algorithms cause high computation complexity, but the QoSiLAN framework's design requires a lightweight approach with a minimum of computation costs, since it is designed to run on thin and also mobile systems with limited Central Processing Unit (CPU) and power sources. That is why I aimed at a simple approach to predict the needed bandwidth as accurate as possible. These requirements are mostly met by linear prediction algorithms. He et al. [HDA05] distinguished between Formula Based (FB) and History Based (HB) algorithms. For the FB algorithms they propose linear

prediction algorithms. Among others a Moving Average (MA) predictor was presented. I also employed and configured it for my application and referenced it as the Mean Estimation (ME), for my evaluation of results. The HB algorithms do not fit my application, since they require a large set of throughput measurements from previous transfers on the same path, which behave similarly. Especially, the similar behaviour of protocols is an assumption, I couldn't reproduce in my evaluations, where similar video streaming flows from different large Content Delivery Networks (CDNs) behaved very individually.

2.7 Resource Management

Admission control should not be present in the Internet. It should be based on the principle of net neutrality [Wik16], which means an equal treatment of digital content. Resource management for QoS applications require a policing and admission control schema to manage incoming resource requests efficiently. In this way QoS resource management breaks with the principle of net neutrality to prioritise some services or flows over others. This is needed to protect sensible content/flows from congestion and distortion. A principle, which should not be harmed in the global Internet can be handled differently in a local environment to enable a better QoE. There, different treatment of services and flows might be based on an agreement of the local users.

2.7.1 Policing and Admission Control for Resources in LANs

A framework for providing end-to-end QoS for individual flows was proposed by M. Yang et al. [Yan+03] with the goal of keeping the scalability of the DiffServ framework. They propose the On-Demand QoS Path framework (ODP), which supports per-flow admission control and end-to-end bandwidth reservation. In contrast to my QoSiLAN framework, the ODP framework targets inter-domain/Internet QoS by involving edge and core-routers. The ODP framework enables scalability by using class-based service differentiation in the network core. The ODP framework reduces the signalling effort by using a hierarchical bandwidth management scheme. From evaluations they conclude that ODP's central control and router-aided approaches provide end-to-end guarantees to individual flows with significantly less overhead than IntServ based QoS solutions like RSVP.

In terms of admission control, one can distinguish between Parameter-Based Admission Control (PBAC) and Measurement Based Admission Control (MBAC) algorithms. Whereas the PBAC algorithms rely on à priori knowledge and accurate network traffic models to allot resources, MBAC algorithms rely on actual measurements and accurate estimation of QoS parameters. Brewer and Ayyagari [BA10] compare and analyse MBAC and PBAC algorithms in test-bed evaluations. They conclude for bursty traffic patterns that the MBAC approach provides better network utilisation and a higher admission rate than the PBAC approach. Similar results from Mancuso and Neclia [MN02] prove true the superiority of MBAC algorithms above PBAC algorithms, in particular for scenarios encompassing the bursty nature of self-similar flows. They discovered, that MBAC algorithms make the system robust to statistical traffic properties. That's why I also chose to investigate more on MBAC algorithms and designed my approach according to this scheme. Moore [Moo02] identified five characteristics for an appropriate MBAC algorithm. First a MBAC must provide a relationship between the traffic characteristic and the calibration control. Second, the estimator must incorporate the statistical nature of traffic. Third, the estimator and the MBAC must be matched to the task required. Fourth, the algorithms must be implementable with realistic resource requirements. Fifth, the policy, performed by the MBAC influences the overall performance critically. Overall, he concludes, that the correct maintenance of the current provision values is more important than the accuracy of short term traffic characterisation. Independently, I also designed my MBAC algorithm for the QoSiLAN framework according to these principles and share the experiences. Jamin et al. [JSD97a] evaluated three MBAC and one PBAC algorithm in terms of the performance of a controlled load service. They configured the PBAC algorithm for capacity bounding. The three MBAC algorithms are based on equivalent bandwidth, acceptance region and measured bandwidth. Although they do not aim at giving final conclusions on their simulation results, their evaluations

reveal that a higher utilisation target than 80 % causes packet loss in the network. In another survey Jamin and Shenker [JSD97b] observed, that all known MBAC algorithms can be reduced to one formula, as shown in (2.9),

$$\hat{\nu} < f(\cdot)\mu - g(\cdot) \tag{2.9}$$

and be tuned with parameters to give the same result curves. In (2.9) $\hat{\nu}$ is the measured load, μ is the link bandwidth, and $f(\cdot)$ and $g(\cdot)$ are functions of the source's reserved rate and the number of admitted sources. Therefore, they conclude and propose to focus future research on the tuning parameters, instead of the algorithms itself. Another observation is the structural limitations of MBAC algorithms. First, long lasting connections will statistically dominate the reservations over short lasting connections. Second, flows that traverse multi hop paths have a higher risk of a rejected admission, if the switches perform admission independently. For the QoSiLAN's MBAC algorithm I considered the limitations and found solutions as described in section 3.6. The application of MBAC algorithms in the context of QoS and Quality of Experience (QoE) was shown by Latré and De Turck [LD13]. The authors propose a MBAC algorithm for provider-based video rate controlling. They define policies that can use MBAC algorithms and video rate control policing for the goal of revenue maximization or QoE. This is a passive approach to react to QoS degradation. Instead, the QoSiLAN framework aims at preventing congestion and interference traffic actively. Liu et al. [LLG14] propose a generic admission control methodology for heterogeneous wireless and wired networks. Their approach is based on a QoS index, which is derived from various connection parameters that characterize the traffic information and their performance requirements, like the number of connections, packet delay requirements or throughput requirements. The proposed admission-control algorithm decides whether or not to admit new connections by estimating the potential impact of the new connection on the QoS index by a mathematical function using the Taylor approximation [Var14]. The values for the Taylor

approximation are estimated and computed by real-time measurements. The network performance is tracked using the QoS index by predicting the potential impact of a new connection admission, based on the index. This approach has some similarity to the QoSiLAN approach, which also analyses the network performance continuously to predict the resource requirements for a connection or flow. In contrast to the methodology from Liu et al., the QoSiLAN framework only measures the throughput parameter, which results in less complexity. The ETSI distinguishes two general resource admission control approaches [ETS08] for their Resource and Admission Control Sub-System (RACS). The *Independent Resource Admission Control* approach is defined as an admission method where a single Functional Entity (FE) controls the resources independently. The *Coordinated Resource Admission Control* approach is defined as an admission method were multiple Resource and Admission Control Functions (RACFs) coordinate the admission control for resources to avoid uncontrolled overbooking. According to the ETSI's definition, the communication shall be organised in a hierarchical way. For the QoSiLAN framework the *Coordinated Resource Admission Control* approach was selected with the QoSiLAN Manager (QM) as a top-tier RACF and the hosts as lower-tier RACFs. In contrast to the ETSI approach the QoSiLAN framework does not give full resource control of a sub-domain to the lower-tier RACFs in a hierarchical way. It utilises them to gather information about the sub-domain and the top-tier RACF makes the resource control decisions, which are then executed by the lower-tier RACFs.

Chapter 3

Research Solution and Evaluation

This chapter describes the research solution and all the contributions in detail. Beside the QoSiLAN framework itself, also the investigated and improved components it consists of are presented. The contribution this thesis presents is the concept of the QoSiLAN framework as well as an investigations into the link layer topology discovery and the enhancements and improvements to statistical protocol identification. Furthermore, a new statistical class based bandwidth prediction algorithm and a new QoS protocol are contributed and presented in detail. Finally, the contributed policing and admission control algorithms, which integrate all the other components and define the QoSiLAN's concept are presented with their respective contributions.

To improve the readability, the methodology as well as the evaluations of the individual components are presented within the sections where they are presented. At first the QoSiLAN framework is described in general to introduce the framework integration of the used components. Then the Link Layer Topology Discovery (LLTD) protocol and its implementation

is introduced for its novel usage as data source in QoS resource policing and management, including proof of concept evaluations. Next, the Enhanced Statistical Protocol IDentification (eSPID) algorithm is presented along with elaborated evaluations including algorithm calibrations. Afterwards, the Statistical Class Based Bandwidth Prediction (SCBP) algorithm is specified and evaluated within the QoSiLAN framework for predicting resource requirements for individual flows. Then, the NSIS QoS Signalling Layer Protocol for Local Area Networks (QSLP-LAN) is presented to enable QoS signalling, including quantitative evaluations highlighting scalability issues. Finally, the proposed policing and admission control algorithm is presented. This section includes a complete proof of concept evaluation, involving all QoSiLAN framework components to prove the whole QoSiLAN framework integration that all individual components operate satisfactorily.

3.1 The QoSiLAN Framework

The QoSiLAN framework aims to provide QoS for unmanaged heterogeneous LANs in a self-organised manner, while not relying on network assistance. When naming QoS in this context, only the QoS parameter of bandwidth reservation is meant. The goal is to reserve bandwidth for single links along the path within the LAN for a single defined flow without infrastructure assistance. Since the network is assumed to be not QoS supportive, QoSiLAN cannot force switches or routers to preserve bandwidth on attached links. Consequently, the end-hosts in the network need to manage the bandwidth resources in the LAN cooperatively - without network assistance. The reservation of bandwidth capacity is achieved by cooperative bandwidth shaping of best-effort traffic to keep enough residual bandwidth for the flow with a QoS demand. All hosts in the network need to cooperate in such a system and control their outgoing traffic in a smart, managed manner. To enable this QoS framework, several key components

are required, as depicted in Figure 3.1 and described in the remainder of this section.

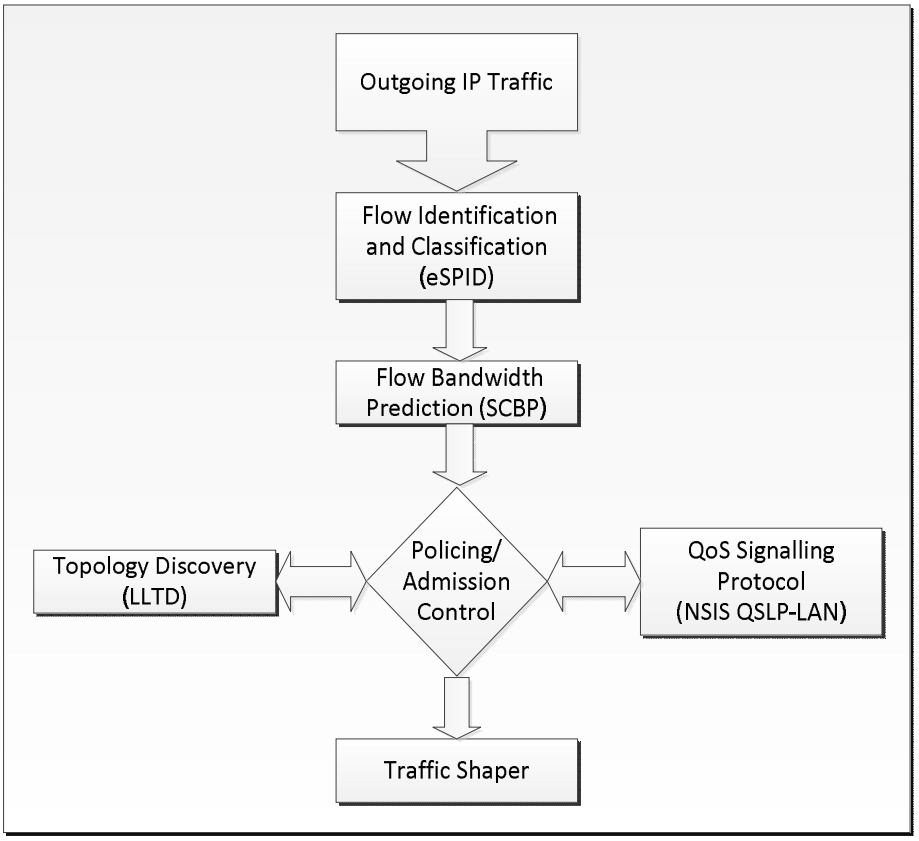


FIGURE 3.1: Overview of the QoSiLAN key components

To achieve autonomous QoS in unmanaged hybrid networks, the hosts need knowledge of the LAN's link layer topology. It is essential for the QoSiLAN framework to exactly know which host is connected to which switch, how the switches are connected with each other and which hosts are connected to a wireless access point. On top of this, the nature of the links e.g., their simplex or duplex characteristic and their capacity is of major importance to manage the LAN's resources efficiently.

To develop the QoSiLAN's self-organisation capability, a decoupling from application support is required. This feature enables wider applicability and

better interoperability when implemented in a consumer environment. To provide independence from application support, the traffic in the network must be monitored and analysed and classified at the flow level to detect and maintain QoS requirements automatically.

When a flow with real-time QoS requirements is successfully detected, in an autonomous QoS system the QoS demands for the flow must be known a priori. Therefore, the resource requirement for an identified flow must be estimated to enable the resource reservation in the LAN.

The hosts in the network must communicate within each other to arrange the QoS requirements of identified flows and to manage the residual bandwidth in the LAN in a cooperative manner on a per-link basis. Therefore, a QoS protocol is needed, which enables cooperative QoS communication and sharing of the relevant QoS information within the hosts.

The host cooperation not only requires a common communication, but also for common policing and admission control to ensure a consistent QoS management. Therefore, one host in the network is selected as QoSiLAN Manager (QM). The QM's responsibility is to gather all information from other hosts in the LAN and to maintain a complete topology map. It works as central policing entity, which decides on the admission of QoS reservation requests from the other hosts.

In the following sections, the previously mentioned required components of the QoSiLAN framework are presented and evaluated.

3.1.1 Methodology

This section describes the general methodology to asses the proposed QoSiLAN framework. It describes the test-bed with the used network infrastructure the evaluations are based on, as well as the software architecture, that

the implementation uses. All components where evaluated using the QoSi-LAN test-bed and analytical calculations. The detailed research approaches and test-bed configurations are presented in their respective sections. The framework integration was proved in the policing and admission control evaluation in section 3.6.2.

3.1.1.1 Home Scenario Evaluation Testbed

This section presents the components and devices the test-bed consists of as well as the the network infrastructure.

3.1.1.1.1 Network Infrastructure

The evaluation test-bed was selected to represent a typical home environment with six active hosts, as presented in Figure 3.2. Three of the hosts are

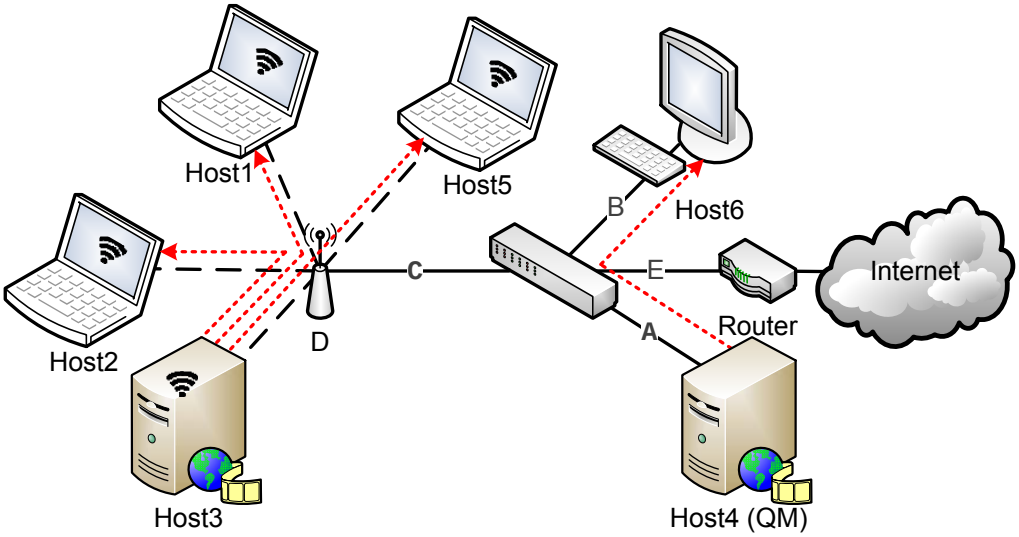


FIGURE 3.2: QoSiLAN Evaluation Scenario

connected using fixed line 100Base-T Ethernet links and three hosts are connected using WiFi IEEE 802.11g links. In particular, the simplex/duplex

nature of the different link types and the hybrid QoS behaviour are represented by this setup. Each wireless and wired host is serving as media server and therefore as a data source. The others are configured as media clients to consume media and to demand resources interactively. We configured Host4 as QoSiLAN Manager (QM) to manage the resources, since the intra-LAN traffic is in focus. For setups where the Internet to LAN traffic is the focus, the router is configured as the QM. The hosts are Netbook devices with the Windows 8 operating system. The router is a Linksys WRT-54GL device [Wik15] running the Linux based DresDen-WirelessRouTer (DD-WRT) operating system. All devices, including the router, are equipped with the portable QoSiLAN framework stack. The test-bed network is isolated from external traffic using the router's NAT and firewall functionality. The Internet link is routed through a LAN, sharing a 100 Mbit Internet link, provided by the facilities of the Technische Hochschule Mittelhessen (University of Applied Sciences Mittelhessen), Germany (THM), which is connected to the Deutsches Forschungs-Netzwerk (German Science Network) (DFN) backbone [Net15]. The DFN Internet backbone X-WIN is a science network, connecting more than 60 Universities, science institutes and science related companies within Germany, Europe and abroad using one of the most powerful fibre based communication networks in the world.

3.1.1.1.2 Software Architecture

The QoSiLAN Manager and QoSiLAN Client implementations were build as depicted in the component diagram and presented in Figure 3.3. The client operating mode can be explicitly defined using command line parameters or the configuration file. It consists of two software modules. The LLTD Mapper Daemon and the QoSiLAN Traffic Manager Daemon. The first is responsible for mapping the LAN's topology. It provides an HTTP listener service to react to mapping requests. The LLTD Mapper Daemon communicates using the LibPCap library installed within the operating

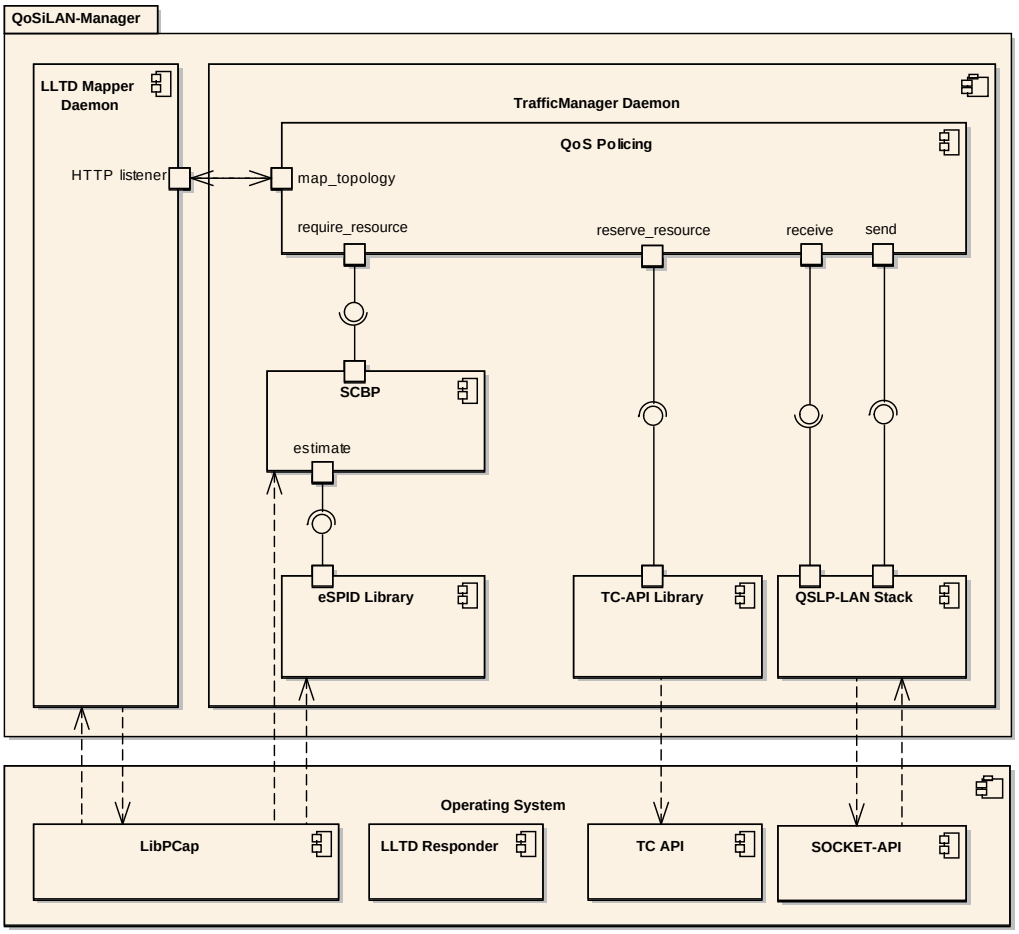


FIGURE 3.3: The QoSiLAN Client Software Architecture

system. After the mapping process finishes, the topology is provided using the same TCP connection in Extensible Markup Language (XML) format. The second, the QoSiLAN Traffic Manager Daemon, requests the topology map initially after start-up and every time the eSPID traffic identification library detects a new host in the network. The eSPID library component monitors the traffic continuously. Once a relevant flow is detected, its call-back is caught by the estimate function within the SCBP library module, which analyses the first ten seconds of the flow and calls a resource_request function within the QoS Policing module. The QoS Policing module holds

the LAN's topology information and decides about the admission of the resource request using an implementation of the QoSiLAN's admission control algorithm described in section 3.6.1.2. The QoSiLAN's signalling procedure is also orchestrated by the QoS Policing module. Messages are sent and received by the QSLP-LAN Stack module, which encapsulates and decapsulated the signalling information according to the NSIS protocol recommendations, as presented in section 3.5. The TCP/IP packets are exchanged using the Operating System (OS)'s Socket API. The Socket API and all Threading API is wrapped for OS independence using the ADAPTIVE Communication Environment (ACE) [Sch13]. The ACE is a wrapper library, that encapsulates all OS specific functionality like multi-threading and network socket communication, to provide a common programming API. Local QoS policies are applied using the OS's traffic control API, wrapped by the TC API Library module. Network throughput tests are run using the iperf server, which runs on all QoSiLAN nodes. For applying the QoS policies, on Linux based operating systems the Netfilter's `tc` command is used, whereas for Windows based operating systems the Windows TC-API is used. The Microsoft LLTD responder daemon is required to run on all hosts in the network, since the QoSiLAN framework communicates indirectly to it, when it runs to the topology mapping process.

3.2 The Link Layer Topology Discovery

In order to manage the traffic in the LAN on a per-link level efficiently, knowledge about the network's link layer topology is essential. The Ethernet's Media Access Control (MAC) protocol does not provide this functionality. In addition, layer 2 Ethernet switches do not offer an interface to access the switch's Address Information Table (AIT). To guess a switch's AIT one can employ the allowed functionality to spoof MAC addresses in the LAN to make the switch learn new MAC addresses and watch the effect in packet forwarding. This principle is addressed by the Link Layer Topology Discovery (LLTD) protocol. The LLTD technology was inspired by research from Richard Black et al. [Mic04], who implemented it as the LLTD protocol for the latest Microsoft operating systems. This protocol enables link layer discovery of hosts, switches, hubs, access points and routers and their inter-connection in a local area network. Up to now, the application of LLTD was limited to mapping a LAN and to identify network configuration problems. The QoSiLAN framework employs this protocol in the context of a novel framework of self-organised QoS, bandwidth discovery, path finding, and resource reservation. The mentioned Microsoft Windows implementation does not provide any API to start or employ the mapping process automatically. Additionally, the resulting map cannot be exported or exchanged with other programs. The only public available implementation is the LLTD responder's source code for Linux operating systems. Therefore, the mapping process was reverse-engineered and implemented to make use of this technology. In the reverse-engineering process two documents provided information to facilitate the work. The first document is a conference paper by Richard Black et al. [BDF04], which reveals the principle of the mapping algorithm. Second is the LLTD protocol specification, which describes the protocol design and the LLTD packet header configuration. In the field of link layer topology discovery no scientific contribution was made by me. The investigations and research on this topic

where necessary to get an integrated system with the provided functionality. In the following the algorithm and the implementation are explained in detail to equip the reader with the knowledge of this key enabler, which is necessary to understand the QoSiLAN framework as a whole.

