Integration of Unidirectional Technologies into a *Wireless Back-haul Architecture*

A thesis submitted for the degree of Doctor of Philosophy by
Mathias Kretschmer
at
School of Information Systems, Computing and Mathematics
Brunel University

May 2012

Abstract

Back-haul infrastructures of today's wireless operators must support the triple-play services demanded by the market or regulatory bodies. To cope with increasing capacity demand, the EU FP7 project *CARMEN* has developed a cost-effective heterogeneous multi-radio wireless back-haul architecture, which may also leverage the native multicast capabilities of broadcast technologies such as DVB-T to off-load high-bandwidth broadcast content delivery. However, the integration of such unidirectional technologies into a packet-switched architecture requires careful considerations.

The contribution of this thesis is the investigation, design and evaluation of protocols and mechanisms facilitating the integration of such unidirectional technologies into the wireless back-haul architecture so that they can be configured and utilized by the spectrum and capacity optimization modules. This integration mainly concerns the control plane and, in particular, the aspects related to resource and capability descriptions, neighborhood, link and Multi Protocol Label Switching (MPLS) Label-Switched Path (LSP) monitoring, unicast and multicast LSP signalling as well as topology forming and maintenance.

During the course of this study we have analyzed the problem space, proposed solutions to the resulting research questions and evaluated our approach. Our results show that the now Unidirectional Technology (UDT)-aware architecture can readily consider Unidirectional Technologies (UDTs) to distribute, for example, broadcast content.

Acknowledgements

Although the efforts that led to this PhD thesis were driven and for a large part performed by the author, such a multi-year endeavor would hardly be possible without support from friends, colleagues or partners. I would like to thank Karl Jonas for eventually getting me started and supporting my efforts along my regular duties. Jens Moedeker deserves a big 'Thanks' for all the valuable discussions during the course of this thesis and especially for reviewing this document and the numerous hints and tips to improve it. I also like to thank Christian Niephaus for his input, the fruitful discussions as well as the contributions to many of our publications. Also 'Thank You' to Thorsten Horstmann and Philipp Batroff for their input, critical questions and for actually getting our demonstrator off the ground.

A big thanks to my friends and family whom I got to meet less frequently during this process. And, a special *Thank You* goes to Chia-Yen for her understanding, support and valuable input.

Last, but not least, I would like to thank my advisor, Dr. Gheorghita Ghinea, for his support, tips and feedback as well as the way he let me freely manage my *project* matching my time and work constraints.

List of Publications

The research presented in this thesis has resulted in the below listed publications.

Journals

- M. Kretschmer, J. Moedeker, and G. Ghinea, “Path signalling in a wireless back-haul network integrating unidirectional broadcast technologies,” *IEEE Transactions on Broadcasting*, Under Review
- M. Kretschmer, P. Batroff, and G. Ghinea, “Topology forming and optimization framework for heterogeneous wireless back-haul networks supporting unidirectional technologies,” *Journal of Computer Communications (COMCOM, Elsevier)*, Under Review

Book Chapters

- M. Kretschmer, C. Niephaus, and G. Ghinea, *Wireless Multi-Access Environments and Quality of Service Provisioning: Solutions and Application*, ch. Heterogeneous Meshed Wireless Back-haul Network Integrating Unidirectional Technologies, pp. 139–160. Information Science Reference, USA/UK: IGI Global, 2012

Peer-reviewed Conferences and Workshops

- M. Kretschmer and G. Ghinea, “Seamless integration of unidirectional broadcast links into qos-constrained broadband wireless mesh access networks,” in *Proc. The 4th International Conference for Internet Technology and Secured Transactions*, 2009
- A. Banchs, N. Bayer, D. Chieng, A. de la Oliva, B. Gloss, M. Kretschmer, S. Murphy, M. Natkaniec, and F. Zdarsky, “Carmen: Delivering carrier grade services over wireless mesh networks,” in *Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2008*, pp. 1–6, Sept. 15–18, 2008
- M. Kretschmer, P. Batroff, C. Niephaus, and G. Ghinea, “Topology discovery and maintenance for heterogeneous wireless Back-Haul networks supporting unidirectional technologies,” in *17th Asia-Pacific Conference on Communications (APCC 2011)*, (Kota Kinabalu, Sabah, Malaysia), Oct. 2011
- M. Kretschmer and G. Ghinea, “An IEEE 802.21-based approach for seamless wireless mobile integration using QoS-aware paths supporting unidirectional links,” in *IEEE Globecom 2010 Workshop on Seamless Wireless Mobility (SWM 2010)*, (Miami, Florida, USA), 12 2010

- D. Henkel, S. Englaender, M. Kretschmer, and C. Niephaus, "Connecting the unconnected - economical constraints and technical requirements towards a Back-Haul network for rural areas," in *IEEE Globecom 2011 Workshop on Rural Communications-Technologies, Applications, Strategies and Policies (RuralComm 2011) (GC'11 Workshop - RuralComm)*, (Houston, Texas, USA), Dec. 2011
- T. Horstmann, M. Kretschmer, C. Niephaus, J. Mödeker, and S. Sauer, "Development framework for prototyping heterogeneous Multi-Radio wireless networks," in *ICCCN 2011 Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN 2011)*, (Maui, Hawaii, USA), July 2011
- M. Kretschmer, C. Niephaus, and G. Ghinea, *Towards QoS Provisioning in a Heterogeneous Carrier-Grade Wireless Mesh Access Networks Using Unidirectional Overlay Cells*, vol. 22 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.* Springer Berlin Heidelberg, 2009
- M. Kretschmer, C. Niephaus, and G. Ghinea, "QoS-aware flow monitoring and event creation in heterogeneous MPLS-based wireless mesh networks supporting unidirectional links," in *9th IEEE Malaysia International Conference on Communications 2009*, (Kuala Lumpur, Malaysia), 2009
- M. Kretschmer, C. Niephaus, D. Henkel, and G. Ghinea, "QoS-aware wireless backhaul network for rural areas with support for broadcast services in practice," in *Fifth IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking (IEEE MeshTech 2011)*, (Valencia, Spain), Oct. 2011
- M. Kretschmer, S. Robitzsch, C. Niephaus, K. Jonas, and G. Ghinea, "Wireless mesh network coverage with QoS differentiation for rural areas," in *First International Workshop on Wireless Broadband Access for Communities and Rural Developing Regions*, (Karlstad, Sweden), 12 2008
- M. Kretschmer, C. Niephaus, and G. Ghinea, "A wireless back-haul architecture supporting dynamic broadcast and white space coexistence," in *ICCCN 2012 Workshops: 6th International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN) (WiMAN 2012)*, (Munich, Germany), July 2012
- M. Kretschmer, T. Horstmann, P. Batroff, M. Rademacher, and G. Ghinea, "Link calibration and property estimation in Self-Managed wireless Back-Haul networks," in *18th Asia-Pacific Conference on Communications (APCC 2012)*, (Ramada Plaza Jeju Hotel, Jeju Island, Korea), Oct. 2012
- M. Kretschmer, C. Niephaus, and G. Ghinea, "Long-range wireless mesh networks for unconnected regions," in *E-Infrastructures and E-Services on Developing Countries: First International ICST Conference, AFRICOMM December 2009, Maputo, Mozambique*, vol. 38, LNICST, Springer, June 2010

- M. Kretschmer, P. Hasse, C. Niephaus, T. Horstmann, and K. Jonas, “Connecting mobile phones via carrier-grade meshed wireless back-haul networks,” in *E-Infrastructures and E-Services on Developing Countries: Second International ICST Conference, AFRICOMM 2010, Cape Town, South Africa* (Springer, ed.), October 2011

- J. Lessmann, A. De La Oliva, C. Sengul, A. Garcia, M. Kretschmer, S. Murphy, and P. Patras, “On the scalability of carrier-grade mesh network architectures,” in *Future Network Mobile Summit (FutureNetw), 2011*, pp. 1 –8, june 2011

- S. Robitzsch, C. Niephaus, J. Fitzpatrick, and M. Kretschmer, “Measurements and evaluations for an ieee 802.11a based carrier-grade multi-radio wireless mesh network deployment,” in *Wireless and Mobile Communications, 2009. ICWMC ’09. Fifth International Conference on*, pp. 272–278, Aug. 2009

- M. Kretschmer, C. Niephaus, T. Horstmann, and K. Jonas, “Providing mobile phone access in rural areas via heterogeneous meshed wireless back-haul networks,” in *Communications Workshops (ICC), 2011 IEEE International Conference on*, pp. 1 –6, june 2011

- C. Niephaus and M. Kretschmer, “Towards an energy management framework for carrier-grade wireless Back-Haul networks,” in *ICCCN 2012 Workshops: 6th International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN) (WiMAN 2012)*, (Munich, Germany), July 2012