3.2.1 Algorithm

The LLTD protocol works for wired IEEE 802.3 and wireless IEEE 802.11 networks. It is based on probing packets. A central entity, called the mapper, sends to each LLTD supporting host, called responder, in the network. All end-devices, which implement the protocol appear in the network topology map the mapper builds. The protocol aims to investigate the address tables of Ethernet switches. Since one cannot read the AIT from most consumer switches, the mapper sends Emit packets to the hosts to make them send probe packets to the switches using faked MAC addresses, according to the connection reasoning technique presented by Yantao Sun et al. [SWS05]. Using this algorithm, it trains and probes each switch's AIT to find out if the hosts are attached to the same or different switches. This works, because according to the IEEE 802.3 specification a switch forwards packets with an unknown MAC address to all ports. Once the switch received a packet with the MAC address as source it forwards incoming packets with this MAC as destination address only to this single port. The other hosts in the network, which have also listened to the training packet also answer the mapper if they also have received the packet or not. For this purpose the LLTD uses MAC address spoofing with addresses starting with `00-0D-3A`. This is a dedicated MAC address range, reserved by Microsoft, for the purpose of LLTD network mapping.
Figures 3.4 and 3.5 show sample scenarios to visualise the probing algorithm. Whereas, in Figure 3.4 both hosts are connected to the same switch, in Figure 3.5 the hosts are connected to different switches, but the two switches are inter-connected. The sequence diagrams in Figures 3.6

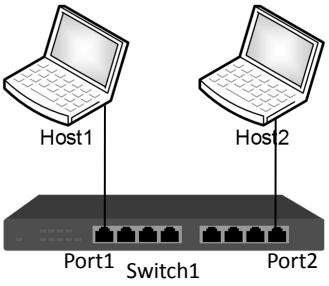


FIGURE 3.4: LLTD Scenario With One Switch

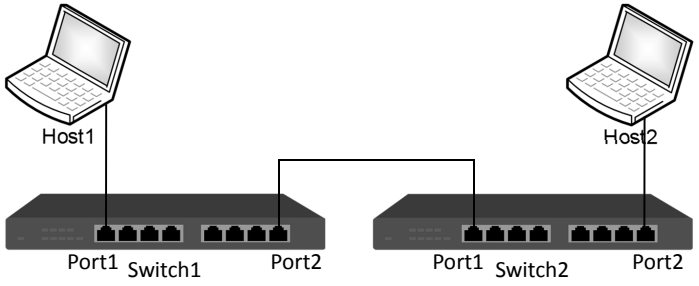


FIGURE 3.5: LLTD Scenario With Two Switches

and 3.7 show the difference in LLTD protocol communication this causes. For clarity, the Mapper's Emit and Query packets are not visualised, but only the Probe packets the Responder sends. One can assume that all of these Probe packets are caused by respective Emit packets from the Mapper to define the address configurations for each Probe packet. Afterwards the Mapper sends Query packets to ask for the 'Sees'-List from the Responders.

In the first case using one switch, Host1 sends a probe packet to Host2's MAC address, which is forwarded to the Host2. From this observation, the switch assigns the Host1's MAC address to the switch's Port1. Host2 sends a packet to Host1 to make the switch assigning Host2's address to its Port2. Next, Host1 sends a packet from an unused address X to it's own address, which makes the switch to learn X to Port1, but does not forward it. Afterwards, Host2 also sends a packet using the same address

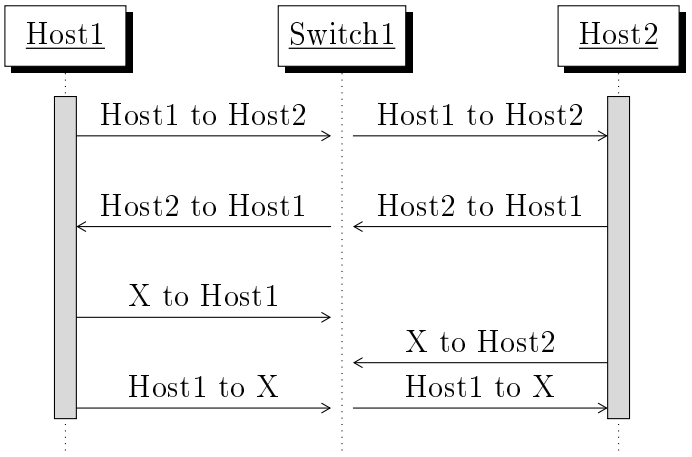


FIGURE 3.6: LLTD Sequence With One Switch

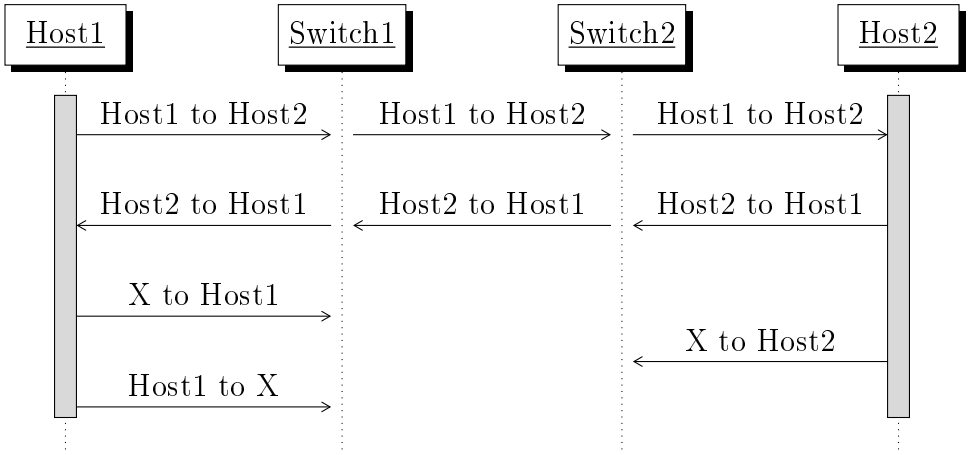


FIGURE 3.7: LLTD Sequence With Two Switches

X as the source to itself. This causes the switch's AIT to change the entry of X from Port1 to Port2. Finally, Host1 sends a packet to address X, which is successfully forwarded to Host2. This happens in contrast to the communication in Figure 3.7 with two switches, where the learning of X happens in different switches. Since there the address X is still a valid entry for Port1 in the Switch1's AIT, the message targeted to X from Host1 is not forwarded, but dropped at Switch1 and therefore does not reach Host2.

Therefore, this packet will not appear in Host2's 'Sees'-list, which will be queried by the Mapper.

As presented in Figure 3.8, the LLTD message stack follows a layered approach, based at the link layer level, using the dedicated Ethernet type `0x889D`.

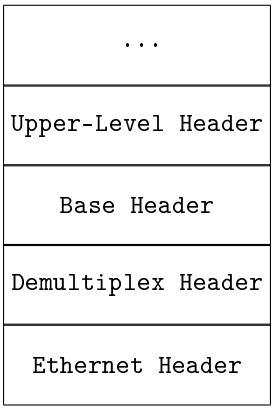


FIGURE 3.8: Position of LLTD Protocol Packet Headers

The Demultiplex Header defines fields for the LLTD protocol version, the type of service (ToS) and the function of the packet. The Base Header contains two 48 bit address fields and a 16 bit identifier. The meaning of the fields depend on the ToS values and the function fields in the Demultiplex Header. The mapping engine works in different states, where particular messages are sent. For the upper-layer header multiple header formats are defined, depending on their function and the protocol state:

- Used in the Command and Emit states:
  - Reset: Sent by the mapper to the broadcast address to reset all LLTD sessions of all responders.

  - Discovery: Sent by the mapper to the Ethernet's broadcast address to encourage all LLTD responders to send Hello packets and to initiate a new discovery session.
  - Hello: Sent form the responders to the broadcast address in response to a discovery frame. Used to announce information about itself.

- Used in the Command state:

  - Emit: Sent from the mapper to a responder to make it send packets with spoofed MAC addresses. An Emit frame is a list of source and destination MAC addresses that is prefixed by number of milliseconds to pause before sending a frame.
  - Charge: Sent from the mapper to a responder to charge credits for Emit frames.
  - ACKnowledgement (ACK): Sent from the responder to the mapper to acknowledge Emit and Charge frames
  - Flat: Sent from a responder to the mapper if there is not enough credit for sending Emit frames.
  - Query: Sent from the mapper to a responder to request a QueryResp frame.
  - QueryResp: Sent from a responder to the mapper in reaction to a Query message. It lists information the responders have observed since the last Query message.
  - QueryLargeTlv: Sent from the mapper to a responder to request a QueryLargeTlvResp frame.
  - QueryLargeTlvResp: Sent from a responder to the mapper. It lists information using large Type Length Values (TLVs).

- Used in the Emit state:

– Probe: Sent from a responder to spoofed MAC addresses. Responders whose topology state engine is in the Command or Emit state add Probe frames that they receive to their 'Sees'-list, noting the Probe's Ethernet source and destination addresses and real source address from the base header.

– Train: Sent from a responder to the spoofed MAC addresses. Train frames are discarded by responders and are only used to train switches.

The discovery process proceeds in several phases, which were implemented in the LLTD mapper application. The LLTD mapper application offers the possibility to communicate either through named pipes with the network simulator or using the winpcap[CAC]/libpcap[NRG] library with the real LAN. Figure 3.9 shows the user interface of the LLTD mapper application. It contains the option to switch the network simulator interface from the named pipes to WinPcap. In addition, there is an area to follow the algorithm actions, similar to a debug output. On the right, there is a tree view, which shows the discovered segment tree. The Address Resolution

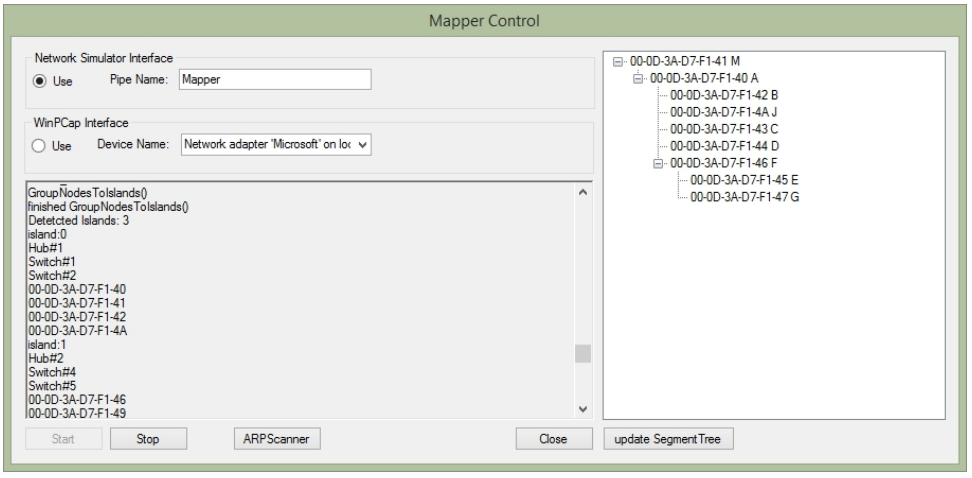


FIGURE 3.9: LLTD Mapper Application with the Test Topology's Segment Tree

Protocol (ARP)Scanner button enables a feature to listen to ARP packets from hosts in the LAN, without LLTD responder support. These are added to the topology loosely, since they cannot be located and assigned to particular segments.

3.2.1.1 Discovery Phase

This phase is used to discover the LLTD responder enabled hosts in the network, to gather their properties and to initialise the variables and lists within the responders memory. At first, the mapper broadcasts Reset frames into the LAN, to indicate a new discovery session to all responders. Next, discovery frames are sent using Ethernet broadcast messages. These are answered by all LLTD responders with Hello packets, again to the broadcast address. Hello packets contain TLVs with all available and relevant information about the responder hosts e.g., their interface type, machine name, link speed, IPv4 and IPv6 addresses and the operating system.

3.2.1.2 Segment Detection

This phase is used to detect segments in the LAN and to create a segment tree. A segment is a part of the network, where connected host are able to monitor each others traffic, like as a 10BaseT Hub. Once the discovery phase is complete, the mapper sends Charge and Emit packets to the Ethernet hosts to make them probing the switches. The Charge frames load credits to the responders to allow the mapper to make the responders send probe packets, by accepting Emit frames. Charge frames are needed for security reasons, to avoid network flooding from malicious mappers or attackers. The Emit frames advise the responders to send probe frames with source and destination addresses the mapper dictates. Once a responder accepts an Emit frame, it sends an ACK frame to the mapper and a

Probe frame to the dictated destination. In response, the mapper sends a Query frame to the responder it sent the Emit and Probe frames to, to receive the Probe's results. The results are sent using QueryResp frames. QueryResp frames contain a so called recevee descriptor list. Every responder maintains a 'Sees'-list, which stores information about Probe frames it has listened to in promiscuous mode during the running discovery session. The sees-list consists of Recevee-Descriptors. The mapper compares the 'Sees'-lists it received from the responders. Wireless nodes, which are connected to the same access point identify themselves during the discovery phase with the same Basic Service Set IDentification (BSSID) value. For that reason, they can be grouped easily within one segment, without testing. Responders with the same sees set are regarded as being within one segment. Responders with a sees subset are located below in the segment tree. In that way the responders are sorted in the segment tree according to the depth-first walk algorithm [Wik13].

3.2.1.3 Island Detection

Nodes with seamless connections without gap are grouped into one island. In the case where multiple ones are detected, the edge switches and hubs are located using the path crossing test algorithm as described by Richard Black et al. [Mic04]. This can detect if deep switches or hubs are located within the gap. Deep switches or hubs are devices, which have no LLTD enabled host connected directly. The path crossing test finds connections between islands by sending probe messages from one island edge node to another island edge node. From the sees sets of the island's nodes the LLTD Mapper decides on the existence of a deep switch/hub or a direct connection between the islands.

3.2.1.4 Finalising

Once the discovery process is finished, the mapper broadcasts Reset frames to indicate the discovery session's end to the responders.

3.2.2 Methodology

The Link Layer Topology Discovery (LLTD) protocol was selected after an intensive related work study, as presented in section2.4.1 and a comparison of existing topology discovery solutions. It was found that the LLTD algorithm is the most appropriate state of the art approach to physical topology discovery in LANs. The protocol was implemented by Microsoft for their Windows operating system products since the release of Windows Vista, including a closed source LLTD Mapper service and a LLTD Responder service for Microsoft Windows and an open source LLTD Responder for Linux based operating systems. However, the most important part, an API to the LLTD Mapper service or an open source implementation of it are not available publicly. Although technical protocol descriptions exist and the algorithm was presented within a conference paper [Mic04], a lot of implementation and algorithmic details of the LLTD mapping process are not published. So, to be able to use the technology within the QoSiLAN framework and for its evaluations the LLTD mapper had to be reverse engineered and re-implemented. This was a major task, since the mapping process is very complex. Especially discovering deep segments and hosts is not trivial, as pointed out in section 3.2.

To reverse engineer the LLTD algorithm and to test the LLTD mapper during the implementation and reverse engineering phase, an Ethernet network simulator was developed, which simulates the behaviour of switches, hubs and LLTD responder nodes. The network simulator, as shown in Figure 3.10 provides basic Ethernet functionality to emulate the Ethernet

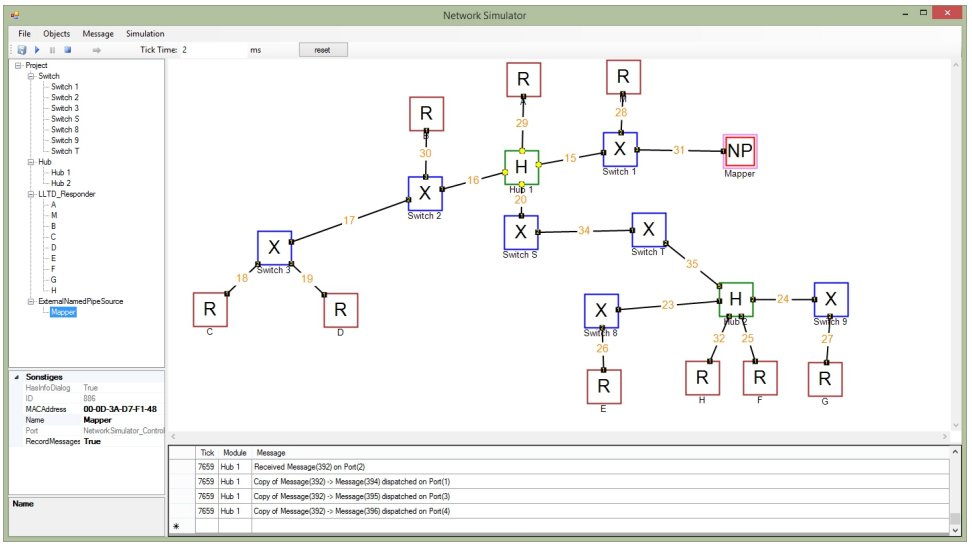


FIGURE 3.10: LLTD Network Simulator with the Test Topology Loaded

communication behaviour, addressing and the switch's Address Information Table (AIT) building behaviour. In addition, the LLTD responder nodes also implement the LLTD responder behaviour for LLTD message sending and responding and the 'Sees'-list. All addresses, tables and lists are inspectable through the user interface. Additionally, the network simulator provides functionality to pause and continue the communication to provide rich debugging possibilities. It provides a live watch feature, to follow the Ethernet packet traversal through the LAN using animations. The LLTD mapper is a separate component, which was developed to support both communication with the simulator through named pipes and Ethernet operation using the libpcap/winpcap API interfaces [NRG; CAC]. The LLTD mapper as well as the Ethernet Simulator were developed using the Microsoft .NET framework [Cor02] and the Mono project framework [Pro04] to support platform independence. For the other evaluation scenarios, which make use of the LLTD features and as a final regression test, the LLTD mapper was tested and productively used within the evaluation test-bed, as presented in Figure 3.2.

3.2.3 Evaluation

The LLTD mapper application provides several output mechanisms. First, it supports a socket interface, which provides XML formatted data to other applications and services. A TCP message containing the 'LLTD' string causes the application to start the discovery's process. When the discovery finished, the resulting topology is sent back using the same TCP connection. In addition, the XML output can be saved into a file. On top of that, the topology can also be converted into the DOT language format, which can be rendered into a Scalable Vector Graphics (SVG) [Fer+11] file using the Graphviz's [Gra13a] DOT tool [Gra13b], as shown in Figure 3.11. This output is an easy way to visually prove the correctness of the mapping process. As one can see, the topology, loaded in the network simulator,

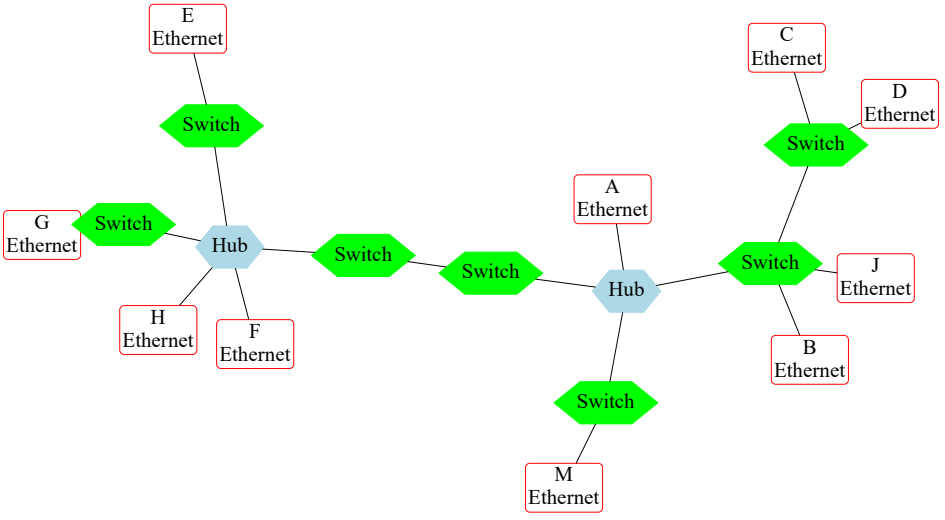


FIGURE 3.11: LLTD Mapper Result from the Test Topology, Rendered in SVG

shown in Figure 3.10, matches with the topology in the resulting SVG file, rendered from the DOT structures generated by the LLTD mapper application as depicted in Figure 3.11.

3.2.4 Conclusion

The LLTD protocol algorithm uses a complex algorithm to deduced the link layer topology from sent and received probe messages. In addition, the protocol provides important information about the hosts in the network, like the link's speed and its duplex properties. The reimplementation was proved by evaluations to provide the link-layer network topology reliably in an self organized manner. The output topology is provided in an human readable, graphical way, as well as being machine readable in XML format through a socket connection.

3.3 The Enhanced Statistical Protocol Identification

In order to provide self-organisation and thus software application independence, the QoSiLAN framework needs to identify flows to reserve resources automatically. This is achieved by enhancing the approach for Statistical Protocol IDentification (SPID) to identify real-time-audio and video-traffic protocols with QoS demands independently, even for encrypted, compressed or tunnelled payload accurately in a lightweight manner.

3.3.1 Algorithm

The SPID algorithm was originally developed by Hjelmvik and John [HJ09] and is a statistical approach to identify applications and protocols for single flows. There is no need to search for unique application signatures and in addition near real-time detection after the 10^{th} packet is possible. Both features define the advantages of this kind of technique. The SPID algorithm works in three steps as presented in Figure 3.12. We enhanced the SPID

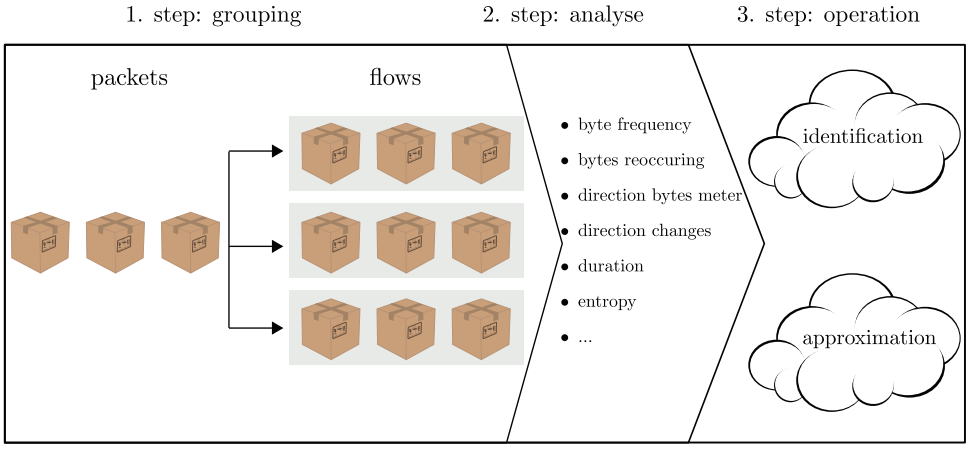


FIGURE 3.12: The Phases of the SPID Algorithm

algorithm by reimplementing it in a more efficient way and by using an optimised set of statistical measures. The algorithm and the phases remain as defined by the original SPID schema.

3.3.1.1 Phase 1

In the first phase all packets are grouped into bi-directional flows. In the case of TCP, a flow starts with a three-way-handshake. A flow is identified by source IP and port, destination IP and port and the transport protocol, called the five-tuple. Packets without a payload, like TCP/ACK packets are skipped and not analysed. When a flow group consists of more than 20 packets, the second phase starts working. Through evaluations, presented in Figure 3.14, I found, that the accuracy of flow detection reaches a stable level above 80 % after 20 packets of observation.