Contents

List of Figures

ACM Advanced Coding and Modulation

AI Abstract Interface

AODV Ad hoc On-Demand Distance Vector Routing Protocol

ARP Address Resolution Protocol

ARQ Automatic Repeat reQuest

AS Autonomous System

ATM Asynchronous Transfer Mode

ATSC Advanced Television Systems Committee

BRA Bidrectional Routing Abstraction

DSR Dynamic Source Routing Protocol

3GPP 3rd Generation Partnership Project

BGP Border Gateway Protocol

CAPEX Capital Expenditures

CARMEN CARrier grade wireless MEsh Network

CDMA Code Division Multiple Access

CMF Capacity Management Function

CIDR Classless Inter-Domain Routing

CR-LDP Constraint-based Routing Label Distribution Protocol

CSMA/CD Carrier Sense Multiple Access/Collision Detection

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance

DAD Duplicate Address Detection

DFS Dynamic Frequency Selection

DHCP Dynamic Host Configuration Protocol

DoS Denial of Service

DPCF Dynamic Protocol Composition Framework

DR Designated Router

DSL Digital Subscriber Line

DTCP Dynamic Tunnel Configuration Protocol

DVB Digital Video Broadcast

DVB-S Digital Video Broadcast - Satellite

DVB-S2 Digital Video Broadcast - Satellite - Second Generation

DVB-T Digital Video Broadcast - Terrestrial

DVB-T2 Digital Video Broadcast - Second Generation Terrestrial

DVB-RCS Digital Video Broadcast - Return Channel Satellite

EGP Exterior Gateway Protocol

EIGRP Enhanced Interior Gateway Routing Protocol

ERO Explicit Route Object

FDMA Frequency Division Multiple Access

FRR Fast Reroute

GAN Generic Access Network

GMPLS Generalized Multiprotocol Label Switching

GPS Global Positioning System

GRE Generic Routing Encapsulation

GSE Generic Stream Encapsulation

GSM Global System for Mobile Communications

HWMP Hybrid Wireless Mesh Protocol

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IGP Interior Gateway Protocol