3.3.1.2 Phase 2

During the second phase the actual statistical analysis starts. It is based on Kullback-Leibler Divergence (KLD) [KL51], as shown in (3.1).

$$D(P||Q) = KL(P,Q) = \sum_{x}^{N} P(x) \log_2 \frac{P(x)}{Q(x)} \tag{3.1}$$

The KLD is a logarithmic measure denoted by D in (3.1) for the difference between two probabilities. It compares the accumulated probability values of N measures (Q) with trained flows (P). In the case where two probabilities are equal ($P = Q$) the result is $D(P||Q) = 0$. For all other cases the result is $D(P||Q) > 0$. The KLD does not behave symmetrically $KL(P,Q) \neq KL(Q,P)$. From an information theory point of view, the KLD expresses how much additional information is needed to describe the value of (P), if a code is optimised for (Q). During phase 2 the KLD is

used to compare the measured flows with the obtained measures, which are stored in a database. For each of the trained protocols (P_i) the KLD is calculated. The best match is the protocol with the lowest divergence. Through extensive evaluations I found 12 measures, which provide good statistical measures for a stable protocol identification. These measures are presented in the following.

- **Byte-Frequency**: The byte frequency measure is applied to the first data packet of a connection. It counts the relative occurrence of byte values. It can be visualised using a histogram graph with 256 bars, as shown in Figure 3.13. From this kind of measure, plain text protocols

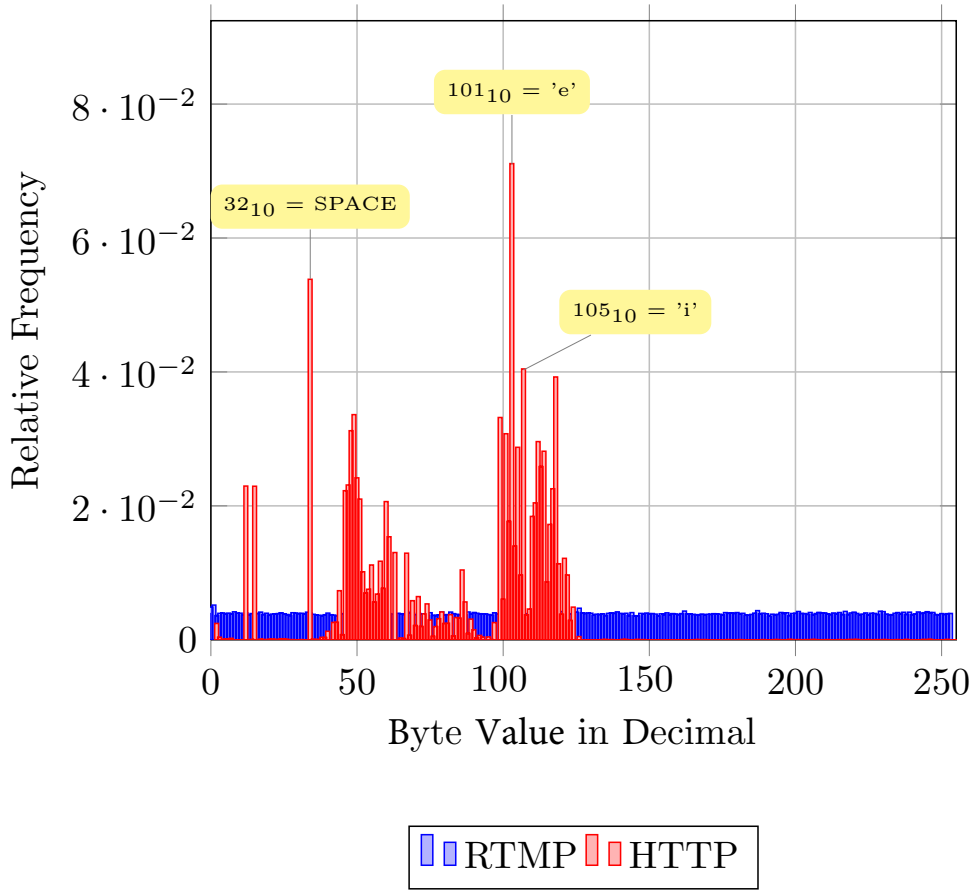


FIGURE 3.13: Byte Frequency Histogram

like HTTP, Internet Relay Chat (IRC) or Simple Mail Transfer Protocol (SMTP) are readily identified, since they contain re-occurring character sequences, which lead to an uneven distribution. An example of this is the HTTP protocol family, whose packets always start with key words like 'GET', 'POST' or 'HTTP/1.1'. Encrypted connections have a very flat histogram, since data uses all byte values more or less equally, which leads to an even distribution.

- **Byte-Frequency of the first 32 bytes**: This measure follows the same approach as the *byte-frequency* measure with the differences, that only the first 32 byte are recognised. It is needed to handle protocols with small UDP packets which contain no plain text information. In this case the resulting divergence can be reduced in contrast to a full packet analysis.

- **Count of Direction Changes**: The count of direction changes measures how often the source and destination addresses swap within the flow group. For n packets in the flow group $n-1$ direction changes are possible. Interactive protocols like Telnet, FTP or Secure SHell (SSH) have significantly more direction changes than streaming protocols like RTP or MMS. Hence, this measure provides a statistical means of differentiating between interactive and streaming protocols.

- **Direction bytes Meter**: This metric measures the bytes, transferred in each direction. It sets the amount of data in a relation, expressed as a percent. Whereas streaming or upload protocols like RTP or SMTP transmit much more data than they receive, download protocols like HTTP or Post Office Protocol version 3 (POP3) receive more data than they send. In contrast, interaction protocols like SSH or Telnet show a more equal direction bytes ratio.

- **Entropy**: The entropy is a measure of the average information content as defined by Shannon [Sha48]. It expresses how many bits are

necessary to store a word. In addition, it is a measure how random the distribution of information content is. According to Shannon, the entropy is defined as described in (3.2),

$$H(I) = -\sum_{i=1}^{N} p_i \cdot \log_2 p_i \tag{3.2}$$

where i represents the piece of information, N the number of possible byte values and P_i the probability of occurrence. The less redundancy the information contains, the higher the entropy value is. Usually, the highest entropy is reached in encrypted messages and for audio and video protocols, whereas plain text messages show a very low entropy. If all byte values equally distributed, the maximum entropy H_0 is defined as shown in (3.3).

$$H_0 = \log_2 N \tag{3.3}$$

- **First Four bytes Checksum**: This metric calculates a simple checksum over the first four payload bytes of a packet. The checksum is calculated using the cross sum algorithm, shown in (3.4).

$$q_n = \sum_{i}^{3} b(i) \tag{3.4}$$

  Although this meter does not guarantee uniqueness, it is a lightweight solution to identify protocols with predefined reoccurring start signatures, like HTTP.

- **Action/Reaction Checksum**: This metric measures the re-occurrence of the first three bytes checksum in the case of direction changes. This measure identifies reoccurring patterns in client server plain protocols like IRC or HTTP, where there are fixed behaviours like keep alive ping pong messages or request and response sequences.

- **First 32 Byte-Pairs Frequency**: This metric measures the re-occurrence of byte pairs with the same value within the first 32 bytes of the payload. The byte pairs do not need to be placed in sequence. Therefore, not only the location of the first byte-pair's item is stored, but also the distance to the next occurrence. Examples for this are the `SSH` string within the SSH banner, the `22` message code in FTP server commands but as well e.g. the letters `t` in the word torrent.

- **First Three bytes Similarity**: This metric checks the similarity of the first three bytes within a flow. Some application protocols have a fixed header within a session, like RTP, Trivial File Transfer Protocol (TFTP) and MPEG-TS. It is a special metric, which is only applied to UDP connections. The absolute values and their position is the subject of interest for this meter, but only if they are equal within the flow.

- **Unicode Frequency**: This metric identifies Unicode character supported plain text protocols. It measures the occurrence of Unicode characters in the relation to the packet size. Unicode characters are assumed, if each second byte carries the value `0x00`, as is the case for the basic Latin characters. Since the English language is used most often by Unicode supported protocols like MMS or in Windows Media Audio (WMA)/Windows Media Video (WMV) Streaming payload, the basic Latin characters are used most often. In that sense, this meter measures the occurrence of basic Latin Unicode characters in the relation to the packet size for the first five packets of a connection.

- **Bit Frequency Offset**: To address bit oriented protocol headers, this metric measures the occurrence of bits within the first four bytes in relation to their position.

$$I_i = \sum_{i=1}^{32} i \cdot 2 + x_i \tag{3.5}$$

In (3.5) the probability value for each of the 32 bits is stored in the vector I.

- **First Packet Size**: This metric does not regard the payload itself, but only its length. In particular, the payload length of the first none empty packet is recorded. In many protocols, the first packet contains information about the connection setup and the configuration details. Therefore, the nature of the first packet is different from the following ones. This behaviour was investigated and evaluated by De Montigny-Leboeuf [De 05], who found this characteristic significant for interactive protocols like Telnet.

Other authors like De Montigny-Leboeuf [De 05] and Hjelmvik [HJ09] also integrated different measures, which I did not integrate for result optimisation reasons. Meters like packet jitter, port number values, packet size or duration of flows harmed the statistical precision of the protocol identification using KLD significantly. The packet jitter meter was significantly harmed by network load and therefore can be considered not to be stable enough to serve for protocol identification. The port number meter was discarded, since some provider run services on non-standard ports, which would lessen the identification precision significantly. The packet size meter would measure the packet size distribution during a connection. We found, that the packet sizes are more dependent on the payload's content than on the actual protocol or media type. The duration of flows is also not regarded as relevant since the eSPID usage within the QoSiLAN framework aims at fast protocol identification and not offline identification. At the beginning of a flow the duration is not known and after the flow has finished the QoS session will have been torn down. The F-Measure, defined in section 2.5, is used to evaluate the effectiveness of the proposed algorithm.

3.3.2 Methodology

Preceded by an intensive related work study, the SPID algorithm was identified as best fitting for the QoSiLAN framework since it fulfils the requirements of the targeted environment. It is lightweight, of high precision and enables protocol identification and application payload identification at the same time, even for encrypted or compressed traffic. After the related work study and first tests, the SPID algorithm was selected as an appropriate base. The implementation was evaluated and found to be not well implemented in terms of performance and memory usage and needed to be re-implemented. In addition, laboratory tests showed that the SPID performance was not as good as expected. Therefore, all measures and parameters were evaluated and optimised. In addition, the measures were reconfigured and additional ones developed and tested to find a better measure configuration for the QoSiLAN framework evaluation test-bed. The complete re-engineered algorithm was called Enhanced Statistical Protocol IDentification (eSPID). The eSPID implementation is written in C++ using the libpcap/winpcap API interfaces [NRG; CAC] for optimised performance.

The Enhanced Statistical Protocol IDentification (eSPID) evaluations, to find the best measure and algorithm configuration were carried out using a set of 3135 flows from 17 different protocols. The set of evaluation flows was recorded from real web-browsing and application usage under various usage scenarios to cover different protocol behaviours for the same protocol. During the development process and for the fine-tuning of the algorithm, the different measures were tested individually to verify their performance and usefulness for the whole set of measures.

3.3.3 Evaluation

Within this section the eSPID algorithm will be calibrated, evaluated and its accuracy quantified. In addition, the evaluation of the main configuration parameters is presented.

3.3.3.1 eSPID Calibration Results

3.3.3.1.1 Number of Packets to Inspect

Important for the algorithm accuracy is the required number of inspected packets for identification. This value was calibrated using evaluations. Figure 3.14 shows the evaluation results. It shows that plain text protocols can

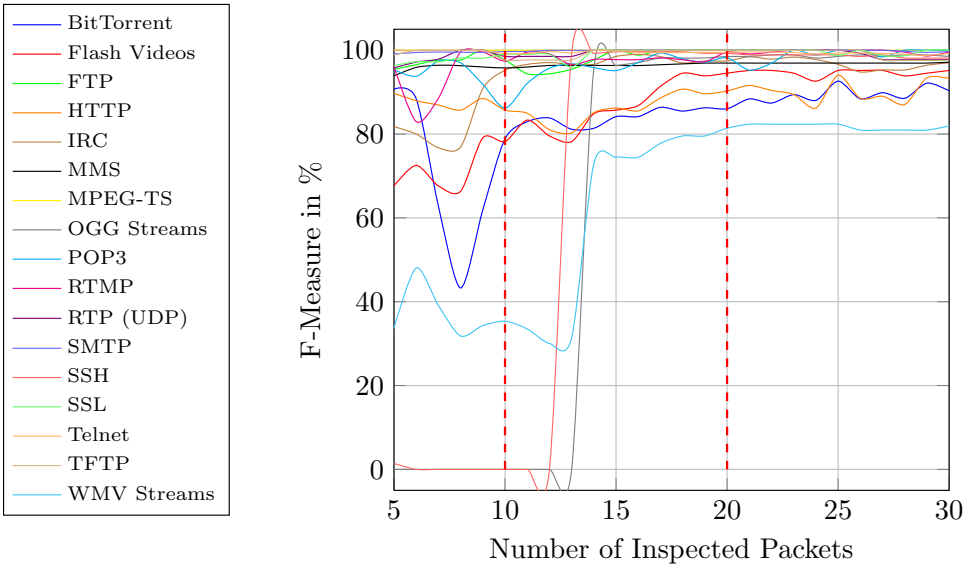


FIGURE 3.14: Number of Inspected Packets versus Algorithm Accuracy (F-Measure)

be identified reliably after receiving 10 packets, where all protocols reach a F-Measure of more than 80 %. Protocols with a high entropy in the payload, like audio (OGG) and video (WMV) need up to 20 packets of inspection

to reach a F-Measure of 80 %. For that reason the number of inspected packets before the KLD is applied was set to 20 packets for all further evaluations. This is because protocols with high content entropy values cannot be identified by content signatures. There, the statistical measures taking into account the connection parameters gain more significance. But these require more packet samples. If a fast identification process is demanded, with less accuracy requirements, the number of packets to inspect can be reduced to 10.

3.3.3.1.2 Number of Trained Flows

To optimise the learning process for generating the fingerprint database, evaluations were used. The results, presented in Figure 3.15 show how

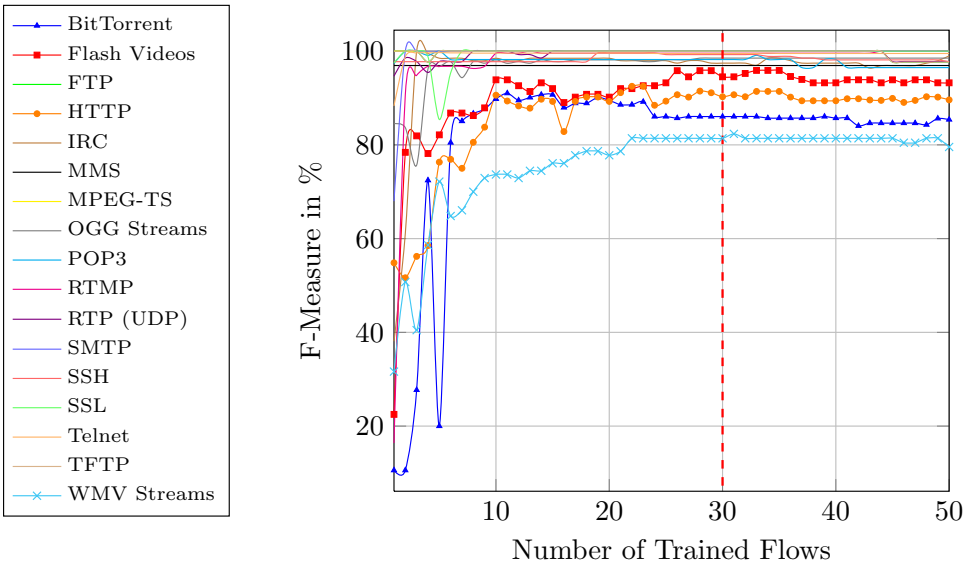


FIGURE 3.15: Flow Training Evaluation

many flows are needed to generate reliable fingerprints for a given protocol. The results show a transient effect in the F-Measure until a number of 30 flows is reached. For more than 30 flows, the statistical learning process does not benefit further for all investigated protocols.

3.3.3.1.3 F-Measure Threshold

In order to obtain receive reliable results, the F-Measure threshold has to be defined. The threshold sorts out false positive results from unknown protocols. The threshold is compared to the KLD sum. If the KLD sum lower than the threshold, the identification is not regarded as sufficiently reliable and therefore the flow is marked as 'unknown'. Figure 3.16 shows

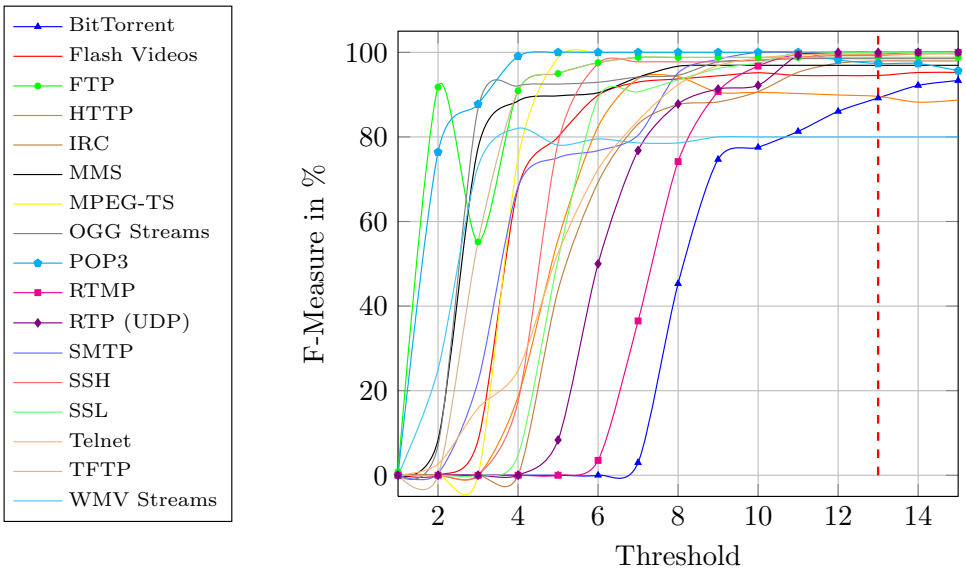


FIGURE 3.16: F-Measure Threshold Evaluation

the evaluative approach and its result. For the F-Measure threshold evaluation, the F-Measure was investigated using different threshold values. The evaluation was carried out using protocol statistics, learnt from 30 flows and the identification was carried out after the 20^{th} packet. The results in Figure 3.16 show the highest F-Measure for all investigated protocols using a threshold of 13. This value was used for all following evaluations.

3.3.3.2 eSPID Evaluation Results

During the protocol meter investigation process a continuing evaluation process validated the meters for their usefulness to the identification precision, as shown in Figure 3.17. The eSPID evaluations were carried out using a set of 3135 flows from 17 different protocols, as listed in Table 3.1. The Protocol column gives the protocol name, the Learned column states how many flows were selected for the learning process. The Validations column provides the number of samples used for the evaluations. The Recall, Precision and F-Measure provide the evaluation results in percent. To assess the eSPID algorithm, different measures were used. In Figure 3.17, the recall, precision, and F-Measure parameter were evaluated according the definitions presented in section 2.5. A complete list of results can be found in Table 3.1. The evaluation shows that for 35 % of the assessed

Protocol	**Learned**	**Validations**	**Recall**	**Precision**	**F-Measure**
BT	30	197	76.65 %	98.05 %	86.04 %
FLV	30	70	98.57 %	90.70 %	94.52 %
FTP	30	461	97.62 %	100.00 %	99.80 %
HTTP	30	127	91.34 %	88.55 %	89.92 %
IRC	30	100	95.00 %	100.00 %	98.51 %
MMS	30	252	94.05 %	100.00 %	96.93 %
MPEG-TS	30	40	100.00 %	100.00 %	100.00 %
OGG	30	122	97.06 %	100.00 %	98.51 %
POP3	30	55	100.00 %	96.49 %	98.21 %
RTMP	30	112	99.11 %	100.00 %	99.55 %
RTP/UDP	30	69	100.00 %	100.00 %	100.00 %
SMTP	30	464	100.00 %	100.00 %	100.00 %
SSH	30	136	98.53 %	100.00 %	99.26 %
SSL	30	41	100.00 %	100.00 %	100.00 %
Telnet	30	812	99.01 %	100.00 %	99.50 %
TFTP	30	42	97.62 %	100.00 %	98.80 %
WMV	30	35	100.00 %	68.63 %	81.40 %

TABLE 3.1: eSPID Evaluation Results.

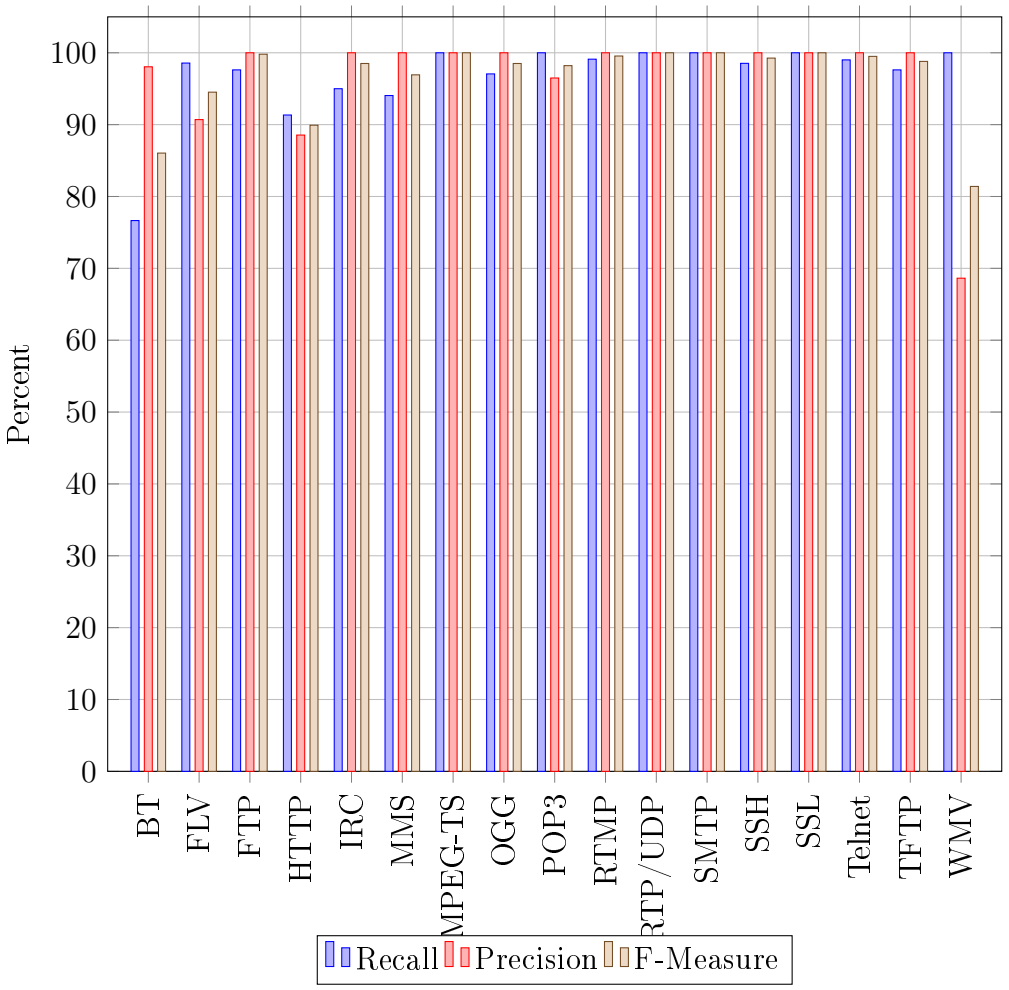


FIGURE 3.17: SPID Evaluation Results

protocols a recall, precision and F-Measure of 100 % were reached. On average a recall of 96.74 %, a precision of 96.61 % and a F-Measure of 96.53 % were reached. The median values for recall 98.57 %, precision 100 % and 98.8 % emphasise the high accuracy that the proposed eSPID algorithm can achieve.

3.3.4 Conclusion

The SPID algorithm was successfully enhanced to provide reliable protocol identification for the QoSiLAN framework's requirements. It can learn new protocols from a set of 30 flows. Protocol identification is reliable after the 20^{th} packet with a F-Measure greater than 81.4 %. A threshold was evaluated to minimise false positive protocol identifications. In comparison to the SPID algorithm, the eSPID algorithm profits from an optimised set of statistical measures, providing better results. Compared to SPID, the eSPID reached also an average 100 % precision with no false positives, but accomplished 96.74 % recall where the SPID by Hjelmvik and John achieved only 91.9 %. Even the F-Measure of 96.53 % was higher than the 95.6 % Hjelmvik and John realised.

3.4 The Statistical Class Based Bandwidth Prediction

To be able to reserve appropriate resources for an identified flow, the amount of required resources must be estimated, right from the start, in the case where this information is not known a priori. The Statistical Class Based Bandwidth Prediction (SCBP) algorithm is designed to predict bandwidth requirements for individual flows, taking into account the different kinds of streaming characteristics. The novel approach for SCBP described in this section enables the average resource usage prediction for the first minute of a flow, ten seconds after the start. The contribution is the whole design and protocol of the Statistical Class Based Bandwidth Prediction (SCBP) algorithm, which was developed from experimental observations.

3.4.1 Algorithm

The inspiration which led to the algorithm came while performing experimental analysis and observations. State of the art video CDN services and clients don't display a continuous traffic characteristic when streaming real-time content. In most cases they produce a traffic peak at the beginning to fill the client-buffers very quickly to enable users to view the first seconds of the videos instantly. Only if users continue to consume the media, additional content chunks are downloaded. This behaviour saves traffic for the content providers in cases where users switch away from the consumption right after the start and it enables a fast start to minimise the wait time at the beginning. It was found that the amount of data transferred within the first peak as well as the start of additional chunk download enables one to draw conclusions on the average media bit-rate consumption of a flow. To optimise the algorithm and to reduce it's mathematical complexity, a

correction factor k was introduced and determined by experimental evaluations. Further, the traffic characteristic is classified to determine the factor k for each class separately to gain more accurate prediction results.
The contributed prediction formula, shown in (3.6) was designed

$$\overline{P_{10}} = \frac{\left(\frac{1}{N_{10}} \sum_{i=0}^{N_{10}} \chi_i\right)^2}{\frac{1}{N_5} \sum_{i=0}^{N_5} \chi_i} k_c = \frac{\left(\overline{B}\right)^2}{\overline{B_a}} k_c \tag{3.6}$$

for a balance of simplicity and accuracy. It takes into account the application in embedded systems with less computation power, which also saves battery life on mobile systems. The prediction formula (3.6) implements the correction factor k_c, which reflects the class' characteristics. The value k_c for each class was found using evaluations, as carried out the following section 3.4.3.

In order to reflect the different kinds of streaming characteristics flows may have, the Statistical Class Based Bandwidth Prediction (SCBP) algorithm classifies the traffic into six different traffic classes (A–F), according to the traffic behaviour within the start-up phase of the transmission, as shown in Figure 3.18. The classes are used to select the correct factor value for the constant k used in the prediction formula (3.6). The clustering of results to classes as well as the corresponding factor k was found using evaluations. To achieve the highest accuracy, the measurement must start at the very beginning of a traffic flow. During the observation time I, m bandwidth measurement values B_i are collected. From this we get the set B, containing m bandwidth measurement samples. For better flow characterisation, we divide the set of B into two subsets: $B_a \subset B := \{B_0, \ldots, B_{\frac{m}{2}}\}$ and $B_b \subset B := \{B_{\frac{m}{2}+1}, \ldots, B_m\}$, where we calculate the maximum ($B_{a,max}$ and $B_{b,max}$), as well as the average ($\overline{B}$, $\overline{B_a}$ and $\overline{B_b}$) values. In addition, B_{max} is determined as the maximum value in the set of B. The prediction, based on the data collected during the first I =10 s,

TABLE 3.2: Traffic Classes Clustering

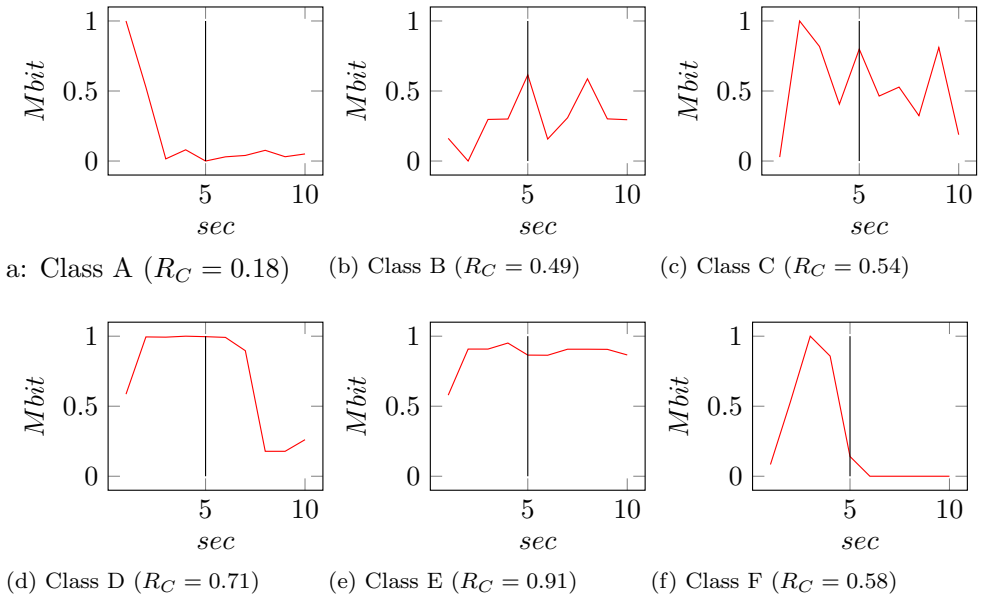
ClassName	Characteristic (R_c)
A	$0 \leq \frac{\overline{B}}{B_{max}} \leq 0.25$
B	$0.25 \leq \frac{\overline{B}}{B_{max}} \leq 0.5$
C	$0.5 \leq \frac{\overline{B}}{B_{max}} \leq 0.6$
D	$0.6 \leq \frac{\overline{B}}{B_{max}} \leq 0.8$
E	$0.8 \leq \frac{\overline{B}}{B_{max}} \leq 1$
F	$\frac{\overline{B}}{B_{max}} \prec 1 \wedge \overline{B_b} = 0$

forecasts the average bandwidth consumption for the whole first 60 s of transmission.

The sampling interval is defined as $s = \frac{I}{m}$. The evaluations are based on $s = \frac{10\,\mathrm{s}}{10\ samples} = 1\ s$. The ratio R_c, defined in (3.7)

$$R_c := \frac{\overline{B}}{B_{max}} \tag{3.7}$$

classifies the flows into the categories A–F, as shown in Table 3.2 and illustrated in Figure 3.18. There, in particular the ratio of bandwidth consumption within the first five and the next five seconds are taken into account. The classes represent dedicated constant values, which are applied to k_c in (3.6), the prediction algorithm. The SCBP algorithm predicts the bandwidth requirements for identified streams for the period of 60 s after I =10 s of observation. Figure 3.18 shows examples for the characteristics in the first ten seconds for the six different traffic classes. One can see, that the criteria basically reflects the first burst length. The class F addresses the special characteristic, when there are no samples in the set of B_b, but the stream has not finished yet. The Figures $3.18a - e$ give examples for the first 10 s characteristic of the traffic classes. There, Figure 3.18e represents the classical continuous streaming case, whereas Figure 3.18f represents the case, where there is a burst at the beginning and no data within the last half of the observation interval of 10 s.



a: Class A ($R_C = 0.18$) (b) Class B ($R_C = 0.49$) (c) Class C ($R_C = 0.54$)

(d) Class D ($R_C = 0.71$) (e) Class E ($R_C = 0.91$) (f) Class F ($R_C = 0.58$)

FIGURE 3.18: Examples for the SCBP Classes

Through evaluations, I validated, that the k_c-values assigned to the classes A–F were applicable to the ranges as defined in Table 3.2. Hence, the classes reflect the bandwidth characteristic, including the start up behaviour of the flow in a simple way. To define the cluster-range for each class, assignment evaluations using the measured error deviations were put into a R_c density histogram to identify the ranges of similar error deviations, as shown in Figure 3.19 An error deviation value if 1 means no error. Values < 1 represent under-estimations, whereas values > 1 represent over-estimations. This clustering is similar to the SVM model approach described in section 2.5.2.2. In this Figure, results above the dotted line, marking an error level of 1 represent over-estimations, whereas values underneath the line represent under-estimations. Results with an error value of 1 on the dotted line represent perfect prediction matches.

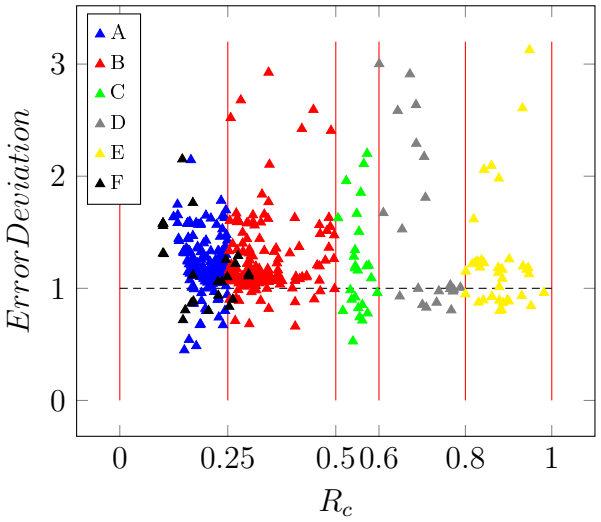


FIGURE 3.19: Class Range Clustering Evaluation Results

3.4.2 Methodology

The Statistical Class Based Bandwidth Prediction (SCBP) algorithm was researched and developed from the motivation to have a simple and light-weight algorithm with low computing complexity. After a literature review and investigation of real-world traffic from major video and audio streaming portals, as described in section 2.6, it was found that the state of the art literature solutions do not handle the characteristics of contemporary Internet media streaming traffic in an appropriate manner. For this reason the SCBP algorithm was designed from the practical observation that streams need a priori classification and case by case handling before predications should be applied. In particular, the first ten seconds of transmission were found to be significant for estimating the overall transfer behaviour. Dependent on the characteristic of the first ten second transfer behaviour, a systematic deviation from the expected results could be discovered. Therefore, intensive evaluations were carried out to optimise the classified results using individual correction factors for each class. In addition, different optimisation approaches were followed in parallel and compared to find the best

optimisation set for the prediction results. All evaluations for the Statistical Class Based Bandwidth Prediction (SCBP) were performed with real Internet traffic from common WebTV, IPTV, Internet Radio and on-demand platforms, located in Germany, United Kingdom, France and the United States of America. The evaluation was carried out according to Figure 3.20 on Host1, connected using 100BaseT Ethernet to an Internet gateway, which provides access to the University's[1] Internet connection. The SCBP implementation is written in C++ using the portable ACE library and the libpcap/winpcap API interfaces [NRG; CAC] for packet analysis. For the

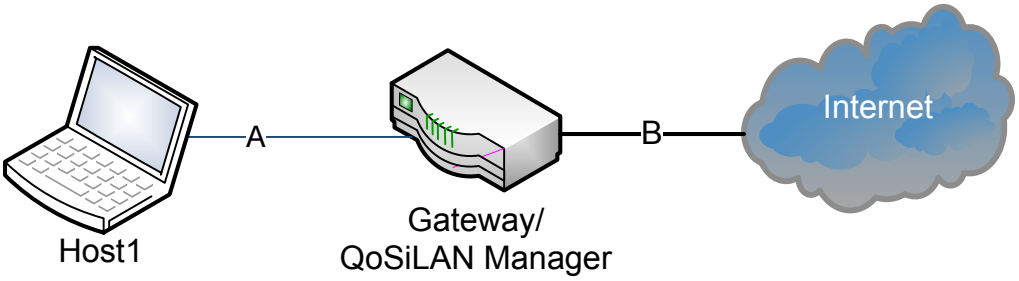


FIGURE 3.20: QoSiLAN Evaluation Scenario

evaluation, the streams were automatically identified and classified using the Enhanced Statistical Protocol IDentification (eSPID) algorithm as discussed in section 3.3. Only streams with a minimum transfer time of 60 s were included in the evaluation, which resulted in a set sized of 463 samples. This allowed us to get the significant average bandwidth consumption value for the first 60 s ($\overline{B}_{60}$) for each identified flow. The ($\overline{B}_{60}$)-value served as a reference and was used to validate the prediction accuracy after ten seconds (A_{10}). The evaluation is described in detail in section 3.4.3.

[1] Technische Hochschule Mittelhessen (University of Applied Sciences Mittelhessen), Germany (THM)

3.4.3 Evaluation

The evaluation helped to calibrate the algorithm and to validate its accuracy. To improve the readability of this section, Table 3.3 provides a compact overview of the most important abbreviations used here. To eval-

TABLE 3.3: Abbreviations

Over-Estimation Rate	OER
Mean Estimation	ME
Prediction Hit Rate	PHR
Mean Prediction Accuracy Ratio	MPAR
Class Based Quality of Service	CBQoS
Class Based Prediction Hit Rate	CBPHR
Class Based Mean Prediction Accuracy Ratio	CBMPAR
Class Less Quality of Service	CLQoS
Class Less Prediction Hit Rate	CLPHR
Class Less Mean Prediction Accuracy Ratio	CLMPAR

uate the SCBP algorithm the test-bed scenario depicted in Figure 3.20 was used. There, Host1 is connected using 100BaseT Ethernet to an Internet gateway, which provides access to the University's[1] Internet connection. The streams were automatically identified and classified using to the eSPID algorithm discussed in section 3.3.

All streams had a minimum transfer time of 60 s. This allowed us to compare the predicted value (P_{10}) with the significant average bandwidth consumption for the first 60 s ($\overline{B}_{60}$). This value served as reference and was used to validate the prediction accuracy after ten seconds (A_{10}).

$$A_{10} = \frac{P_{10}}{\overline{B}_{60}} \tag{3.8}$$

[1] Technische Hochschule Mittelhessen (University of Applied Sciences Mittelhessen), Germany (THM)

Other important measures are the Prediction Hit Rate (PHR), the Mean Prediction Accuracy Ratio (MPAR) and the Over-Estimation Rate (OER).

- The Prediction Hit Rate specifies the number of predictions, which are in the range of $0.8 \preceq A_{10} \preceq 2$, as percentage. This range was specified to filter unacceptable outliers for consumer media traffic prediction.
- The Mean Prediction Accuracy Ratio specifies the mean prediction accuracy $\overline{A}_{10}$ over all samples. Since the prediction causes network blocking through QoSiLAN reservations as resources are set aside, the Mean Prediction Accuracy Ratio reveals the mean network blocking rate.
- The Over-Estimation Rate specifies the number of prediction accuracy ratios with $A_{10} > 1$ as a percentage. This is an important measure for QoS critical applications, since it reflects the QoS assurance probability a prediction algorithm configuration can achieve.

For the evaluations a set of 1442 monitored media streams was used. All these parameters were evaluated for the whole stream set ClassLess (CL) and per class individually Class-Based (CB). The k-value in (3.6) was optimised for each of these three measures individually to find the best method for optimisation as defined in the following:

- The k-value for Prediction Hit Rate was evaluated for the maximum Prediction Hit Rate in (3.6), to receive the maximum number of results in the acceptable range.
- For the Mean Prediction Hit Rate (MPHR), the k-value was evaluated to give in (3.6) a result in average of 1, which reflects the best mean network blocking or utilisation rate.

- The QoS optimization's k-value was evaluated to result in (3.6) an average of 90 % Over-Estimation Rate to ensure a minimum of 90 % QoS assurance.

Finding the best optimisation configuration is challenging. First, the Mean Prediction Accuracy Ratio (MPAR) should be near to 1, to ensure the best network utilisation. This would be reflected by a traditional error calculation like the Root Mean Square Error (RMSE). Second, at the same time as many as A_{10} results should match the acceptable range, to exclude under- and over-utilisation. Third, for QoS reasons it is important to have as many as possible A_{10} results greater than 1, to avoid under-provisioning of resources. To find the best optimisation, which meets all of these requirements at the same time, the Euclidean distance ($\boldsymbol{d}$) was employed, as shown in (3.9).

$$d = \sqrt{(1 - MPAR)^2 + (1 - PHR)^2 + (1 - OER)^2} \tag{3.9}$$

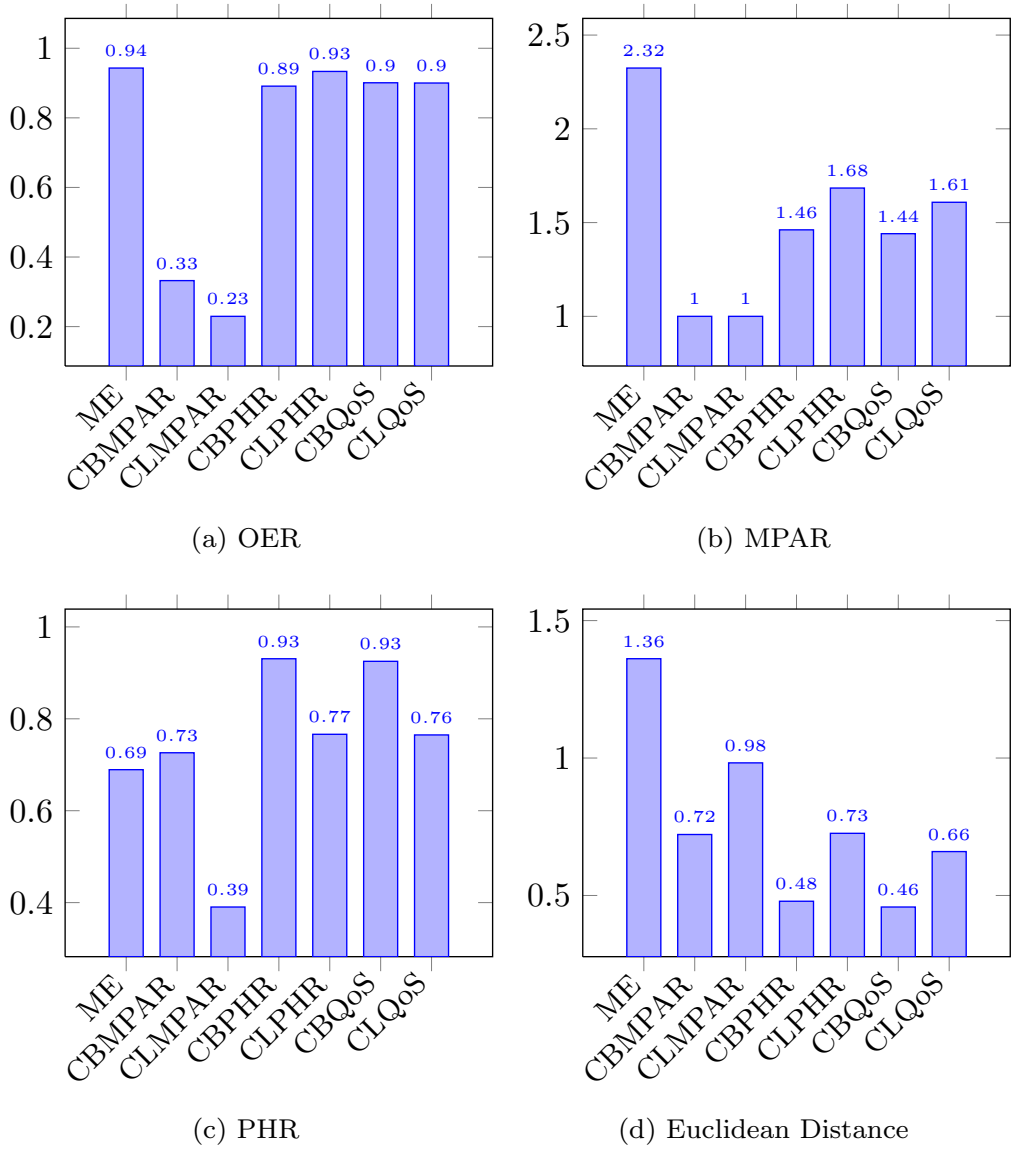
It takes all three parameters into account and provides the minimum $\boldsymbol{d}$-value, which reflects the best fitting algorithm. In Figure 3.21d) a smaller value of $\boldsymbol{d}$ indicates a better performance, taking into account the three measures MPAR, Prediction Hit Rate and OER. Table 3.4 gives an overview

TABLE 3.4: Overview of Results

optimization	k	$\widetilde{MPAR}$	$\boldsymbol{d}$	$\overline{MPAR}$	PHR	PAR>1	PAR<1
CLME	1.0000	1.1561	1.3613	2.3241	68.93 %	94.31 %	5.69 %
CBMPAR	class based	0.8676	0.7218	1.0000	72.61 %	33.22 %	66.78 %
CLMPAR	0.6335	1.2023	0.9824	1.0000	39.04 %	22.95 %	77.05 %
CBPHR	class based	1.1949	0.4790	1.4613	93.07 %	89.11 %	10.89 %
CLPHR	1.0670	0.7324	0.7261	1.6842	76.63 %	93.34 %	6.66 %
CBQoS	class based	1.2335	0.4580	1.4408	92.51 %	90.08 %	9.92 %
CLQoS	1.0189	1.1779	0.6597	1.6083	76.49 %	90.01 %	9.99 %

of the results achieved for the different configurations. There, also the median ($\widetilde{MPAR}$) and the mean accuracy ($\overline{MPAR}$) values are listed, which enable a first assessment of the results. While the mean accuracy value emphases the average network utilisation, the median accuracy value gives an indication about the distribution of values. Figures 3.21a)-c) give an overview of the three different optimisation aspects, whereas Figure 3.21d) shows the results after the Euclidean distance calculation, which reflects a combination of the previous ones.

In Figure 3.21a) the Over-Estimation Rate (OER) optimisation, with an target value of 1 indicates that the aspects ME with 0.94 and the Class-Less Prediction Hit Rate (CLPHR) with 0.93 perform best. The ClassLess QoS Range (CLQoS) reaches 0.9 and performs only on the third place followed by the Class-Based QoS range (CBQoS) aspect with 0.89. In Figure 3.21b the Mean Prediction Accuracy Ratio (MPAR) results with an target value of 1 indicate the optimum for the ClassLess Mean Prediction Accuracy Ratio (CLMPAR) and Class-Based Mean Prediction Accuracy Ratio (CBMPAR) aspects, which both reach 1. This was expected, but the CBQoS aspect performs next best reaching 1.44. The Prediction Hit Rate (PHR) results in Figure 3.21c show the best performance for the Class-Based Prediction Hit Rate (CBPHR) with 0.93 and the CBQoS reaching 0.93. These values should be near to 1 to be best. This is interesting and shows that the optimisation for PHR without classification cannot provide an average ratio better than 0.77. Regardless, the class-based approach gives a better density of results in the acceptable range. The Euclidean distance assessment, with a target value of 0, presented in Figure 3.21d illustrates the advantage of the CBQoS approach reaching 0.46. It does not perform best in all categories, but collectively over all optimisation aspects it shows the best performance overall as revealed by the Euclidean distance. In addition, it verifies the class based approach, since all class based approaches perform better than their classless counterparts.



(a) OER

(b) MPAR

(c) PHR

(d) Euclidean Distance

FIGURE 3.21: Performance Assessment of Optimisation Results

3.4.4 Conclusion

The statistical class-based bandwidth prediction algorithm using the CBQoS optimisation configuration was evaluated and shown to perform the best for predicting bandwidth requirements for the relative long period of 60 s.

On first viewing the results achieved without classification look very good and perform in some configurations similar to the class-based approaches or even better. Finally the Euclidean distance comparison in Figure 3.21d reveals the advantage of the CBQoS optimisation approach, since it brings the deviation of results into account, combining the different measures. Also, I showed, as presented in Figure 3.21, that a 100 % prediction accuracy with a minimum error is not desirable, as it causes a higher under-estimation probability than e.g., a QoS optimisation case. Generally, it is more critical for the media streams, if their resources are under-provisioned, since they are not properly protected from congestion. In contrast, blocking of more resources than needed is critical for the overall network performance and utilisation. In the application of QoS, a little over-estimation of resources is also to be regarded as positive, since this allows the streams to pre-buffer faster at the receiver, which results in more robustness against the variability of network performance and provides a better stream isolation against disturbing traffic and congestion.