IGRP Interior Gateway Routing Protocol

IMF Interface Management Function

IP Internet Protocol

IPFIX IP Flow Information Export

IPv4 Internet Protocol

IPv6 Internet Protocol, Version 6

IS-IS Intermediate system to intermediate system

ISP Internet Service Provider

LDP Label Distribution Protocol

LER Label Edge Router

LLTM Link Layer Tunneling Mechanism

LSP Label-Switched Path

LSR Label-Switched Router

LTE Long Term Evolution

MAC Media Access Control

MANET Mobile Adhoc Network

MBMS Multimedia Broadcast Multicast Service

MCS Modulation and Coding Scheme

MDR MANET Designated Router

MPEG Moving Picture Experts Group

MICS Media Independent Command Service

MIES Media Independent Event Service

MIH Media Independent Handover

MIIS Media Independend Information Service

MIHF Media Independent Handover Function

MIHF++ Media Independent Handover Function++

MIIS Media Independent Information Service

MIMS Media Independent Messaging Service

MPE Multi Protocol Encapsulation

MPLS Multi Protocol Label Switching

MPR Multipoint Relay

MR-WMN Multi-Radio Wireless Mesh Network

MSC Mobile-services Switching Centre

MTU Maximum Transmit Unit

NBMA Non-broadcast Multiple Access

NLOS None Line of Sight

NSIS Next Steps in Signalling

OFDMA Orthogonal Frequency Division Multiple Access

OFDM Orthogonal Frequency Division Multiplex

OLSR Optimised Link State Routing

OPEX Operational Expenditure

OSI Open Systems Interconnection

OSPF Open Shortest Path First

OSPF-TE Open Shortest Path First - Traffic Engineering

PCC Path Computation Client

PCE Path Computation Element

PCEP Path Computation Element Protocol

PDU Protocol Data Unit

PDV Packet Delay Variation

PID Packet ID

PLR Point of Local Repair

PMF Pipe Management Function

PMIP Proxy Mobile IP

PMP Pipe Management Protocol

PMT Program Map Table

QLANE QoS-aware LAN Emulation

QoS Quality of Service

RAN Radio Access Network

RCS Return Channel System

RFC Request for Comments

RIP Routing Information Protocol

RIPng Routing Information Protocol next generation

RPC Remote Procedure Call

RREQ Route Request

RREP Route Reply

RERR Route Error

RSVP Resource ReSerVation Protocol

RSVP-TE Resource ReSerVation Protocol - Traffic Engineering

RTT Round Trip Time

SAA Stateless Address Autoconfiguration

SAP Service Access Point

SBM Subnetwork Bandwidth Manager

SC-FDMA Single Carrier Frequency Division Multiple Access

ScF Self-configuration Function

SDMA Space-division multiple access

SDT Service Description Table

SENF Simple and Extensible Network Framework

SF Statistics Function

SFN Single Frequency Network

SLA Service Level Agreement

SONET Synchronous Optical Networking

STP Spanning Tree Protocol

TC Topology Control

TCF Terminal Control Function

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TE Traffic Engineering

TIM Technology Independent Monitoring

TLV Type-Length-Value

TMF Topology Management Function

TTL Time to live

UDL Unidirectional Link

UDLR Unidirectional Link Routing

UDP User Datagram Protocol

UDT Unidirectional Technology

UDTs Unidirectional Technologies

UHF Ultra High Frequency

UMA Unlicensed Mobile Access

UMTS Universal Mobile Telecommunications System

ULE Unidirectional Light Encapsulation

U-NII Unlicensed National Information Infrastructure

USO Universal Service Obligation

UT User Terminal

VHF Very High Frequency

VLAN Virtual Local Area Network

VM Virtual Machine

VoIP Voice-over-IP

VPN Virtual Private Network

WAP WiBACK Access Point

WC WiBACK Coordinator

WGW WiBACK Gateway

WiBACK Wireless Back-Haul

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Network

WMN Wireless Mesh Network

WN WiBACK Node

WSN Wireless Sensor Network

Chapter 1

Introduction

Wireless operators, in developed or emerging regions, must support the triple-play service offerings demanded by the market or by regulatory bodies through so-called Universal Service Obligations (USOs). Such USOs often also require the coverage of a large percentage of the population, which, especially in emerging regions, lives in vast rural areas outside the larger cities. Since individual operators might face different constraints such as available spectrum licenses or technologies, the EU FP7 CARrier grade wireless MEsh Network (CARMEN) project[1] has developed a carrier-grade heterogeneous multi-radio wireless back-haul architecture which may be deployed to extend, complement or even replace traditional operator equipment. In its initial design, this architecture focuses on bidirectional technologies and therefore cannot natively support unidirectional technologies, such as Digital Video Broadcast - Terrestrial (DVB-T) or Advanced Television Systems Committee (ATSC), which could be exploited to off-load the distribution of triple-play broadcast payload. To support off-loading of broadcast content, such as *live* TV or radio programming, *it is the goal of this study to integrate support for unidirectional technologies into all affected aspects of this heterogeneous wireless back-haul architecture.* This extended architecture has been developed at Fraunhofer FOKUS[2] and is referred to, as Wireless Back-Haul (WiBACK)[3] throughout this study.

Based on such architectural support, smart content distribution and pre-recording scheduling architectures, such as the recently proposed *Dynamic Broadcast* [23, 24] system, or demand-based broadcast scheduling systems, such as [25], could be implemented at the service level. In [23], the authors argue that by cleverly decomposing traditional TV programming into live and non-live segments and optimizing the distribution schedule of live and non-live content, parts of the precious Very High Frequency (VHF) or Ultra High Frequency (UHF) spectra could be temporarily, or even permanently, freed up. In such a scenario, the WiBACK architecture, as proposed in this study, could coordinate the spectrum sharing and trading between broadcast and typical wireless operator spectra and orchestrate a coordinated use of so-called *white spaces.*

---

[1] http://www.ict-carmen.eu

[2] http://www.fokus.fraunhofer.de

[3] http://www.wiback.org

Moreover, to support the deployment of cost-effective wireless back-haul networks in rural areas, especially in emerging regions [8], IEEE 802.11 hardware is often considered for long distance back-haul links, since it offers powerful 40MHz-wide Orthogonal Frequency Division Multiplex (OFDM) PHY technology in semi-professional quality at consumer prices. In order to increase the Media Access Control (MAC) layer efficiency for such point-to-point or even broadcast deployments, various MAC protocol optimizations and alternatives have been proposed, such as Time Division Multiple Access (TDMA)-like schemes [26, 27] or even unidirectional acknowledgement-free operation. On a conceptual level, the latter would essentially decompose a bidirectional technology into a pair of unidirectional technologies and would therefore require similar Unidirectional Technology (UDT) integration considerations as, for example, DVB-T.