3.5 The QoSiLAN Communication Protocol

The hosts in the network, supporting the QoSiLAN framework need to communicate with each other in order to coordinate the resource reservations. To achieve this, the novel QoS Signalling Layer Protocol for Local Area Networks (QSLP-LAN), a NSIS Signaling Layer Protocol (NSLP), was designed according to the latest IETF recommendations. This protocol provides the required features to enable per link resource reservations in an end-to end manner in LANs employing the support from the other hosts instead the LAN's infrastructure. The QSLP-LAN protocol enables the communication and cooperation of hosts to negotiate and implement the QoS functionality. As shown in Figure 3.1, the protocol is used by the policing and admission control module to coordinate the QoS policies within the hosts in the network. The QoSiLAN framework's protocol is designed according to the latest IETF NSIS recommendations [Man+10], and named NSIS QSLP-LAN. The core elements of the QoSiLAN approach are the policing and signalling procedures. Whereas existing QoS protocols communicate end-to-end and QoS aware network elements are involved, in QoSiLAN it is different. One end-system, preferably the originator of the traffic, informs a QM host about the needed resources. A QM could be any host in the network. In an optimal configuration, the gateway is assigned the role of the QM, since it can also control the outgoing traffic directly. Using information from the network topology map, the QM generates sophisticated policy resource requests and sends them to all other hosts in the network to indirectly achieve resource reservation for the physical links along the data path by collaborative traffic shaping. Within this section, the QoSiLAN protocol message types and the signalling procedures are presented.

3.5.1 Algorithm

The QoSiLAN's QoS signalling protocol, QSLP-LAN, provides the mechanisms to exchange QoS information and signalling commands within the QoSiLAN peers on top of the NSLP framework, adapted for the QoSiLAN QoS framework. First, the QSLP-LAN message format is described in detail. Second, the protocol algorithm itself is explained and the signalling procedure discussed.

3.5.1.1 QSLP-LAN Message Format

The QoSiLAN protocol is implemented using the NSIS message format, similar to the QoS NSLP header format as described in RFC5974 [MKM10], section 5.1.
The QoSiLAN protocol uses, like other common QoS protocols, a soft state mechanism. This means, the hosts keep a reservation state as long as a time out has not been reached. To keep states, a reservation refresh message (QoSiLAN_RESERVE), with the same parameters as the first one, must be sent before the time-out is reached.

- **Reservation (QoSiLAN_RESERVE)**
  The reservation messages use the NSLP common header, as all GIST NSLP objects [SH10] do. The message type is interim defined to be $0xF1 = QoSiLAN\_RESERVE$). The format of a QoSiLAN_RESERVE message is described as shown in Definition 3.1:
  In contrast to a QoS-NSLP RESERVE message, all listed objects

```
QoSiLAN_RESERVE  =  COMMON_HEADER
                    RSN REFRESH_PERIOD
                    BOUND_SESSION_ID QSPEC
```

DEFINITION 3.1: QoSiLAN_Reserve Message Format

are mandatory. The format of the objects is designed according to RFC5974 [MKM10], section 5.1.3. The RSN, `type=0x02`, defines the order in which reservations were sent and are incremented on message sent. The REFRESH_PERIOD, `type=0x03`, defines the lifetime of the reservation and its soft state. The BOUND_SESSION_ID, `type=0x04`, defines the nature of the binding, e.g. bi-directionality, and the `session-id` is used to identify the reservation. The QoS Specification (QSpec) object, `type=0x0B`, defines the QoS information. It lists the requested resources using the `Traffic Model (TMOD-1)` parameter, which contains the peak data rate for traffic shaping, the `Excess Treatment` parameter defining the shaping policy, as specified in RFC5975 [Ash+10]. In addition, Iintroduced the new `Reservation Path Parameter for IPv4 (RPP-IPv4)` using the dedicated `parameter-id = 0x3C`, to communicate the five-tuple, which defines what flow the reservation is for, as shown in Figure 3.22. The

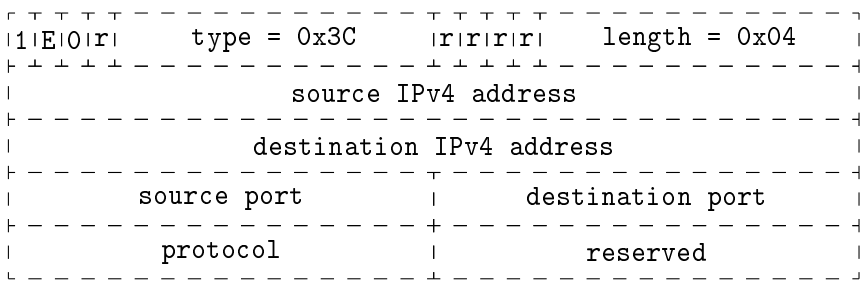


FIGURE 3.22: Reservation Path Parameter for IPv4

first 32-bit are defined according to the QSpec Parameter Header [Ash+10] with a length of `0x04` 32-bit words. Next, the source and destination IPv4 addresses occupy 32-bit each. For the source and destination ports and the transport protocol ID 8-bits are reserved each. For IPv6 a similar parameter header format with 64-bit IP address fields needs to be defined accordingly. Reservations are deleted either on soft state time out or by sending a QoSiLAN_REQUEST

message using the BOUND_SESSION_ID without a QSpec object defined.

- **Response (QoSiLAN_RESPONSE)**
  The response message is intended to report success or error codes to the requesting node. The success case is indicated using the `error=0x00` value. If a reservation state is rejected, a negative acknowledgement is signalled through a `error=0x47` value set in the INFO_SPEC header. The message type is interim defined to be $0xF2 = QoSiLAN\_RESERVE$. The format of a QoSiLAN_RESPONSE message is described as shown in Definition 3.2:

  All objects are mandatory. The RSN object is taken from the corresponding QoSiLAN_REQUEST message to match the response to a request. The INFO_SPEC contains information about success or errors. It defines a 8-bit `error-code`, a 4-bit `error class` a 4-bit `error-source identifier-type` and a 8-bit `error-source identifier-length`, as defined in RFC5974 [MKM10], Sub-Section 5.1.3.6.

```
QoSiLAN_RESPONSE  =  COMMON_HEADER
                     RSN INFO_SPEC
```

DEFINITION 3.2: QoSiLAN_RESPONSE Message Format

- **Notify (QoSiLAN_NOTIFY)**
  The notify message is intended to report significant best effort traffic flow statistics to the QoSiLAN manager. The message type is interim defined to be $0xF3 = QoSiLAN\_NOTIFY$. The format of a QoSiLAN_NOTIFY message is described as shown in Definition 3.3:

  All objects are mandatory. The objects are taken from the corresponding QoSiLAN_REQUEST and QoSiLAN_RESPONSE messages as described before. The QoSiLAN_NOTIFY message also uses the RPP-IPv4 within the QSpec header to announce detected flows.

```
QoSiLAN_NOTIFY  =  COMMON_HEADER
                   INFO_SPEC QSPEC
```

DEFINITION 3.3: QoSiLAN_RESPONSE Message Format

3.5.1.2 QSLP-LAN Signalling Procedure

Figure 3.23 shows the signalling in the network, used for bandwidth reservation in the sample LAN, as presented in Figure 3.24.

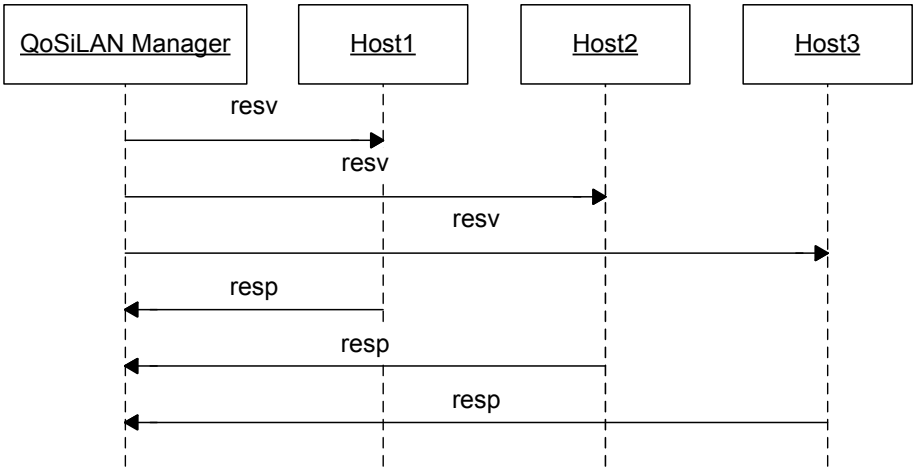


FIGURE 3.23: QM Initiated Message Flow

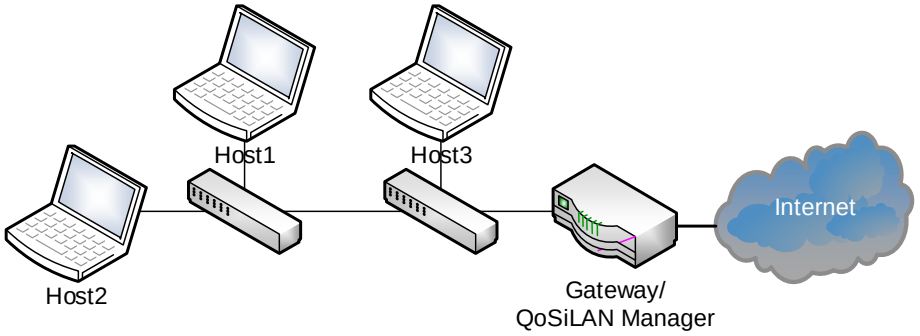


FIGURE 3.24: Sample Signalling Scenario

In this LAN one QM host and three client hosts are connected using two different switches. All of them, except the switches, are QoSiLAN aware.

The initiator, in this case Host1, sends a reservation request message (QoSiLAN_RESERVE) to the QM. The reservation request shall contain at least the physical addresses and the IP addresses of the two communicating parties and the requested resource, and the predicted bandwidth to reserve. The QM analyses the location of the hosts and sends sophisticated reservation requests (QoSiLAN_RESERVE) for all other nodes to Host1, Host2 and Host3 in the network, based upon the LAN topology and the LAN's traffic status knowledge, to encourage all hosts to obey the limitations of the affected physical links. In return, the nodes acknowledge the request to the QM (QoSiLAN_RESPONSE), as well as the QM reports (QoSiLAN_RESPONSE) the result code to the initiator (Host1) at the end. To cover the case where the sending and receiving host do not support QoSiLAN, but the QM is a gateway and detects the flow, another example is given. As shown in Figure 3.25, on gateway initiation no re-

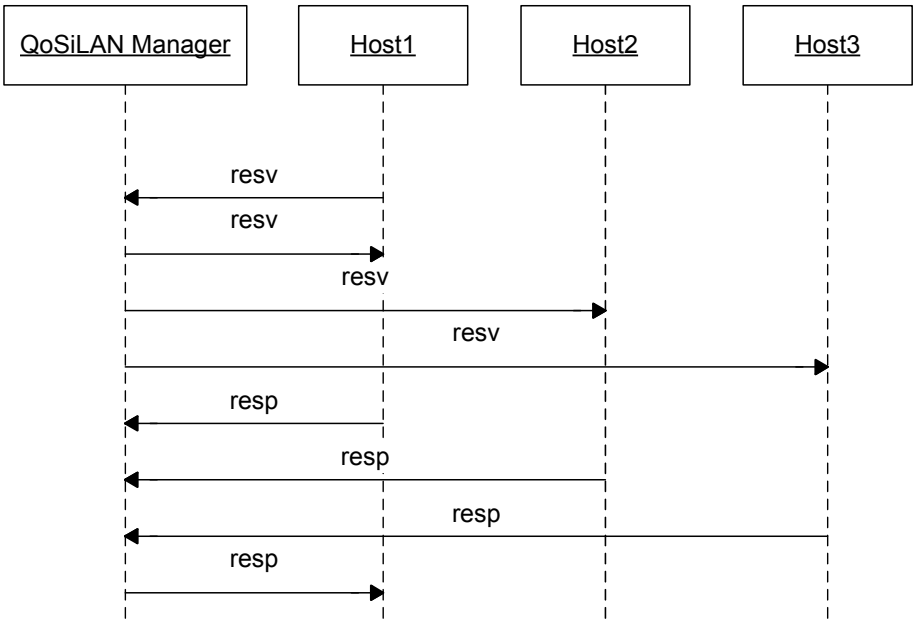


FIGURE 3.25: Host Initiated Message Flow

quest and response communication to other initiation nodes is required. But reservation requests and responses are sent to and from each host in

the network, initiated from the gateway. To avoid unnecessary signalling effort, the hosts do not report their traffic statistics regularly, but only triggered by events. In the case of a significant best effort traffic detection, a host reports the monitored bandwidth of the stream to the QM using QoSiLAN_RESPONSE messages. Details on the policing and admission control algorithm are described in section 3.6.1.3.

The NSIS GIST protocol also includes a host discovery feature, which is used to detect and identify the QoSiLAN enabled hosts in the network. The QM selection is performed by the smallest switch/hub - hop distance to the router, evaluated using the LLTD protocol. If the router itself is QoSiLAN enabled, it proposes a distance of zero. If the router is not QoSiLAN enabled and two host have the same distance, the host with the longer uptime period is selected.

3.5.2 Methodology

The QoS Signalling Layer Protocol for Local Area Networks (QSLP-LAN) was designed after intensive literature research and state of the art Internet protocol specification investigations. The latest IETF recommendations were followed strictly to specify the new protocol accordingly for QoSiLAN's novel QoS framework. The evaluations in section 3.5.3 also include a critical quantitative analytical view on the IETF's NSIS specification and the overhead it causes in a distributed scenario. The NSIS-based QSLP-LAN implementation is written in C++ using the ACE ACE-Wrapper communication framework library for lightness and portable multi-threading and socket communication support.

The QSLP-LAN was evaluated within the test-bed presented in Figure 3.2. The communication was tested on Windows OS-based hosts, as well as on the Linux-based DD-WRT enabled router. The Wireshark [Fou15b] and the Tcpdump tools [Tcp15] were used to inspect and analyse the signalling

data. The QSLP-LAN was also assessed indirectly within the evaluations for the policing and admission control function, presented in section 3.6.1.3. Additionally, the analytical evaluation was performed using mathematical calculations and projections assisted by Microsoft's Excel [Cor15a] application for statistical analysis as presented in section 3.5.3.

3.5.3 Evaluation of Protocol Scalability

This section evaluates the overhead, scalability and performance of the protocol. A typical QoSiLAN_RESERVE message has a length of 124 bytes, including the UDP, IPv4 and GIST header. The GIST message header has a size of 12 bytes. The GIST NSLP data header has a length of 4 bytes. The GIST payload consists of a QoS message, which starts with a QoS NSLP header with a length of 4 bytes, followed by the QoS objects. All QoS objects start with a 2 byte common header. The minimum size is 4 bytes, additionally 32 bit fields are indicated by the length parameter at byte 2 of the QoS common object header. These are the RSN, EpochID, Refresh Period, Bound Session ID and QSpec. The sizes of all headers are listed in Table 3.2. The QSpec contains QSpec objects with a Common QSpec header at the start. The QSpec objects start with a 4 byte parameter header, followed by the object's payload header. These are the TMOD-1, Excess Treatment and the RPP-IPv4. The sum of headers gives a total message length of 116 bytes, as listed in Table 3.2.

The QoSiLAN_Response message has a size of 32 bytes, as listed in Table 3.3. The INFO_SPEC object consists of a 4 byte common header and the 4 byte Error Source Identifier (ESI), which carries the IPv4 address of the message source. For each host in the network, the QM has to generate one message for each IP destination in the LAN, which leads to the network request message load (B_r), generated by the QM, which can be described

Header Name	size [bytes]
GIST	12
GIST NSLP DATA	4
QoS NSLP	4
QoS RSN	4
QoS EpochID	8
QoS Refresh Period	4
QoS Bound Session ID	36
QSpec Common Object	4
QSpec Common QSpec	4
QSpec Object	4
QSpec TMOD-1	24
QSpec Excess Treatment	8
QSpec RPP-IPv4	20
Sum	136

TABLE 3.2: QoSiLAN Request Header Sizes

Header Name	size [bytes]
GIST	12
GIST NSLP DATA	4
QoS NSLP	4
QoS RSN	4
QoS INFO SPEC	8
Sum	32

TABLE 3.3: QoSiLAN Response Header Sizes

using (3.10).

$$f(B_r) = (N_H - 1)^2 s_{req} + (N_H - 1)^2 s_{rep} \tag{3.10}$$

Here, N_H represents the number of host and s_{req} is the size of the request and s_{rep} the size of response messages, respectively. The messages from and to the QM itself are not routed across the network and therefore not relevant to the network load. Figure 3.26 shows the scalability problem of the proposed protocol. It shows three graphs for the number of messages and the generated amount of data, scaled by the number of hosts in the network, represented in a single Figure. As one can see, already 25 hosts in

the network cause over 129 024 bytes of traffic occupied by 1152 messages sent in the network for a single QoSiLAN reservation session establishment. The 1 MB mark is hit with 68 hosts causing 8978 messages. The number of 213 hosts, which generate approximately 10 MB of traffic using 89 888 messages. Over 1 billion messages are needed for a LAN with 709 hosts, generating 112 283 136 bytes of data. This shows, that there is an immense scalability problem, caused by the NSIS protocol stack. When targeting larger scale scenarios, the use of a proprietary protocol and consolidated messages to utilise the network's MTU is strongly recommended.

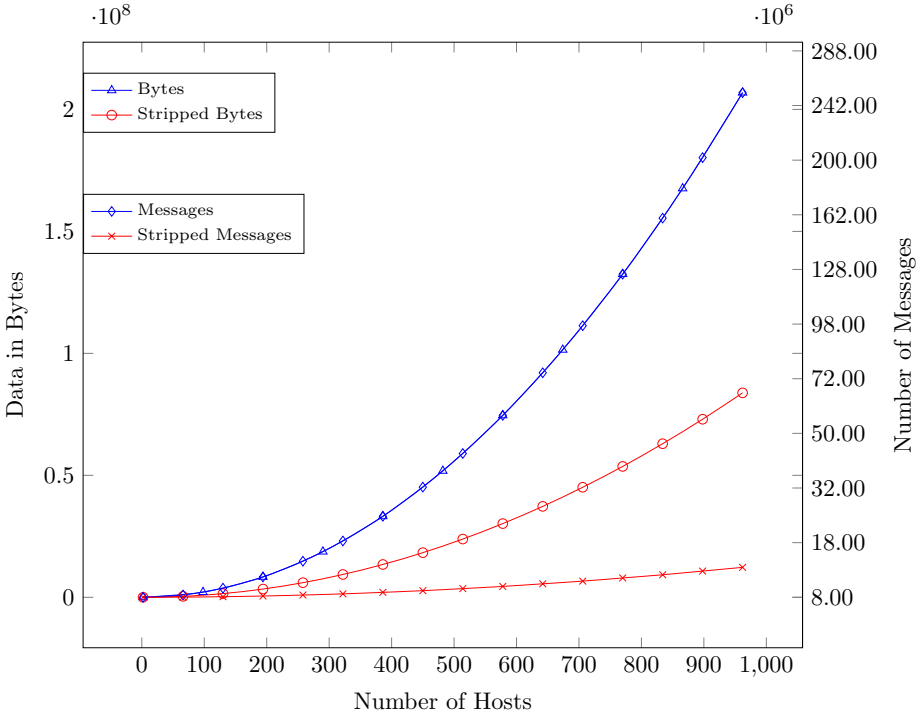


FIGURE 3.26: QoSiLAN Signalling Effort

An approach to reduce the number of messages and the amount of data needed would be to pack the individual policy information affecting one host into a single message, using the maximum path MTU for the transfer. Also, a more minimalistic, proprietary message design with compression

enabled would significantly save signalling resources. This case is integrated into Figure 3.26 and described as 'Stripped Messages' and 'Stripped Bytes'. A QoSiLAN_Request message can be stripped to 83 bytes instead of 136 bytes, when removing all structural NSIS's TLV information and keeping only the essential data - without using data compression. This is a reduction of approx. 39 %. When using a MTU size of 1400 bytes, 16 messages would fit into one packet. As depicted in Figure 3.26, the amount of data needed to to establish a QoS reservation state in a network with 709 nodes would be reduced from 112 283 136 bytes to 45 493 248 bytes, which is a reduction of 59 %. The number of messages would be reduced by 94 % from 1 002 528 to 59 472. In a 25 node network, the number of messages would be reduced from 1152 to 39 and the amount of data needed would scale down from 105 984 bytes to 41 864 bytes.

3.5.4 Conclusion

The QSLP-LAN was designed following the latest IETF recommendations for new signalling protocols. Since the QoSiLAN approach is designed only for small networks with no more than 25 hosts, the number of messages and data used for signalling is not regarded as important. As expected, the signalling effort for a cooperative approach is very high. Anyway, the evaluation proved, that a redesign of the signalling procedure would be needed in case of higher scalability demands. The approach of stripping messages to the essential payload without structural data and using of the LAN's MTU to accumulate messages was analysed. The analytical evaluation proofed the benefit this approach particularly in scenarios with more than 25 hosts. Whereas the number of messages scales proportionally with the used amount of data for the approach using the NSIS message format, the stripped approach provides drastic savings for the number of messages and reduces the required amount of data significantly.

3.6 Policing and Admission Control

The policing and admission control module is the central entity within the QoSiLAN framework, which aggregates all information from the other parts of the system and defines the behaviour of the whole framework. The policing and admission control decisions are made, reflecting the network topology information, protocol identification results, bandwidth predictions and traffic information from other hosts using the QSLP-LAN in order to manage the network in an efficient manner.

3.6.1 Algorithm

The self-organised, host based resource management within a LAN requires detailed knowledge about the available resources on all links between the LAN entities. Only if the link layer topology and the capacity of its link are known, can the QoSiLAN framework manage the resources autonomously. In the following, the management procedures to acquire and to use the required information are described.

3.6.1.1 Resource Discovery

One host in the network, preferably the gateway, fulfils the role of the QM. The Internet gateway is always a good choice, since it manages the traffic from and to the Internet, anyway. Therefore, it is able to analyse the forwarded traffic in both directions and to support the non-QoSiLAN-enabled hosts in the LAN when they communicate with Internet destinations. The QM maps the network topology and acts as resource coordinator. The mapping process is executed each time a new host is discovered in the network. In addition to the LLTD protocol, the host discovery is based on broadcast and ARP packet monitoring to also detect non-QoSiLAN-

and non-LLTD-enabled hosts. The QM maintains the network topology map generated by the LLTD Mapper module, as described in section 3.2. It contains information about the link layer topology between the hosts, switches, hubs and access points in the LAN. In addition to the topology itself, the bottleneck capacity of each link needs to be evaluated. This is done by measurement during network idle times. The mapper advises the hosts to measure the bottleneck bandwidth of their links by active probing. These commands are sent using QoSiLAN_NOTIFY messages with empty RPP-IPv4 parameters, as described in section 3.5.1. For this purpose, the iperf TCP and UDP bandwidth performance measurement tool [ipe13] is used, which runs on all machines in server mode to enable bandwidth measurement. The LLTD Mapper service in the QM gathers the results from all hosts and adds this information to the connection information data within the topology map.

3.6.1.2 Policing Procedure

The policing procedure describes the algorithm used to employ all the information from the companion services, which provide the data needed to make final QoS decisions. Figure 3.27 shows the sequence diagram for this algorithm. First, each host in the network monitors its outgoing traffic continuously. To make autonomous QoS policing possible, the QM needs to gather and maintain all information about the LAN and its links by using the information from the LLTD topology mapper module. This includes the physical paths, as well as the measurement results of the bottleneck bandwidth μ_l, measured by the hosts. As depicted in Figure 3.27, the eSPID module, running on each host, analyses the outgoing flows to identify data streams with QoS requirements, like video and audio transport. E.g., once a host discovers a VoIP communication, it estimates the bandwidth requirements P_B for this particular flow using the SCBP algorithm.

The QoSiLAN MBAC algorithm decides upon the admission of the reservation, taking into account the different measures. If the admission was granted, a *QoSiLAN_REQUEST* message is sent to the QM to request the bandwidth reservation for the detected resources. As explained in 3.5, a *QoSiLAN_REQUEST* message contains the five-tuple; the sender and receiver IP- and port-addresses, as well as the estimated bandwidth.