Thus, the goal of this study is: *To identify, describe and integrate UDTs into the WiBACK architecture, so that the spectrum and capacity management functions can transparently configure and utilize them when beneficial under the current physical radio conditions, payload characteristics, receiver distribution or content type.* The minimum requirement towards a node's connectivity is considered to be the availability of at least one transmit-capable and one receive-capable interface. Reflecting the capabilities of today's wireless technologies, a typical node can be assumed to be equipped with at least one bidirectional interface, while additional unidirectional interfaces may be available. The proposed approach for the integration of UDTs should therefore exploit the advantages of both types of interfaces, considering the trade-off between increased complexity depending on the level of UDT support and the potential gains in terms of network-wide capacity and stability.

In the following sections, we introduce wireless back-haul networks and provide a summary of their requirements, focusing on the support of heterogeneous, and in particular unidirectional, technologies. We then describe the characteristics of UDTs and introduce the advantages and challenges of integrating UDTs into the WiBACK architecture.

The author was a member of the EU FP7 CARMEN consortium which studied and specified a carrier-grade wireless mesh network. His responsibility was the integration of unidirectional broadcast cells into the wireless back-haul architecture developed by the CARMEN consortium. The WiBACK architecture developed by Fraunhofer FOKUS is based on the consolidated outcomes of the CARMEN project and is referred to throughout this study as the reference platform for our UDT integration efforts.

1.1 Wireless Back-haul Networks

A wireless operator back-haul network as depicted in Figure 1.1 needs to support combined voice and broadband data services as well as an increasing demand for high-bandwidth multimedia content which can be provided via i.e the Multimedia Broadcast Multicast Service (MBMS). This service bundle is also referred to as *triple-play.* Such a network typically consists of a hybrid wired and wireless infrastructure, which is built upon reliable hardware and the deployment has been carefully planned. Where wireless links are used for back-hauling, they are typically operated in the licensed spectrum using STM-1 or Gigabit-Ethernet point-to-point wireless technology which is often organized as a redundant ring to provide a backup connectivity, see, for example [28].

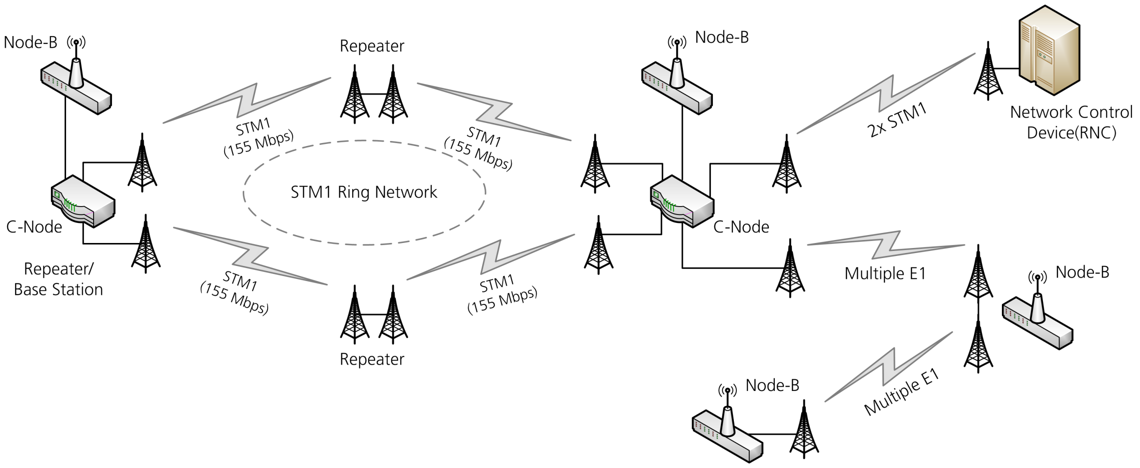

Figure 1.1: Typical wireless operator back-haul network

1.1.1 Use Cases and Requirements

A typical operator back-haul network is planned, built and configured for its specific application at a fixed place for an estimated maximum user bandwidth demand. The majority of the traffic flows from the backbone to the user terminals while on the return path a lower bandwidth demand is assumed. Communication between user terminals within the same back-haul network is possible but only accounts for a rather small percentage of the overall traffic. Such a rather static network configuration allows a reliable operation within the planned operational parameters. However, operators are looking into alternatives in order to temporarily extend their coverage during special events with high demand for communication, to serve as an alternative for destroyed infrastructure after a natural catastrophe or to provide more cost-effective coverage for rural or developing areas where traditional infrastructures are not economically feasible [29, 8].