The QM receives the *QoSiLAN_REQUEST* message and checks the map and the available resources for the flow's route within the LAN. If the requested resources are available, the QM sends individual *QoSiLAN_REQUEST* messages to each host in the network. These messages contain the QSpec parameters with the bandwidth limits to be obeyed by the receiver to the described target IP address. These messages are generated for each host and all relevant target LAN IP addresses individually, since the affected links on a route to another host in the LAN differ, depending on its location within the LAN's topology. Each QoSiLAN enabled host checks the request for validity and if the requested resources are available locally. In any instance of an error or a resource conflict, the host sends a negative acknowledgement *QoSiLAN_RESPONSE* message back, which results in a roll back of the QoS session. If the request is accepted, the host sends a positive acknowledgement *QoSiLAN_RESPONSE* message back to the QM. In the case of a negative acknowledgement, the QM tears the QoS session down, by sending corresponding *QoSiLAN_REQUEST* messages.

Once all QoSiLAN enabled hosts in the network respond with a positive acknowledgement message using the *QoSiLAN_RESPONSE* headers, the QM confirms the resource request by responding with a positive acknowledgement *QoSiLAN_RESPONSE* message to the requesting host. From this moment, the flow is protected by traffic shaping rules on all hosts in the network, which apply to all flows except the protected one. The reservation state is deleted either on soft-state time out or on end-of-stream discovery, like TCP FIN flag detection.

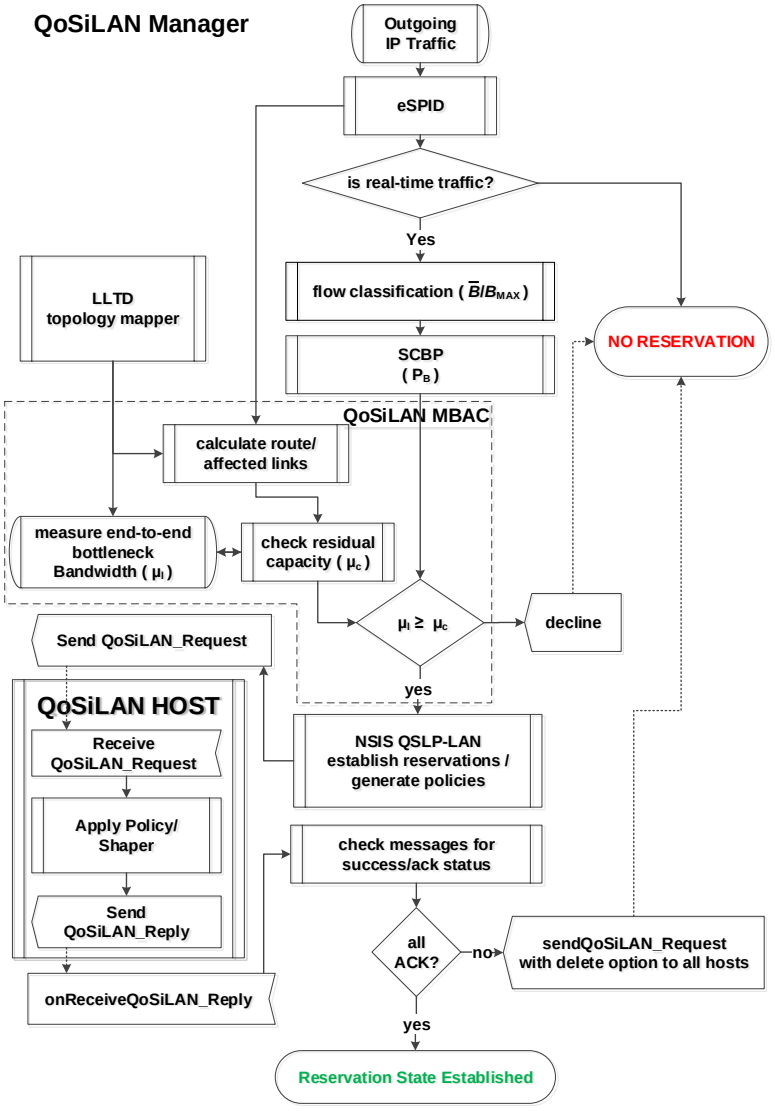


FIGURE 3.27: QoSiLAN Policing Information Flow Diagram

Figure 3.28 illustrates the resource allocation timeline, to give an overview of the timing of the algorithm. It shows the analysis phase in the first ten seconds, including the identification after 20 packets. The classification and prediction time $I = 10s$ is the time, when the eSPID algorithm identified

the flows protocol or application successfully and the signalling process starts. After 60 s, the average bandwidth occupation is used to update the reservation bandwidth for the identified flow. This is repeated every 60 s to keep the reservation state alive and to update the amount of resources.

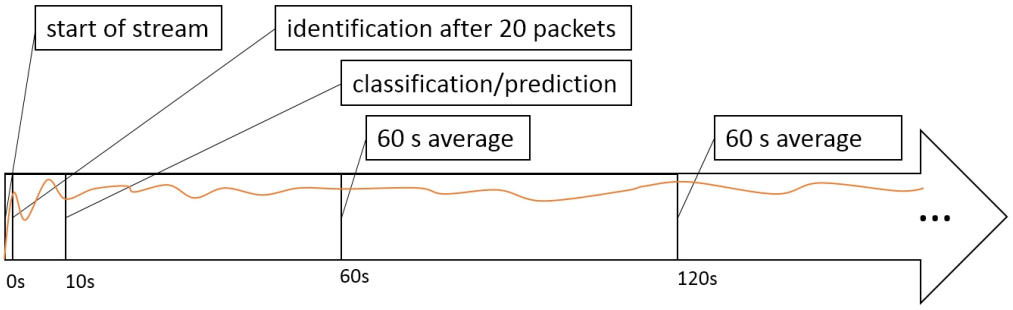


FIGURE 3.28: Resource Allocation Procedure Timeline

3.6.1.3 Admission Control

The proposed admission control algorithm works according to the principles of coordinated resource admission control [ETS08]. This approach aims to prevent uncontrolled overbooking of network resources. The QM host provides the function of a final decision point. It carries out priority considerations in terms of network resource availability based on client requests. It also takes care that resource reservation requests do not block best effort signalling traffic and that individual links are not over-booked, thus maintaining the residual capacity μ_r. According to comparative simulation results from S. Jamin et al. [JSD97a], a network utilisation rate of 77 % is achievable in a multi-hop scenario, with a utilisation target of 80 % $= 1 - \mu_r$ and no packet loss due to congestion. For this, I defined a threshold of $\mu_r = 20$ % residual capacity, which shall not be blocked by reservations and instead is reserved for best effort traffic. The algorithm in

(3.11) defines when a reservation will be denied by the QM,

$$\mu_l \geq \mu_c = p_b^{\alpha} + \mu_r + \sum_{i=1}^{N} \widehat{\nu}_i \tag{3.11}$$

if the sum of current reservation states $\widehat{\nu}$ for one of the affected connections/links including the predicted bandwidth p_b and residual capacity μ_r for a flow α exceeds the link capacity μ_l.

This algorithm enforces bandwidth allocation by collaborative shaping. The shaping task is left to the client's operating system functionality provided by well known Traffic Control (TC) APIs [Too15; Cor15c]. By default, the hosts in the network shape their traffic to the residual bandwidth, to protect the reservation's traffic from congestion. This works fine for a two-host network. The probability of multiple hosts using the full capacity of residual bandwidth and therefore exceeding it in sum, grows with every host joining the network. Therefore, an additional control function is needed to manage the residual bandwidth and to share this resource for best effort traffic among the hosts. A reactive approach, as proposed by N. Bayer et al. [Hoc+08] cannot be applied to this scheme, since the traffic characteristic is not predictable enough. As explained in section 3.4 the traffic is not continuous, but often bursty with pause periods in many cases. Therefore a QoS degradation by congestion is hard to detect by the receiver. Hence, the proposed algorithm was designed as described in the following.

Since each host in the network is monitoring its outgoing traffic continuously, it detects streams with significant best effort traffic throughput rates. A throughput rate is regarded as significant if the output rate r_o exceeds the residual capacity by more than 50 %. If this is the case, the host informs the QM about its current average best effort traffic output rate and the destination of this flow. The QM collects this information from all hosts and maintains it per link in the network map. The QM takes care that the

sum of best effort traffic rates from all hosts will not exceed μ_r for each single link in the LAN. If a reservation violation is detected by the QM, the affected hosts are advised to shape their traffic accordingly for traffic crossing the affected links. For this, the number of hosts utilising this link n_h is used to calculate the new residual bandwidth r_o for best-effort traffic throughput according to (3.12).

$$r_o \geq \frac{1}{n_h}\mu_r \tag{3.12}$$

3.6.2 Methodology

The policing and admission control algorithm was designed according to the best-practices found in literature and in existing systems. The literature research helped to identify the crucial parts and to select the appropriate approach. The concrete implementation of the algorithm was designed by employing the features of the different QoSiLAN framework's modules and to use them efficiently. The policing and admission control module is written in C++ employing APIs to all other modules and additionally accessing the Microsoft Windows Traffic Control API [Cor15c] and the Linux netfilter-tc [Cor15c] tools for active traffic management and control. The inter-module communication was realized through local socket communication or by calling exposed library APIs as presented in section 3.1.1.1.2.

The policing and admission control evaluations in section 3.6.3 show the integration of the whole QoSiLAN framework, its behaviour and the interaction of the individual key components. Therefore, the evaluation uses the complete scenario, as shown in Figure 3.2. A complete resource reservation procedure was selected, which drives the network into an overloaded situation with QoS degradation of the application streams with QoS demands. The evaluation setup was also designed to assess the Measurement Based

Admission Control (MBAC) and policing algorithm presented in section 3.6.1.3 in an demonstrative way.

3.6.3 Proof of Concept Evaluation

To evaluate the QoSiLAN framework's concept and the effectiveness of its admission and policing algorithms the scenario, as shown in Figure 3.29, was configured to overload the network for proving the system in a critical

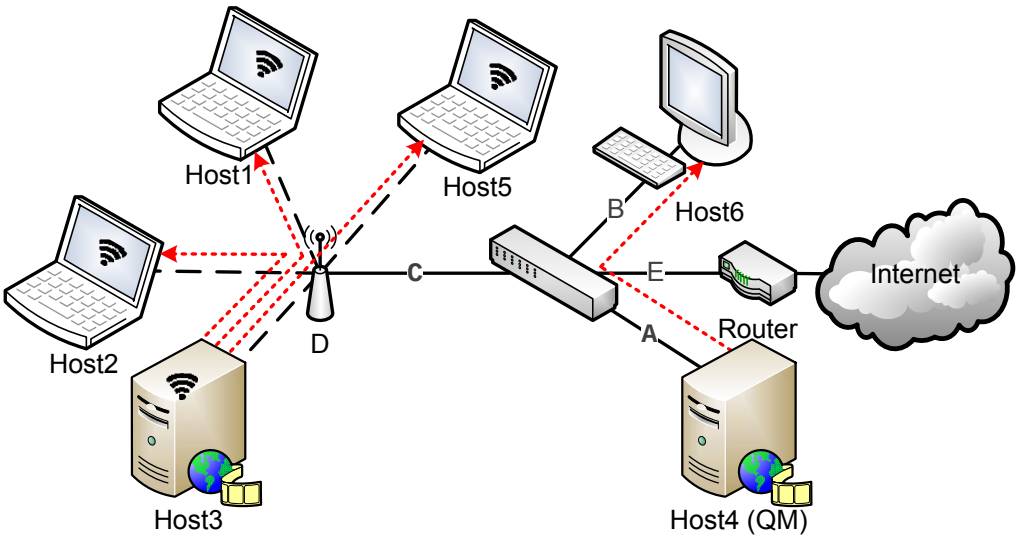


FIGURE 3.29: QoSiLAN Evaluation Scenario

situation. In addition, the features of hybrid inter-access-medium QoS, per link bandwidth reservation and simplex/duplex handling are shown. Host4 was set up as QM. Before the start of the evaluation, Host4 already performed the LLTD mapping process and tested the wireless link TCP throughput from the wireless Host3 to the other wireless Host1 with $\mu_l =$ 10.6 Mbps. This is a realistic throughput for IEEE 802.11g connections with both nodes connected to the same wireless access point due to the simplex nature of the access medium. The tested throughput from the fixed Host6 to the wireless Host1 was tested to be 20.6 Mbps. The maximum

TCP throughput between the fixed hosts Host4 and Host6 was tested to be 87.7 Mbps. The tests were carried out under congestion free network conditions and a very good link quality for the wireless hosts using the iperf TCP and UDP bandwidth performance measurement tool [ipe13]. At the beginning, the network was idle. For my evaluations, Host1 started requesting a video stream, called Stream1, from Host4 with an average bandwidth of 8 Mbps, as shown in Figure 3.30. To guide the reader, Table 3.4 shows a tabular view on the actions performed within the evaluation.

TABLE 3.4: Evaluation Action Schedule

time [s]	source host	target host	action
-10	4	4	LLTD Mapping
-5	4	any	Performance Measuremt
0	1	4	Request Stream1
0	4	1	Start of Stream1
2	4	local	SPID Identification [Stream1:Video]
10	4	local	SCBP Prediction (Stream1)
10	4	4	QoSiLAN_Req (Stream1)
10	4	any	QoSiLAN_Req (Stream1)
19	2	3	Request Stream2
19	3	2	Start of Stream2
20	3	local	SPID Identification [Stream2:None]
29	3	local	SCBP Prediction (Stream2)
29	3	4	QoSiLAN_Status (Stream2)
27	5	3	Request Stream3
27	3	5	Start of Stream3
28	3	local	SPID Identification [Stream3:Video]
37	3	local	SCBP Prediction (Stream3)
37	3	4	QoSiLAN_Req (Stream3)
37	4	any	QoSiLAN_Req (Stream3)
37	any	local	Apply new policy

After 20 packets the eSPID module identified the stream type as video and after 10 s the average output bandwidth p_b^1 of Stream1 was predicted to be 8.0 Mbps. The admission control algorithm decided to admit the

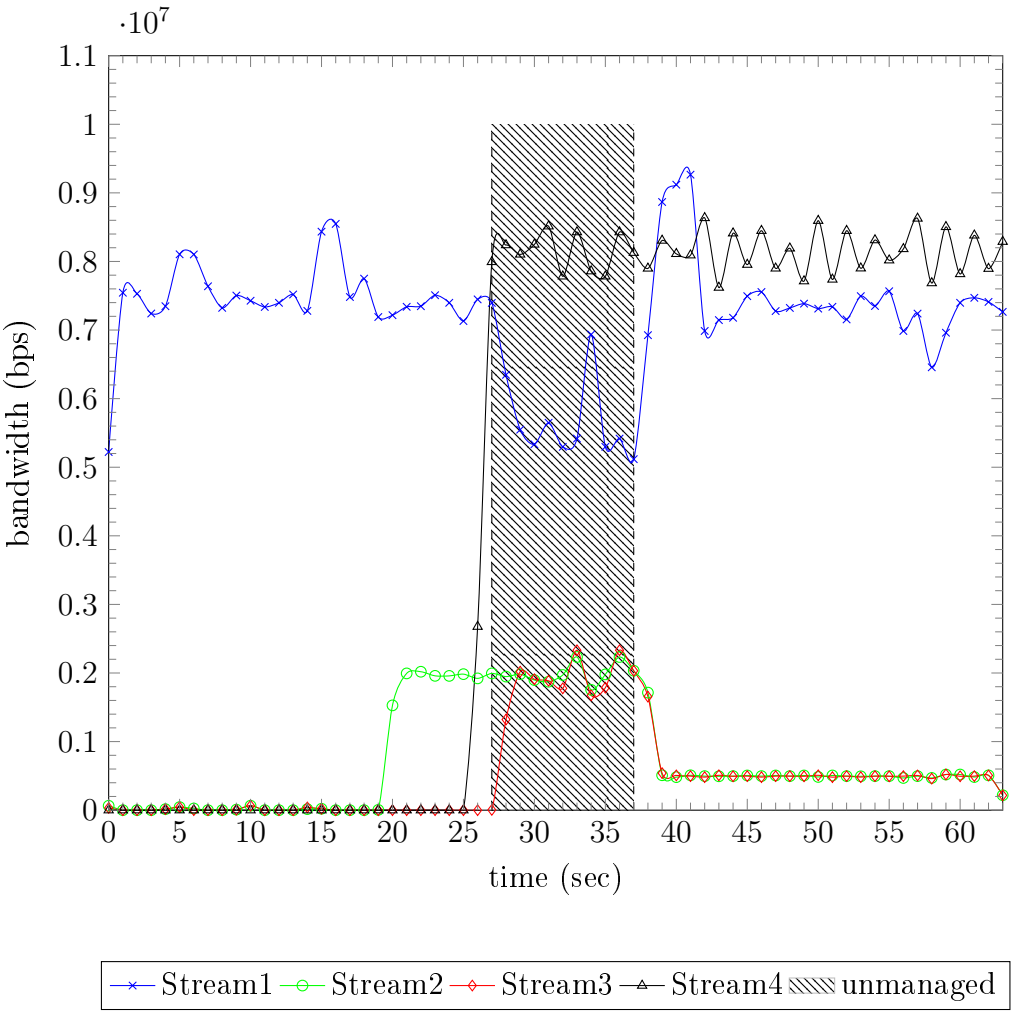


FIGURE 3.30: QoSiLAN Policing and Admission Control Evaluation

reservation, since it is in the range of 80 % link capacity 8.48 Mbps = $\mu_l \geq$ μ_c = 8.0 Mbps of the wireless link D. The QM initiated the QSLP-LAN signalling and advised all wireless hosts to shape their outgoing traffic to the other wireless hosts to μ_r = 2.12 Mbps residual bandwidth. 20 s after start, Host2 also requested TCP data for Stream2 from Host3 with a data rate of 2 Mbps, which was predicted with p_b^2 = 2.1 Mbps. This stream was not identified as audio or video stream and therefore no QoSiLAN reservation was initiated. The predicted bandwidth of Stream2 (p_b^2) was

smaller than the residual bandwidth μ_r thus causing no QoS problems. Although, for this flow no reservation was requested, the QM was informed about the current bandwidth occupation. After that, both streams were

TABLE 3.5: Traffic Shaping Policies

Source	**Host1**	**Host2**	**Host3**	**Host4**	**Host5**	**Host6**
Target				**Limit**		
Host1	-	500 Kbps	500 Kbps	1 Mbps	500 Kbps	500 Kbps
Host2	500 Kbps	-	500 Kbps	1 Mbps	500 Kbps	1 Mbps
Host3	500 Kbps	500 Kbps	-	1 Mbps	500 Kbps	1 Mbps
Host4	1 Mbps	1 Mbps	1 Mbps	-	1 Mbps	no limit
Host5	500 Kbps	500 Kbps	500 Kbps	1 Mbps	-	500 Kbps
Host6	1 Mbps	1 Mbps	1 Mbps	61.86 Mbps	1 Mbps	-

running in parallel without any disturbance or interference. When Host5 also requested 2.1 Mbps TCP traffic for Stream3 27 s after its start, the wireless link capacity is exceeded and the Stream1 is disturbed significantly, as shown in Figure 3.30. This situation represents an unmanaged state, as highlighted in Figure 3.30 as 'unmanaged' area, where no QoS is applied and congestion occurred. This situation lasted for 10 s. As soon as Host3 detected the significant outgoing traffic, it informed the QM about the amount. The QM detected the conflict and advised all wireless hosts to shape their outgoing traffic to the other wireless targets to 500 Kbps. This happened at 37 seconds, where the new policies were applied. Afterwards, the QoS protection for Stream1 was adapted again and Stream1 returned to its desired throughput level. For targets outside the wireless link D a shaping limit of 1 Mbps was communicated. As soon as this new policy was applied by the hosts, the throughput of Stream1 recovered to it's desired state, as depicted in Figure 3.30. In parallel, a video stream from Host4 to Host6, called Stream4, with $p_b^3 = 8.3$ Mbps was detected. Since no reservations were applied to the links A and B the reservation is admitted and does not affect the other reservations. This new reservation is also not communicated to the wireless hosts, since the bottleneck bandwidth for the path from the wireless hosts to Host6 is lower then the residual

bandwidth on link B. Host6 has no limits to Host4, due to the duplex nature of the Ethernet links and no reservations applied to that path. In addition, there is no need to communicate the Ethernet reservation to the wireless nodes, since their outgoing traffic limit is lower than the one on the Ethernet links. Host4 limits its outgoing traffic for other flows than Stream4 to $p_b^3 - \mu_r^A = 8.3\,\text{Mbps} - 70.16\,\text{Mbps} = 61.86\,\text{Mbps}$ according the policing rules. Table 3.5 shows the shaping policies as applied at the end of the evaluation, when all reservation states were active. As one can see, all hosts receive individual policies according to their location in the network. In this way per-link reservation states were enforced accurately.

3.6.4 Conclusion

The evaluation proved the QoSiLAN's concept using its MBAC algorithm for validity and operation. All components of the system worked together to provide QoS autonomously by collaborative resource reservation without network support. The LLTD Mapper detected the LAN's topology accurately. The eSPID algorithm identified the media correctly. The SCBP predicted the resource demand accurately. The QSLP-LAN communication was proven to signal the QoS demand, as well as the QoS session establishment operated successfully. The MBAC was proven to fulfil its requirements for resource management and reacted as expected to resource conflicts. The resource reservation worked seamlessly across access technologies. According to the system design, the QoS state establishment is reactive and therefore delayed, which results in a short period of degraded QoS, but after the QoSiLAN reservation is communicated to all hosts, the resource reservation provides a sufficient protection against congestion.

Chapter 4

Conclusion and Future Work

In this thesis a novel QoS framework was presented, which enables bandwidth resource reservations for self-organised QoS in unmanaged, hybrid, LANs on a per link basis, without required support from the network infrastructure. This cooperative and host-based QoS framework is enabled by multiple key technologies, which were presented, researched and improved to fit them for supporting the QoSiLAN framework functionality. Since this framework employs a broad range of enabler technologies, the presented work focused on providing a proof of concept to verify the functional operation and to investigate the strengths and weaknesses of the proposed QoS framework. Contributions were presented from the fields of Enhanced Statistical Protocol IDentification (eSPID), Statistical Class Based Bandwidth Prediction (SCBP), QoS Signalling Layer Protocol for Local Area Networks (QSLP-LAN), Link Layer Topology Discovery (LLTD) and Policing and Admission Control for Local Area Network (LAN) Management.

- eSPID: The eSPID algorithm was presented and successfully evaluated in section 3.3 for enabling self-organised and application independent QoS. An implementation was presented to analyse precisely

the outgoing traffic from the hosts to identify and classify protocol streams for QoS applications.

- SCBP: The SCBP algorithm was presented in section 3.4 for forecasting resource requirements, like the amount of needed bandwidth, accurately. It was successfully evaluated to support the self-organised work of the QoSiLAN framework in a smart manner.

- NSIS QSLP-LAN: The novel QSLP-LAN protocol for communication between the QoSiLAN nodes, supporting the new QoS framework was specified in section 3.5 in detail, to enable the QoS signalling and QoS related information exchange. It was evaluated to accomplish the task of cooperative, host based QoS management in LANs, although the evaluations revealed a scalability issue. A proposed solution to the scalability issue was analytically presented.

- LLTD Protocol: The application of the LLTD protocol and its reimplementation for use with the QoSiLAN framework was successfully shown in section 3.2. It was evaluated to discover and map the LAN's link layer topology, accurately.

- Policing and Admission Control for LAN Management: The QoSiLAN framework with its measurement-based and statistically collected input parameters requires an individual policing and admission control configuration to manage the QoS in the LAN efficiently. A MBAC algorithm was presented and proven in section 3.6.1.3 successfully to manage the LAN resources efficiently.

In this way, this work is a broad approach for link based resource management in unmanaged networks, which was presented to work autonomously without network infrastructure support in a host-cooperative manner.

4.1 Discussion

The QoSiLAN framework was presented to enable QoS resource reservation on a per link basis in an unmanaged network scenario. To accomplish this goal, multiple enabler technologies were needed as described in chapter 3. However, the QoSiLAN framework may not compete with other established QoS frameworks, as investigated in section 2.3, in terms of reservation set-up time and other omitted QoS optimisation parameters like delay, packet loss or jitter, particularly not with those following a cross layer approach, involving lower layer information and control. However, these QoS parameters are also improved for flows with applied QoSiLAN reservations, but only in an implicit manner, since the QoSiLAN framework takes care to leave enough residual bandwidth and therefore prevents packet buffer overflows. By design, the QoSiLAN framework cannot influence the transmission parameters on single links in the network, since network infrastructure elements and certain access technologies are out of scope of the proposed framework. Nevertheless, if the LAN infrastructure supports e.g., DiffServ functionality, packets of flows which are admitted for a QoSiLAN QoS reservation can be marked with DiffServ flags also by the QoSiLAN stack.

Another advantage of the QoSiLAN framework is the deployment flexibility, which allows the applied QoS framework to work, even if not all hosts in the network support or implement the QoSiLAN framework. E.g., if the gateway is QoSiLAN enabled, but the hosts not, traffic from and to the Internet can be managed by that gateway and resources can be reserved on behalf of the hosts by the gateway. If within the LAN only a media Network Attached Storage (NAS) server supports the QoSiLAN stack, but the clients do not, this server will manage the traffic it is involved with using the QoSiLAN stack. In this scenario, clients do not need to support the framework essentially. The same applies the other way around, where

clients support the QoSiLAN stack, but the server does not. The most critical situation appears when two non-QoSiLAN enabled hosts in a switched LAN communicate directly using shared links. In this case, only passive bottleneck detection and QoS monitoring may discover a QoS problem for other flows, but there is no possibility of influencing this unmanaged traffic or to prevent a reservation state congestions.

4.2 Future Work

The presented research focused on the investigation and development of a QoS framework to enable network bandwidth capacity management in unmanaged infrastructures. The research of the QoSiLAN framework should be continued in order to optimise the performance and its usefulness.

The eSPID algorithm leaves room for optimisation for special sets of features and measures with a more specialised set of protocols. In addition, it should be evaluated, if targeting for more general flow characteristics like identifying streaming, real-time transmissions, gaming or file transfers rather than targeting individual protocols, which would need new learning each time a new protocol version or novel media codec appears. As it is a semi-supervised machine learning algorithm, it might be further extended to an unsupervised algorithm, which autonomously identifies new protocols and classifies new flows accurately.

The SCBP resource estimation algorithm might be improved to forecast the required resources even more precisely and earlier. The requirement of the current proposed solution for a 10 s capture time is crucial. A faster estimation would prevent short-time QoS harm. As the NSIS QSLP-LAN evaluations have shown, the proposed protocol design has a scalability problem, which should be further investigated. The proposed optimisation for message payload stripping and consolidation should be further evaluated as

well as the NSIS approach should be re-engineered to find a solution with improved scalability behaviour.

The physical topology discovery can be further improved to support more network topology structures and new kinds of access technology. In addition, the proposed active measurement based bandwidth capacity discovery method should be extended with a passive capacity monitoring approach as Katabi et al. or [KBY01] and Guo et. al. [Sch+14] propose. The QoSiLAN framework's policing and admission control algorithms can be further developed to manage the resources in an even smarter manner and to use more measures like CPU or memory resources or even other network measures.

The unconsidered topic of security mechanisms for the QoSiLAN framework leaves room for further research and investigation to design a security schema with authorisation and authentication of hosts, which protects QoSiLAN enabled hosts from malicious resource request by attackers, but preserves the auto-configuration ability of the framework, which should not rely on user interaction.

In addition, the QoSiLAN framework provides a technology base for smart network analysis. Since many nodes in the LAN support per flow QoS monitoring and analysis, the collected measures could be exchanged between hosts or be collected in a database by the QoSiLAN Manager (QM) to compare QoS properties on an per flow and per host basis. This enables the framework to identify and locate QoS relevant network issues on single links. For example, this could be used to tell the user in case of non-recoverable QoS problems which link or host may be responsible for the current problem. Is the lack of QoE the responsibly of the external provider/server or is it located within the own LAN? If the problem is located in the LAN, the framework can tell what the reason is, like another active reservation or traffic sources which might be active in the LAN and which user or host is causing it. If there is a link capacity or bottleneck

problem in the LAN, the QoSiLAN framework could propose dedicated sophisticated hardware upgrade recommendations.

Further, it should be investigated, whether the QoSiLAN framework could help support departments from network providers to improve remote assistance to identify and solve customer's network problems, since it analyses and maintains many statistical and qualitative data about the state of the network.

Chapter 5

Bibliography

References

[Ada+11] Florian Adamsky, Christopher Kohnen, Christian Uberall, Veselin Rakocevic, Muttukrishnan Rajarajan, and Rudolf Jager. "A Novel Concept For Hybrid Quality Improvements in Consumer Networks". In: *Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on*. IEEE. 2011, pp. 39–43.

[AGG04] The Architecture, Transport Working Group, and DSL Home Technical Working Group. *Multi-Service Delivery Framework for Home Networks*. DSL Forum, Technical Report. Aug. 2004.

[All15] Open Services Gateway initiative Alliance. *Open Services Gateway initiative: The Dynamic Module System for Java*. OSGi Specification. `https://www.osgi.org/`, last checked: 30. 10. 2015. 2015.

[AM14] S. Aroussi and A. Mellouk. "Survey on machine learning-based QoE-QoS correlation models". In: *Computing, Management and Telecommunications (ComManTel), 2014 International Conference on*. Apr. 2014, pp. 200–204. DOI: 10.1109/ComManTel.2014.6825604.

[Apo+99] G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, and T. Przygienda. *QoS Routing Mechanisms and OSPF Extensions*. RFC 2676 (Experimental). Internet Engineering Task Force, Aug. 1999. URL: http://www.ietf.org/rfc/rfc2676.txt.

[Ash+10] G. Ash, A. Bader, C. Kappler, and D. Oran. *QSPEC Template for the Quality-of-Service NSIS Signaling Layer Protocol (NSLP)*. RFC 5975 (Experimental). Internet Engineering Task Force, Oct. 2010. URL: http://www.ietf.org/rfc/rfc5975.txt.

[Awd+01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. *RSVP-TE: Extensions to RSVP for LSP Tunnels*. RFC 3209 (Proposed Standard). Updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711, 6780, 6790. Internet Engineering Task Force, Dec. 2001. URL: http://www.ietf.org/rfc/rfc3209.txt.

[BA10] O.T. Brewer and A. Ayyagari. "Comparison and analysis of measurement and parameter based admission control methods for Quality of Service (QoS) provisioning". In: *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*. 2010, pp. 184–188. DOI: 10.1109/MILCOM.2010.5679561.

[BCS94] R. Braden, D. Clark, and S. Shenker. *Integrated Services in the Internet Architecture: an Overview*. RFC 1633 (Informational). Internet Engineering Task Force, June 1994. URL: http://www.ietf.org/rfc/rfc1633.txt.

[BDF04] R. Black, A. Donnelly, and C. Fournet. "Ethernet topology discovery without network assistance". In: *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*. Oct. 2004, pp. 328–339. DOI: 10.1109/ICNP.2004.1348122.

[Bej+03] Yigal Bejerano, Yuri Breitbart, Minos N. Garofalakis, and Rajeev Rastogi. "Physical Topology Discovery for Large Multi-Subnet Networks". In: *in Proc. IEEE Infocom*. 2003, pp. 342–352.

[BJ00] A. Bierman and K. Jones. *Physical Topology MIB*. RFC 2922 (Informational). Internet Engineering Task Force, Sept. 2000. URL: http://www.ietf.org/rfc/rfc2922.txt.

[Bla+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services*. RFC 2475 (Informational). Updated by RFC 3260. Internet Engineering Task Force, Dec. 1998. URL: http://www.ietf.org/rfc/rfc2475.txt.

[BMR99] N. Brownlee, C. Mills, and G. Ruth. *Traffic Flow Measurement: Architecture*. RFC 2722 (Informational). Internet Engineering Task Force, Oct. 1999. URL: http://www.ietf.org/rfc/rfc2722.txt.

[BNW03] R. Bless, K. Nichols, and K. Wehrle. *A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services*. RFC 3662 (Informational). Internet Engineering Task Force, Dec. 2003. URL: http://www.ietf.org/rfc/rfc3662.txt.

[Boo88] Blue Book. *G.711 : Pulse code modulation (PCM) of voice frequencies*. public document. http://www.itu.int/rec/T-REC-G.711-198811-I/en. 1988.

[BPW13] M. Boucadair, R. Penno, and D. Wing. *Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF)*. RFC 6970 (Proposed Standard). Internet Engineering Task Force, July 2013. URL: http://www.ietf.org/rfc/rfc6970.txt.

[Bra+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*. RFC 2205 (Proposed Standard). Updated by RFCs 2750, 3936, 4495, 5946, 6437, 6780. Internet Engineering Task Force, Sept. 1997. URL: http://www.ietf.org/rfc/rfc2205.txt.

[Bre+11] L. Brewka, P. Skoldström, J. Nelis, H. Wessing, and C. Develder. *Automatic provisioning of end-to-end QoS into the home*. Nov. 2011. DOI: 10.1109/TCE.2011.6131140.

[Bri09] Robert Briel. "German IPTV numbers set to grow". In: *Broadband TV News* (June 2009). http://www.broadbandtvnews.com/2009/06/17/german-iptv-numbers-set-to-grow/.

[BZ97] R. Braden and L. Zhang. *Resource ReSerVation Protocol (RSVP) – Version 1 Message Processing Rules*. RFC 2209 (Informational). Internet Engineering Task Force, Sept. 1997. URL: http://www.ietf.org/rfc/rfc2209.txt.

[CAC] Davis (California) CACE Technologies. *WinPcap: The Windows Packet Capture Library*. URL: http://www.winpcap.org/.

[CAS+10] Marco CASTRUCCI, Guido ODDI, Gabriele TAMEA, and Vincenzo SURACI. *Application QoS Management and Session Control in a Heterogeneous Home Network using Inter-MAC layer support*. Future Network and Mobile Summit 2010 Conference Proceedings. 2010.

[CCC07] Jiann-Liang Chen, Ming-Chiao Chen, and Yi-Ru Chian. *QoS management in heterogeneous home networks*. Computer Networks 51, Elsevier B.V. Jan. 2007.

[Cho+03] Tim Chown, Tiziana Ferrari, Simon Leinen, Roberto Sabatino, Nicolas Simar, and Stig Venaas. "Less Than Best Effort: Application Scenarios and Experimental Results". In: *Proceedings of the Second International Workshop on Quality of Service in Multiservice IP Networks*. QoS-IP 2003. London, UK, UK: Springer-Verlag, 2003, pp. 131–144. ISBN: 3-540-00604-4. URL: http://dl.acm.org/citation.cfm?id=646464.693899.

[CIS14] CISCO. *Cisco Discovery Protocol Configuration Guide, Cisco IOS Release 15M&T*. http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cdp/configuration/15-mt/cdp-15-mt-book.pdf, last checked: 30. 03. 2015. Cisco Systems, Inc. 2014.

[CK13] S. Cheshire and M. Krochmal. *Multicast DNS*. RFC 6762 (Proposed Standard). Internet Engineering Task Force, Feb. 2013. URL: http://www.ietf.org/rfc/rfc6762.txt.

[CLG95] Song Chong, San-qi Li, and J. Ghosh. "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM". In: *Selected Areas in Communications, IEEE Journal on* 13.1 (Jan. 1995), pp. 12–23.

[Con02] A.E. Conway. "A passive method for monitoring voice-over-IP call quality with ITU-T objective speech quality measurement methods". In: *Communications, 2002. ICC 2002. IEEE International Conference on*. Vol. 4. 2002, 2583–2586 vol.4.

[Cor02] Microsoft Corporation. *The Microsoft .NET Framework*. online resource. http://www.microsoft.com/net. 2002.

[Cor09] Microsoft Corporation. *"[MS-LLTD]: Link Layer Topology Discovery (LLTD) Protocol Specification"*. Open Specifications Documentation. `https://msdn.microsoft.com/en-us/library/cc233983.aspx`. 2009.

[Cor13] Microsoft Corporation. *RSVP Service*. `https://msdn.microsoft.com/en-us/library/windows/desktop/aa374144.aspx`, last checked: 30. 10. 2015. 2013.

[Cor15a] Microsoft Corporation. *Excel Application*. `https://products.office.com/en-us/Excel`, last checked: 30. 03. 2015. 2015.

[Cor15b] Microsoft Corporation. *"[MS-MMSP]: Microsoft Media Server (MMS) Protocol"*. Open Specifications Documentation. `https://msdn.microsoft.com/en-us/library/cc234711.aspx`. 2015.

[Cor15c] Microsoft Corporation. *Traffic Control Application Programming Interface (TC API)*. online resource. `https://msdn.microsoft.com/en-us/library/windows/desktop/aa374468.aspx`. 2015.

[CR01] R. G. Cole and J. H. Rosenbluth. "Voice over IP performance monitoring". In: *SIGCOMM Comput. Commun. Rev.* 31.2 (2001). `http://doi.acm.org/10.1145/505666.505669`, pp. 9–24.

[Cra+98] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. *A Framework for QoS-based Routing in the Internet*. RFC 2386 (Informational). Internet Engineering Task Force, Aug. 1998. URL: `http://www.ietf.org/rfc/rfc2386.txt`.

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". English. In: *Machine Learning* 20.3 (1995), pp. 273–297. ISSN: 0885-6125. DOI: `10.1007/BF00994018`. URL: `http://dx.doi.org/10.1007/BF00994018`.

[DA99] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*. RFC 2246 (Proposed Standard). Obsoleted by RFC 4346, updated by RFCs 3546, 5746, 6176. Internet Engineering Task Force, Jan. 1999. URL: http://www.ietf.org/rfc/rfc2246.txt.

[De 05] Annie De Montigny-Leboeuf. "Flow attributes for use in traffic characterization". In: *Communications Research Centre Canada, Tech. Rep* (2005).

[Dor] Technical University Dortmund. *HOMEPLANE - Home Media Platform and Networks*. Research Project. http://www.homeplane.org/, last checked: 30.03.2015.

[Dow99] Allen B. Downey. "Using pathchar to estimate Internet link characteristics". In: *SIGCOMM Comput. Commun. Rev.* 29.4 (1999), pp. 241–250.

[DR06] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.1*. RFC 4346 (Proposed Standard). Obsoleted by RFC 5246, updated by RFCs 4366, 4680, 4681, 5746, 6176. Internet Engineering Task Force, Apr. 2006. URL: http://www.ietf.org/rfc/rfc4346.txt.

[Dro97] R. Droms. *Dynamic Host Configuration Protocol*. RFC 2131 (Draft Standard). Updated by RFCs 3396, 4361, 5494, 6842. Internet Engineering Task Force, Mar. 1997. URL: http://www.ietf.org/rfc/rfc2131.txt.

[EAM06] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. "Traffic classification using clustering algorithms". In: *MineNet '06: Proceedings of the 2006 SIGCOMM workshop on Mining network data*. Pisa, Italy: ACM, 2006, pp. 281–286. ISBN: 1-59593-569-X. DOI: http://doi.acm.org/10.1145/1162678.1162679.

[EKJ05] W. Endemann, R. Kays, and K. Jostschulte. "Practical limitations of Ethernet-based inhouse multimedia distribution". In: *Consumer Electronics, IEEE Transactions on* 51.2 (May 2005), pp. 507–513.

[Erm+07] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson. "Offline/realtime traffic classification using semi-supervised learning". In: *Performance Evaluation* 64.9-12 (Oct. 2007), pp. 1194–1213. DOI: http://dx.doi.org/10.1016/j.peva.2007.06.014.

[Est+96] Martin Ester, Hans-peter Kriegel, Jörg S, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *A density-based algorithm for discovering clusters in large spatial databases with noise*. AAAI Press, 1996, pp. 226–231.

[ETS08] ETSI. *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control Sub-System (RACS): Functional Architecture (ES 282 003)*. European Standard Telecommunication Series, May 2008.

[Eur15] DigitalTV Europe. "German TV growth boosts Deutsche Telekom revenues". In: *DigitalTVEurope.net* (May 2015). http://www.digitaltveurope.net/366671/german-tv-growth-boosts-deutsche-telekom-revenues/.

[FAV08] A. Farrel, A. Ayyangar, and JP. Vasseur. *Inter-Domain MPLS and GMPLS Traffic Engineering – Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*. RFC 5151 (Proposed Standard). Internet Engineering Task Force, Feb. 2008. URL: http://www.ietf.org/rfc/rfc5151.txt.

[Fay+96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996. ISBN: 0-262-56097-6.

[Fen+06] Huifang Feng, Yantai Shu, Shuyi Wang, and Maode Ma. "SVM-Based Models for Predicting WLAN Traffic". In: *Communications, 2006. ICC '06. IEEE International Conference on*. Vol. 2. June 2006, pp. 597–602.

[Fer+11] Jon Ferraiolo, Anthony Grasso, Jun Fujisawa, Jonathan Watt, Chris Lilley, Cameron McCormack, Dean Jackson, Erik Dahlström, Doug Schepers, and Patrick Dengler. *Scalable Vector Graphics (SVG) 1.1 (Second Edition)*. W3C Recommendation. `http://www.w3.org/TR/2011/REC-SVG11-20110816/`. W3C, Aug. 2011.

[FM10] R.H. Filho and J.E.B. Maia. "Network traffic prediction using PCA and K-means". In: *Network Operations and Management Symposium (NOMS), 2010 IEEE*. Apr. 2010, pp. 938–941.

[For06] UPnP Forum. *For UPnP Version 1.0, UPnP-QoS Architecture v2*. `http://www.upnp.org/specs/qos/UPnP-qos-Architecture-v2-20061016.pdf`, last checked: 30.09.2015. 2006.

[For14] UPnP Forum. *UPnP-QoS Architecture v3, For UPnP Version 1.0*. `http://upnp.org/specs/qos/UPnP-qos-Architecture-v3.pdf`, last checked: 30.09.2015. 2014.

[Fou15a] Open Networking Foundation. *OpenFlow Switch Specification Version 1.5.0*. `https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf`. Open Networking Foundation. 2015.

[Fou15b] Wireshark Foundation. *Wireshark*. https://www.wireshark.orglast checked: 30. 03. 2015. 2015.

[Fri02] Jeffrey E. F. Friedl. *Mastering Regular Expressions*. Ed. by Andy Oram. 2nd ed. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2002. ISBN: 0596002890.

[G A09] et. al. G. Ash. *QoS NSLP QSPEC Template, Internet Draft (draft-ietf-nsis-qspec-21), Work in Progress*. 2009. URL: http://tools.ietf.org/html/draft-ietf-nsis-qspec-21.

[G1008] G.107. *The E-model: a computational model for use in transmission planning*. prepublished version. Aug. 2008.

[Gha05] S. Ghahramani. *Fundamentals of Probability: With Stochastic Processes*. Pearson/Prentice Hall, 2005. ISBN: 9780131453401. URL: http://books.google.ca/books?id=M1GUQgAACAAJ.

[Goo02] B. Goode. "Voice over Internet protocol (VoIP)". In: *Proceedings of the IEEE* 90.9 (Sept. 2002), pp. 1495–1517.

[Gra13a] Graphviz. *Graph Visualization Software*. http://www.graphviz.org/Documentation.php, last checked: 30. 10. 2015. 2013.

[Gra13b] Graphviz. *The DOT Language*. "http://www.graphviz.org/content/dot-language, last checked: 30. 10. 2015. 2013.

[Gro+96] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications*. RFC 1889 (Proposed Standard). Obsoleted by RFC 3550. Internet Engineering Task Force, Jan. 1996. URL: http://www.ietf.org/rfc/rfc1889.txt.

[Gro02] D. Grossman. *New Terminology and Clarifications for Diffserv*. RFC 3260 (Informational). Internet Engineering Task Force, Apr. 2002. URL: http://www.ietf.org/rfc/rfc3260.txt.

[Gro07] Object Management Group. *Data Distribution Service for Real-time systems*. Object Management Group. Jan. 2007. URL: http://www.omg.org/.

[Haf+05] Patrick Haffner, Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang. "ACAS: automated construction of application signatures". In: *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*. Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 197–202. ISBN: 1-59593-026-4. DOI: http://doi.acm.org/10.1145/1080173.1080183.

[Han+05] R. Hancock, G. Karagiannis, J. Loughney, and S. Van den Bosch. *Next Steps in Signaling (NSIS): Framework*. RFC 4080 (Informational). Internet Engineering Task Force, June 2005. URL: http://www.ietf.org/rfc/rfc4080.txt.

[HBA14] Amira Y Haikal, M Badawy, and Hesham A Ali. *Towards Internet QoS Provisioning Based on GenericDistributed QoS Adaptive Routing Engine*. Scientific World Journal. 2014. URL: http://www.biomedsearch.com/nih/Towards-Internet-QoS-Provisioning-Based/25309955.html.

[HDA05] Qi He, Constantine Dovrolis, and Mostafa Ammar. "On the predictability of large transfer TCP throughput". In: *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. SIGCOMM '05. Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 145–156. ISBN: 1-59593-009-4. DOI: 10.1145/1080091.1080110. URL: http://doi.acm.org/10.1145/1080091.1080110.

[HJ09] Erik Hjelmvik and Wolfgang John. "Statistical Protocol IDentification with SPID: Preliminary Results". In: *Swedish National Computer Networking Workshop*. last checked: 30. 10. 2015.

2009. URL: http://spid.sourceforge.net/sncnw09-hjelmvik_john-CR.pdf.

[Hoc+08] D. Hock, N. Bayer, R. Pries, M. Siebert, D. Staehle, V. Rakocevic, and B. Xu. "QoS provisioning in WLAN mesh networks using dynamic bandwidth control". In: *Wireless Conference, 2008. EW 2008. 14th European*. 2008, pp. 1–7. DOI: 10.1109/EW.2008.4623896.

[Hun+07] O. Hundt, R. Kays, B. Aznar, W. Endemann, and C. Schilling. "Methods to Improve the Efficiency of Wireless LAN for Multimedia Home Networks". In: *Consumer Electronics, IEEE Transactions on* 53.2 (May 2007), pp. 397–404.

[IEE05] IEEE. *IEEE 802.1AB Standard for Local and metropolitan area networks Station and Media Access Control Connectivity Discovery*. Institute of Electrical and Electronics Engineers, Inc. May 2005.

[Ini06] Home Gateway Initiative. *Home Gateway Technical Requirements, Release 1*. HGI, Technical Report. July 2006.

[ipe13] iperf. *TCP and UDP bandwidth performance measurement tool*. https://code.google.com/p/iperf/, last checked: 30. 03. 2015. 2013.

[IPT06] ITU-T Focus Group on IPTV standardization. "Classification of IPTV services based on network QoS requirements". In: *2nd FG IPTV meeting* (2006). http://www.itu.int/md/T05-FG.IPTV-C-0127/en.

[IPT09] ITU-T Focus Group on IPTV standardization. "H.323 : Packet-based multimedia communications systems". In: *ITU Recommendation* (2009). http://www.itu.int/rec/T-REC-H.323-200912-I.

[ISO14] ISO/IEC. *Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats (23009-1:2014)*. Standard. Geneva, CH: International Organization for Standardization, May 2014.

[ISO15] ISO/IEC. *Information technology – Generic coding of moving pictures and associated audio information – Part 1: Systems (13818-1:2015)*. Standard. Geneva, CH: International Organization for Standardization, May 2015.

[ITU07] ITU. *Architecture of MediaHomeNet (Recommendation ITU-T J.190, 2nd Edition)*. International Telecommunications Union. July 2007.

[ITU08] ITU. *A generic home network architecture with support for multimedia services (Recommendation ITU-T H.622)*. International Telecommunications Union. June 2008.

[J M09] A. McDonald J. Manner G. Karagiannis. *NSLP for Quality-of-Service Signaling (QoS NSLP), Internet Draft (draft-ietf-nsis-qos-nslp-16), Work in Progress*. 2009. URL: `http://tools.ietf.org/html/draft-ietf-nsis-qos-nslp-16`.

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN: 0-13-022278-X.

[Jia+07] Hongbo Jiang, Andrew W. Moore, Zihui Ge, Shudong Jin, and Jia Wang. "Lightweight application classification for network management". In: *INM '07: Proceedings of the 2007 SIGCOMM workshop on Internet network management*. Kyoto, Japan: ACM, 2007, pp. 299–304. ISBN: 978-1-59593-788-9. DOI: `http://doi.acm.org/10.1145/1321753.1321771`.