The carrier-grade wireless back-haul architecture developed by the CARMEN consortium addresses the aforementioned scenarios and aims at providing a comparable service quality as a typical operator network at lower Capital Expenditures (CAPEX) and Operational Expenditure (OPEX) [29]. In order to achieve this goal, this architecture can integrate complementary heterogeneous technologies optimally chosen for a specific deployment scenario. After the equipment has been set up its self-management components [30] aim at forming and maintaining an optimized meshed wireless back-haul network among the participating nodes. While in operation, the network monitors its operational parameters and attempts to adapt its configuration to adjust to environmental or usage pattern changes. To better support triple-play services, such as high bandwidth live multimedia content, the CARMEN-based UDT-aware WiBACK architecture, as proposed in this study, supports the utilization of unidirectional broadcast technologies where available. Since the WiBACK architecture provides support for heterogeneous technologies, it also offers the possibility for a smooth gradual transition towards future emerging technologies, such as IEEE 802.22 [31].

A wireless back-haul network is assumed to be run by an operator, commercially or community driven, with the goal of providing a solution that compares to a traditional wireless back-haul network in terms of Quality of Service (QoS)-guarantees and overall reliability and service availability. Therefore, a number of assumption which differentiate such a wireless back-haul network from typical Wireless Mesh Networks (WMNs) have been made by the CARMEN project. A WiBACK network as considered in this thesis is made up of fixed, rugged multi-radio nodes with the assumption that the vast majority is operated 24/7. Directional antennas are used to help reduce interferences or external noise and to increase the range of back-haul links. The self-management components aim at operating links among nodes in an orthogonal point-to-point mode whenever possible. Therefore, the main reasons for link state changes are varying link condition due to (rain-)fading, competing external transmitters or interferences with other noise sources. This might lead to variations of link capacity, introduce jitter and increase the packet loss rate, while traffic load changes on back-haul links occur due to varying user demand. Furthermore, in certain frequency bands, regulatory restrictions might require the detection of coexisting transmitters and demand that such channels are to be avoided. In order to efficiently utilize the scarce wireless resources, the WiBACK architecture aims at choosing the most efficient technology for a given content type and user distribution. Nodes need to be protected against general Denial of Service (DoS) attacks or channel jamming but not against WMN phenomenas such as intentionally misbehaving nodes since basic wireless security will be assumed to restrict the participation to authorized WiBACK Nodes (WNs).

1.2 Heterogeneous Technologies

Choosing the most suitable technology for given link or spectrum characteristics as well as payload requirements can increase the efficiency and eventually the overall network capacity. For example, for broadcast delivery only a single sender is active on a channel. Hence, complicated MAC protocols introducing additional overhead can be avoided. Similarly, in order to avoid the increased MAC overhead on long distance point-to-point back-haul links, such connectivity may be implemented via individual unidirectional links operating on orthogonal channels and polarization planes. To organize such a heterogeneous wireless back-haul network, sophisticated cross-layer management protocols exploiting the distinct capabilities of the various radio technologies are needed in order to most efficiently utilize the available spectrum while ensuring that carrier grade service availability and reliability requirements are met.

The majority of todays wireless technologies provide bidirectional connectivity. Within the context of this study, the minimum requirement for a technology to be integrated is the ability to address and transmit a frame from one node to another or to receive a frame on a specific interface while being able to identify the sender address.

1.2.1 Unidirectional Technologies

Most Internet Engineering Task Force (IETF) protocols, such as Open Shortest Path First (OSPF) [32] or Border Gateway Protocol (BGP) [33], or even subfunctions of e.g. the Internet Protocol, Version 6 (IPv6) [34] itself make the assumption that links between nodes are bidirectional, which means that if node A can reach node B, node B can reach node A via the same link. This assumption is convenient since messages can be exchanged directly without a need to detect more complex return paths. Also link-local heartbeat monitoring or technology-specific link layer optimizations can be easily implemented. Bidirectionality can be assumed with most wired and to some degree even with most wireless network technologies since the link layer implementations of such technologies usually provide bidirectional connectivity.

Looking at the physical layer, however, all common network technologies (Copper/Fiber Ethernet, Wireless WLAN, satellite modems, etc.) consist of a transmitting function and a receiving function, implementing a unidirectional transmit-only resp. a unidirectional receive-only function. It is usually at the link layer (i.e. IEEE 802 technologies) or even a higher layer (i.e. Digital Video Broadcast - Return Channel Satellite (DVB-RCS)), where those separate functions are combined to provide bidirectional connectivity to the upper layers. *Therefore, in the context of this study, the term Unidirectional Technology (UDT) refers to a network technology that, to the upper layers, only provides unidirectional connectivity among one sending interface and potentially numerous receiving interfaces.*

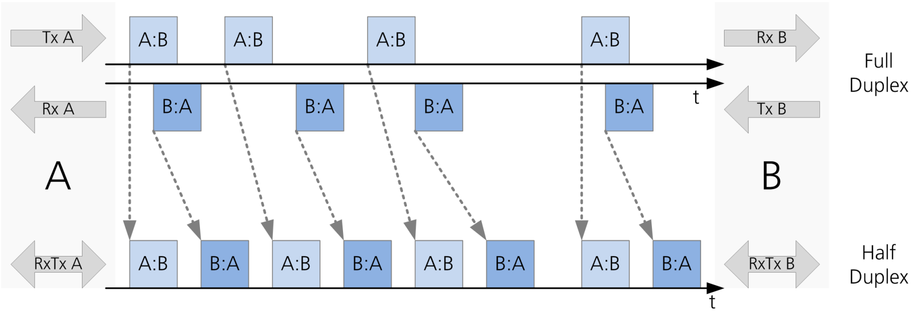

Figure 1.2: Bidirectional link in FDD/SDD vs. TDD mode

Often, transmitters and receivers operate on the same physical channel. This mode is referred to as *half-duplex*, see Figure 1.2. In this case, a MAC protocol is responsible for proper medium access coordination to avoid collisions due to multiple transmitters trying to send a frame at the same time. Typical examples for such MAC protocols are IEEE 802.3 Carrier Sense Multiple Access/Collision Detection (CSMA/CD) [35] or IEEE 802.11 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) [36]. Another option is the use of TDMA schemes where access to a common medium is controlled via, often periodic, assignments of transmit time slots. Figure 1.3 illustrates the MAC overhead of coordinated and uncoordinated access schemes. Shared medium access typically introduces a certain percentage of overhead, but provides an adaptive sharing of the wireless channel resources. Its efficiency depends on the MAC protocol being used. Under such a scheme, multiple nodes can communicate with each other directly, which simplifies the implementation of higher layer protocols since no extra functionality is required.

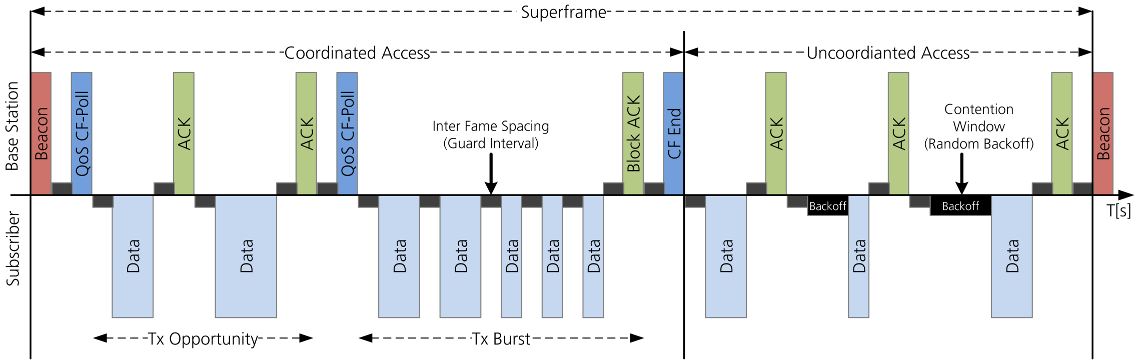

Figure 1.3: Overhead of a typical shared medium MAC protocol

Many modern network technologies solve or avoid the channel access coordination problem by assigning dedicated channels to each transmitter. Hence, common channel access coordination can be avoided and a single transmitter can reach one or multiple receivers. Today's commonly used broadcast technologies, e.g. Digital Video Broadcast (DVB), are implemented using this approach. Combining two pairs of one transmitter and one receiver each, yields commonly used technologies such as IEEE 802.3 100BaseT, fiber links, point-to-point microwave radios or laser links. Channel separation is implemented via Frequency Division Multiple Access (FDMA), Space-division multiple access (SDMA) or a combination thereof. This mode is referred to as *full duplex*, see Figure 1.2. Code Division Multiple Access (CDMA), on the other hand, avoids the typical channels access coordination issues, but is mainly designed for fixed-bandwidth services such as voice calls.