[JSD97a] S. Jamin, S.J. Shenker, and P.B. Danzig. "Comparison of measurement-based admission control algorithms for controlled-load service". In: *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*. Vol. 3. 1997, 973–980 vol.3. DOI: 10.1109/INFCOM.1997.631035.

[JSD97b] Sugih Jamin, Scott Shenker, and PB Danzig. "Measurement-based admission control algorithms for controlled-load service: A structural examination". In: *Univ. of Michigan, Ann Arbor, MI, Rep. CSE-TR-333-97* (1997).

[Kar+04] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. "Transport layer identification of P2P traffic". In: *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. Taormina, Sicily, Italy: ACM, 2004, pp. 121–134. ISBN: 1-58113-821-0. DOI: http://doi.acm.org/10.1145/1028788.1028804.

[KBY01] D. Katabi, I. Bazzi, and Xiaowei Yang. "A passive approach for detecting shared bottlenecks". In: *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*. 2001, pp. 174–181. DOI: 10.1109/ICCCN.2001.956236.

[Kil08] K. Kilkki. "Quality of Experience in Communications Ecosystem". In: *Journal of Universal Computer Science* 14.5 (2008). http://www.jucs.org/jucs_14_5/quality_of_experience_in, pp. 615–624.

[KL51] Solomon Kullback and Richard A. Leibler. "On information and sufficiency". In: *Annals of Mathematical Statistics* 22 (1951), pp. 49–86.

[Koe+10a] Christopher Koehnen, Christian Uberall, Veselin Rakocevic, Muttukrishnan Rajarajan, and Rudolf Jaeger. "QoSiLAN - A Heterogeneous Approach to Quality of Service in Local Area Networks". In: *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*. IEEE. 2010, pp. 109–112.

[Koe+10b] Christopher Koehnen, Christian Ueberall, Florian Adamsky, Veselin Rakocevic, Muttukrishnan Rajarajan, and Rudolf Jaeger. "Enhancements to Statistical Protocol IDentification (SPID) for Self-Organised QoS in LANs". In: *ICCCN 2010 Track on Network Algorithms, Performance Evaluation and Theory (NAPET) (ICCCN 2010 NAPET)*. Zurich, Switzerland, Aug. 2010.

[Koe+15] Christopher Koehnen, Christian Ueberall, Muttukrishnan Rajarajan, Rudolf Jaeger, and Veselin Rakocevic. "Autonomous QoS Management and Policing in Unmanaged Local Area Networks". In: *Journal of Computer Networks and Communications* 2015 (2015).

[Koz05] Charles M. Kozierok. *TCP/IP Guide*. NO STARCH PRESS, 2005.

[Kum+06] Sailesh Kumar, Sarang Dharmapurikar, Fang Yu, Patrick Crowley, and Jonathan Turner. "Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection". In: *SIGCOMM Comput. Commun. Rev.* 36.4 (Aug. 2006), pp. 339–350. ISSN: 0146-4833. DOI: 10.1145/1151659.1159952. URL: http://doi.acm.org/10.1145/1151659.1159952.

[LD13] Steven Latre and Filip De Turck. "Joint In-network Video Rate Adaptation and Measurement-Based Admission Control: Algorithm Design and Evaluation". English. In: *Journal of Network and Systems Management* 21.4 (2013), pp. 588–622. ISSN: 1064-7570. DOI: 10.1007/s10922-012-9255-z. URL: http://dx.doi.org/10.1007/s10922-012-9255-z.

[LH09] J. Laulajainen and M. Hirvonen. "Automatic QoS control in UPnP home networks". In: *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*. July 2009, pp. 455–460. DOI: 10.1109/ISCC.2009.5202276.

[LLC98] Hwa-Chun Lin, Shou-Chuan Lai, and Ping-Wen Chen. "An algorithm for automatic topology discovery of IP networks". In: *Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on*. Vol. 2. June 1998, 1192–1196 vol.2.

[LLG14] Chi Harold Liu, Kin K Leung, and Athanasios Gkelias. "A Generic Admission-Control Methodology for Packet Networks". In: *Wireless Communications, IEEE Transactions on* 13.2 (2014), pp. 604–617.

[LMK07] Hyunyong Lee, SungTae Moon, and JongWon Kim. *Enhanced UPnP QoS Architecture for Network-adaptive Streaming Service in Home Networks*. Aug. 2007. DOI: 10.1109/TCE.2007.4341563.

[Lou+10] Maxime Louvel, Jacques Pulou, Alain Plantec, and Jean-Philippe Babau. "Quantity of Resource aggregation for heterogeneous resource reservation for multimedia applications". In: *ETFA*. 2010, pp. 1–4.

[Lou+11] Maxime Louvel, Pierre Bonhomme, Jean-Philippe Babau, and Alain Plantec. "A Network Resource Management Framework for Multimedia Applications Distributed in Heterogeneous Home Networks". In: *AINA*. 2011, pp. 724–731.

[LPB13] Maxime Louvel, Alain Plantec, and Jean-Philippe Babau. "Resource management for multimedia applications, distributed in open and heterogeneous home networks". In: *Journal of Systems Architecture - Embedded Systems Design* 59.3 (2013), pp. 121–134.

[Man+05] Jukka Manner, Tapio Suihko, Markku Kojo, and Mika Liljeberg. *Kimmo Raatikainen. Localized rsvp. Internet draft (work in progress) draft-manner-lrsvp-04, Internet Engineering Task Force.* 2005. URL: `http://tools.ietf.org/html/draft-manner-lrsvp-04`.

[Man+10] J. Manner, R. Bless, J. Loughney, and E. Davies. *Using and Extending the NSIS Protocol Family.* RFC 5978 (Informational). Internet Engineering Task Force, Oct. 2010. URL: `http://www.ietf.org/rfc/rfc5978.txt`.

[MF05] J. Manner and X. Fu. *Analysis of Existing Quality-of-Service Signaling Protocols.* RFC 4094 (Informational). Internet Engineering Task Force, May 2005. URL: `http://www.ietf.org/rfc/rfc4094.txt`.

[Mic04] Richard Black et.al at Microsoft Research. *Ethernet Topology Discovery without Network Assistance.* Proceeding. 2004.

[MKM10] J. Manner, G. Karagiannis, and A. McDonald. *NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling.* RFC 5974 (Experimental). Internet Engineering Task Force, Oct. 2010. URL: `http://www.ietf.org/rfc/rfc5974.txt`.

[MN02] Vincenzo Mancuso and Giovanni Neglia. *Performance Improvements on Self-Similar Traffic Using Measurement-Based Admission Control.* "`http://www-sop.inria.fr/members/Vincenzo.Mancuso/MNB02.pdf`", last checked: 30. 10. 2015. 2002.

[Moo02] Andrew William Moore. *Measurement-based management of network resources.* Tech. rep. PhD dissertation, submitted to the University of Cambridge, Corpus Christi College. June 2002. URL: `http://www.cl.cam.ac.uk/~awm22/publications/moore2002phd.pdf`.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. http://nlp.stanford.edu/IR-book/information-retrieval-book.html. Cambridge University Press, 2008. ISBN: 0521865719.

[MZ05] Andrew W. Moore and Denis Zuev. "Internet traffic classification using bayesian analysis techniques". In: *SIGMETRICS Perform. Eval. Rev.* 33.1 (2005), pp. 50–60. ISSN: 0163-5999. DOI: http://doi.acm.org/10.1145/1071690.1064220.

[NA08] Thuy TT Nguyen and Grenville Armitage. "A survey of techniques for internet traffic classification using machine learning". In: *Communications Surveys & Tutorials, IEEE* 10.4 (2008), pp. 56–76.

[Net15] Deutsches Forschungs-Netzwerk (German Science Network). *Das Wissenschafftsnetz X-WIN*. http://www.dfn.de/xwin, last checked: 30.03.2015. 2015.

[Ngu+12] Thuy T. T. Nguyen, Grenville Armitage, Philip Branch, and Sebastian Zander. "Timely and Continuous Machine-learning-based Classification for Interactive IP Traffic". In: *IEEE/ACM Trans. Netw.* 20.6 (Dec. 2012), pp. 1880–1894. ISSN: 1063-6692. DOI: 10.1109/TNET.2012.2187305. URL: http://dx.doi.org/10.1109/TNET.2012.2187305.

[Nic+98] K. Nichols, S. Blake, F. Baker, and D. Black. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. RFC 2474 (Proposed Standard). Updated by RFCs 3168, 3260. Internet Engineering Task Force, Dec. 1998. URL: http://www.ietf.org/rfc/rfc2474.txt.

[NRG] Network Research Group (NRG). *libpcap - the Packet Capture library*. http://sourceforge.net/projects/libpcap/.

[nt96] ITU-T Study Group 12 - End-to-end transmission performance of networks and terminals. *P.800 : Methods for subjective determination of transmission quality.* public document. `http://www.itu.int/rec/T-REC-P.800-199608-I/en`. 1996.

[Pap+05] K. Papagiannaki, N. Taft, Zhi-Li Zhang, and C. Diot. "Long-term forecasting of Internet backbone traffic". In: *Neural Networks, IEEE Transactions on* 16.5 (Sept. 2005), pp. 1110–1124. ISSN: 1045-9227. DOI: `10.1109/TNN.2005.853437`.

[Pax+98] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. *Framework for IP Performance Metrics.* RFC 2330 (Informational). Internet Engineering Task Force, May 1998. URL: `http://www.ietf.org/rfc/rfc2330.txt`.

[Ped+05] P. Pedreiras, P. Gai, L. Almeida, and G.C. Buttazzo. "FTT-Ethernet: a flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems". In: *Industrial Informatics, IEEE Transactions on* 1.3 (Aug. 2005), pp. 162–172.

[PM15] R. Pantos and W. May. *HTTP Live Streaming, Internet Draft (draft-pantos-http-live-streaming-18), Work in Progress.* 2015. URL: `https://tools.ietf.org/html/draft-pantos-http-live-streaming-18`.

[PNA12] P.NAMS. *Parametric non-intrusive assessment of audiovisual media streaming quality.* In force. Oct. 2012.

[Pro04] Mono Project. *Cross Platform, Open Source .NET Framework.* online resource. `http://www.mono-project.com/`. 2004.

[QUA] QUALCOMM. *StreamBoost, Qualcomm Introduces StreamBoost Technology to Optimize Performance and Capacity of Home Networks.* `https://www.qualcomm.com/news/releases/2013/01/04/qualcomm-introduces-streamboost-technology-optimize-performance-and`, last checked: 30.10.2015.

[Quo+15] Do Le Quoc, V. D'Alessandro, B. Park, L. Romano, and C. Fetzer. "Scalable Network Traffic Classification Using Distributed Support Vector Machines". In: *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*. June 2015, pp. 1008–1012. DOI: 10.1109/CLOUD.2015.138.

[RA14] Anas Al-Roubaiey and M. AL-Rhman Alkhiaty. *QoS-Aware Middleware for Ubiquitous Environment: A Review and Proposed Solution*. Journal of Computational Engineering. 2014.

[RK79] Cornelis van Rijsbergen and Joost Keith. *Information Retrieval*. 2nd ed. Butterworth, Boston, UK, 1979. ISBN: 0408709294.

[RM12] E. Rescorla and N. Modadugu. *Datagram Transport Layer Security Version 1.2*. RFC 6347 (Proposed Standard). Internet Engineering Task Force, Jan. 2012. URL: http://www.ietf.org/rfc/rfc6347.txt.

[Ros+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. *SIP: Session Initiation Protocol*. RFC 3261 (Proposed Standard). Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141, 6665, 6878. Internet Engineering Task Force, June 2002. URL: http://www.ietf.org/rfc/rfc3261.txt.

[Ros90] M.T. Rose. *Management Information Base for network management of TCP/IP-based internets: MIB-II*. RFC 1158 (Proposed Standard). Obsoleted by RFC 1213. Internet Engineering Task Force, May 1990. URL: http://www.ietf.org/rfc/rfc1158.txt.

[Rou+04] Matthew Roughan, Subhabrata Sen, Oliver Spatscheck, and Nick Duffield. "Class-of-service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification". In:

Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. IMC '04. Taormina, Sicily, Italy: ACM, 2004, pp. 135–148. ISBN: 1-58113-821-0. DOI: 10.1145/1028788.1028805. URL: http://doi.acm.org/10.1145/1028788.1028805.

[RS99] E. Rescorla and A. Schiffman. *The Secure HyperText Transfer Protocol*. RFC 2660 (Experimental). Internet Engineering Task Force, Aug. 1999. URL: http://www.ietf.org/rfc/rfc2660.txt.

[Sch+03] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550 (INTERNET STANDARD). Updated by RFCs 5506, 5761, 6051, 6222, 7022. Internet Engineering Task Force, July 2003. URL: http://www.ietf.org/rfc/rfc3550.txt.

[Sch+14] Mirko Schiavone, Peter Romirer-Maierhofer, Fabio Ricciato, and Andrea Baiocchi. "Towards Bottleneck Identification in Cellular Networks via Passive TCP Monitoring". English. In: *Ad-hoc, Mobile, and Wireless Networks*. Ed. by Song Guo, Jaime Lloret, Pietro Manzoni, and Stefan Ruehrup. Vol. 8487. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 72–85. ISBN: 978-3-319-07424-5. DOI: 10.1007/978-3-319-07425-2_6.

[Sch13] Douglas C. Schmidt. *The ADAPTIVE Communication Environment*. online resource. http://www.cse.wustl.edu/~schmidt/ACE.html. 2013.

[SD02] G. Sivaradje and P. Dananjayan. "Dynamic resource allocation algorithm for next generation wireless multimedia. services". In: *Communication Systems, 2002. ICCS 2002. The 8th International Conference on*. Vol. 2. Nov. 2002, 752–754 vol.2.

[SH10] H. Schulzrinne and R. Hancock. *GIST: General Internet Signalling Transport*. RFC 5971 (Experimental). Internet Engineering Task Force, Oct. 2010. URL: http://www.ietf.org/rfc/rfc5971.txt.

[Sha48] Claude Elwood Shannon. "A mathematical theory of communication". In: *Bell system technical journal* 27 (1948).

[SSK98] Rachit Siamwalla, Rosen Sharma, and Srinivasan Keshav. "Discovering internet topology". In: *Cornell University, Ithaca NY* (1998). http://www.cs.cornell.edu/skeshav/papers/discovery.pdf.

[Sur+10] Vincenzo Suraci, Guido Oddi, Nico Mattiacci, and Andrea Angelucci. *Admission Control and Drop Strategies in a UPnP-QoS Controlled Home Network*. IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications. 2010.

[SWS05] Yantao Sun, Zhimei Wu, and Zhiqiang Shi. "The physical topology discovery for switched Ethernet based on connections reasoning technique". In: *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on* 1 (Oct. 2005), pp. 44–47. DOI: 10.1109/ISCIT.2005.1566795.

[Sys05] Cisco Systems. *Managing Peer-To-Peer Traffic With Cisco Service Control Technology*. White Paper. 2005.

[Tcp15] Tcpdump. *Tcpdump, a powerful command-line packet analyzer*. http://www.tcpdump.org/, last checked: 30.03.2015. 2015.

[Tec07] IneoQuest Technologiest. *Why IPTV/IP Video Transport is Different from Data and Voice*. IneoQuest Application Note. http://www.dailyiptv.com/whitepaper/why-iptv-is-different-from-data-and-voice/. 2007.

[TIA06] TIA. *TIA-1057 Standard for Link Layer Discovery Protocol for Media Endpoint Devices*. TELECOMMUNICATIONS INDUSTRY ASSOCIATION. May 2006.

[TKC05] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. "Core Vector Machines: Fast SVM Training on Very Large Data Sets". In: *J. Mach. Learn. Res.* 6 (Dec. 2005), pp. 363–392. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1046920.1058114.

[Too15] IP Route Tools. *tc - show / manipulate traffic control settings*. http://linux.die.net/man/8/tc,last checked: 30. 10. 2015. 2015.

[Var14] Various. *Calculus/Taylor series*. Wikibooks.org. https://en.wikibooks.org/wiki/Calculus/Taylor_series. 2014.

[WC06] J. Welch and J. Clark. *A Proposed Media Delivery Index (MDI)*. RFC 4445 (Informational). Internet Engineering Task Force, Apr. 2006. URL: http://www.ietf.org/rfc/rfc4445.txt.

[Wes+03] L. Westberg, M. Jacobsson, M. de Kogel, S. Oosthoek, D. Partain, V. Rexhepi, and P. Wallentin. *Resource Management in Diffserv On DemAnd (RODA) PHR, Internet Draft (draft-westberg-rmd-od-phr-04), Work in Progress*. 2003. URL: http://tools.ietf.org/html/draft-westberg-rmd-od-phr-04.

[Wik13] Wikipedia. *Depth-first search*. http://en.wikipedia.org/wiki/Depth-first_search, last checked: 30. 10. 2015. 2013.

[Wik15] The Free Encyclopedia Wikipedia. *Linksys WRT54G series*. http://en.wikipedia.org/wiki/Linksys_WRT54G_series, last checked: 30. 03. 2015. 2015.

[Wik16] The Free Encyclopedia Wikipedia. *Net neutrality*. https://en.wikipedia.org/wiki/Net_neutrality, last checked: 30. 03. 2016. 2016.

[Wro97] J. Wroclawski. *The Use of RSVP with IETF Integrated Services*. RFC 2210 (Proposed Standard). Internet Engineering Task Force, Sept. 1997. URL: `http://www.ietf.org/rfc/rfc2210.txt`.

[WZ12] Nigel Williams and Sebastian Zander. *Real Time Traffic Classification and Prioritisation on a Home Router using DIFFUSE*. Tech. rep. 120412A. Melbourne, Australia: Centre for Advanced Internet Architectures, Swinburne University of Technology, Dec. 2012. URL: `http://caia.swin.edu.au/reports/120412A/CAIA-TR-120412A.pdf`.

[Yan+03] Mei Yang, Yan Huang, J. Kim, Meejeong Lee, T. Suda, and M. Daisuke. "An end-to-end QoS framework with on-demand bandwidth reconfiguration". In: *Computer Communications, 2003. CCW 2003. Proceedings. 2003 IEEE 18th Annual Workshop on*. 2003, pp. 66–74. DOI: `10.1109/CCW.2003.1240792`.

[Yia+11] Yiannis Yiakoumis, Kok-Kiong Yap, Sachin Katti, Guru Parulkar, and Nick McKeown. "Slicing home networks". In: *Proceedings of the 2nd ACM SIGCOMM workshop on Home networks*. ACM. 2011, pp. 1–6.

Publications

[Koe+15] Christopher Koehnen, Christian Ueberall, Muttukrishnan Rajarajan, Rudolf Jaeger, and Veselin Rakocevic. "Autonomous QoS Management and Policing in Unmanaged Local Area Networks". In: *Journal of Computer Networks and Communications* 2015 (2015).

[Ada+11] Florian Adamsky, Christopher Kohnen, Christian Uberall, Veselin Rakocevic, Muttukrishnan Rajarajan, and Rudolf Jager. "A Novel Concept For Hybrid Quality Improvements in Consumer Networks". In: *Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on*. IEEE. 2011, pp. 39–43.

[Koe+10a] Christopher Koehnen, Christian Uberall, Veselin Rakocevic, Muttukrishnan Rajarajan, and Rudolf Jaeger. "QoSiLAN - A Heterogeneous Approach to Quality of Service in Local Area Networks". In: *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*. IEEE. 2010, pp. 109–112.

[Koe+10b] Christopher Koehnen, Christian Ueberall, Florian Adamsky, Veselin Rakocevic, Muttukrishnan Rajarajan, and Rudolf Jaeger. "Enhancements to Statistical Protocol IDentification (SPID) for Self-Organised QoS in LANs". In: *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*. IEEE. 2010, pp. 1–6.

Further Publications

[Dev+13] Oskar van Deventer, Jost de Wit, Marc Guelbahar, Bin Cheng, Felix Marmol-Gomez, Christian Koebel, Christopher Koehnen, Georg Rozinaj, and Bjoern Stockleben. "Towards next generation Hybrid broadcast broadband, results from FP7 and HbbTV 2.0". In: *IBC2013 Conference* (2013).

[Koe+13a] Christian Koebel, Christopher Koehnen, Nils Hellhund, Bastian Zeller, Ray van Brandenburg, Arjen Veenhuizen, Janina Renz, Michael Probst, and Bjoern Stockleben. "D4.3.1: EVALUATION: Intermediate Middle-ware Software Components for Content Synchronization". In: *Deliverable HBB-NEXT, EU-Project FP7:Call11*. `http://cordis.europa.eu/project/rcn/100252_en.html`. Mar. 2013.

[Koe+13b] Christopher Koehnen, Nils Hellhund, Janina Renz, and Jennifer Mueller. "Inter-Device and Inter-Media Synchronization in HBB-NEXT". In: *Media Synchronization Workshop (MediaSync) at NEM Summit 2013*. Nantes, France, Oct. 2013.

[Pro+13] Michael Probst, Stefan Heller, Manuel Melic, Sachin Agarwal, Felix Gomez Marmol, Gines Dolera, Bin Cheng, Ray van Brandenburg, Joost de Wit, Gregor Rozinaj, Pavol Podhradsky, Ivan Kotuliak, Janina Renz, Mark Guelbahar, Christian Koebel, Christopher Koehnen, and Bastian Zeller. "D6.1.2: Intermediate HBB-NEXT System Architecture". In: *Deliverable HBB-NEXT, EU-Project FP7:Call11*. `http://cordis.europa.eu/project/rcn/100252_en.html`. Mar. 2013.

[Ueb+13] Christian Ueberall, Christopher Koehnen, Veselin Rakocevic, Rudolf Jaeger, Erich Hoy, and Muttukrishnan Rajarajan. "Recommendations in a heterogeneous service environment". In: *Multimedia tools and applications* 62.3 (2013), pp. 785–820.

[Koe+12] Christopher Koehnen, Nils Hellhund, Christian Koebel, Ray van Brandenburg, Arjen Veenhuizen, Janina Renz, Michael Probst, Jennifer Mueller, and Sachin Agarwal. "D4.2: DESIGN AND PROTOCOL: Middleware Components Content Synchronisation/Cloud Service Offloading"". In: *Deliverable HBB-NEXT, EU-Project FP7:Call11*. `http://cordis.europa.eu/project/rcn/100252_en.html`. Sept. 2012.

[KKH12] Christopher Koehnen, Christian Koebel, and Nils Hellhund. "A DVB/IP streaming testbed for hybrid digital media content synchronization". In: *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on*. Sept. 2012, pp. 136–140. DOI: `10.1109/ICCE-Berlin.2012.6336493`.

[Pro+12] Michael Probst, Bjoern Stockleben, Jennifer Mueller, Bettina Heidkamp-Tchegloff, Janina Renz, Felix Gomez Marmol, Gines Dolera, Ivan Kotuliak, Tomas Kovacik, Sebastian Schumann, Christopher Koehnen, Marcel Raner, Ray van Brandenburg, and Stefan Heller. "D6.3.1: Report on test applications for enablers of WP3/WP4/WP5". In: *Deliverable HBB-NEXT, EU-Project FP7:Call11*. `http://cordis.europa.eu/project/rcn/100252_en.html`. Sept. 2012.

[Sto+12] Bjoern Stockleben, Jennifer Mueller, Bettina Heidkamp-Tchegloff, Janina Renz, Felix Gomez Marmol, Gines Dolera, Ivan Kotuliak, Tomas Kovacik, Sebastian Schumann, Christopher Koehnen, Marcel Raner, Ray van Brandenburg, and Stefan Heller. "D6.3.1 Report on Test Applications for Enablers of WP3/WP4/WP5". In: *Deliverable HBB-NEXT, EU-Project FP7:Call11*. `http://cordis.europa.eu/project/rcn/100252_en.html`. Sept. 2012.

[Ueb+09] Christian Ueberall, Rajarajan Muttukrishnan, Veselin Rakocevic, Rudolf Jaeger, and Christopher Koehnen. "Recommendation index for DVB content using service information". In:

Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on. IEEE. 2009, pp. 1178–1181.

[Jae+08] Rudolf Jaeger, Christopher Koehnen, Christian Ueberall, Marcel Becker, and Fahd Bellot. "System Architecture for advanced iTV Services in an IPTV Environment". In: (2008).