The obvious advantage of full-duplex operation is that no medium access coordination is required which typically yields a higher channel utilization efficiency. Moreover, link latency as well as channel capacity are predictable, as long as no other external interferences exist. The disadvantage is that when a transmitter is idle, its unused channel resources can not readily be used by other transmitters in need of extra resources. Furthermore, additional components such as Layer 2 switches are required to form multipoint-to-multipoint networks. In the wired network domain, the above can usually be tolerated since capacity of cables or fiber is almost abundant, multiple cables can be run in parallel and switching fabric is affordable. Most links forming the Internet are built on this principle of predictable orthogonal *channels* as well as capacity over-provisioning to address QoS issues. For example, major Internet Exchanges Points such as MAE-east or DE-CIX are permanently being upgraded to cope with the ever increasing demand. Despite the technical advantages of wired technologies, setting up wired connectivity is often not economically feasible, especially in rural or temporary deployment scenarios.

In the wireless domain, however, capacity is a scarce resource. Therefore, QoS-aware spectrum and capacity management modules, even in a near-static wireless back-haul network, need to address additional constraints compared to their equivalents in wired networks. The main issue is the rather limited amount of wireless resources compared to the expected traffic volume in triple-play back-haul networks. As described above, QoS-related issues, mainly capacity constraints leading to excessive queuing and eventually packet loss, are often addressed by capacity over-provisioning. In a wired network this paradigm is heavily relied on. And, as long as the packet forwarding engine can cope, adding extra bandwidth is mainly a question of bringing another fiber-link on-line. In the wireless domain, the total bandwidth available to wireless networks in licensed and unlicensed spectrum is only a fraction of the capacity of a single multi-mode fiber. Additionally, a fiber is an almost perfect point-to-point link with sufficient shielding against unintended channel crosstalk, and an almost arbitrary number of links can be set up in parallel. In the wireless domain, the spectrum must be considered as a shared medium where spatial *channel* separation is limited by the physics of free space wave propagation and interferences caused by reflections.

By using time, frequency, as well as spatial multiplexing, the capacity between peering nodes can be increased. Due to physical link characteristics, local interferences with other transmitting devices, different transmit power or receiver sensitivity, link performance might become highly asymmetric. In some cases, a bidirectional wireless link might even loose connectivity in one direction and turn into a unidirectional link [37], effectively breaking the MAC protocol, thus rendering the link behavior unpredictable for QoS-ware capacity considerations.

Advantages of Unidirectional Technologies

Limited wireless spectrum resources should be utilized as efficiently as possible. Hence, it is crucial to choose the proper channel coding and MAC protocol for given payload and link partner requirements in order to increase the spectrum utilization and therefore the overall throughput.

Technologies such as IEEE 802.11, 802.16 or 802.22 as well as DVB-T or its successor DVB-T2 are all capable of using variable and mostly even dynamically adaptable modulation and coding schemes, in order to adapt to varying channel characteristics. They do, however, rely on different framing structures and differ significantly regarding medium access coordination strategies. Hence, while the raw spectral efficiency might be comparable, the effective *MAC layer* efficiency, especially in broadcast mode, may vary significantly. Here, the unidirectional i.e. DVB technologies can provide a higher efficiency compared to bidirectional i.e. IEEE technologies, since they operate without an actual medium access coordination protocol and merely provide packet-based framing via, for example, Multi Protocol Encapsulation (MPE) or the more recent Generic Stream Encapsulation (GSE).

Wireless cells provided by such UDTs consist of only a single logical transmitter and one or multiple receivers. Since medium access does not need to be coordinated, the transmission latency can be very low and even exact transmission timing is possible. In the wireless domain, every transmitting radio can be seen as natively providing unidirectional broadcast connectivity, able to deliver messages to a potentially unlimited number of receivers almost isochronously. The line-of-sight coverage area is only limited by the transmission power and the gain of the antennas. Unidirectional wireless cells are therefore ideal to deliver rich live multi-media content, stock quotes, network updates to any receiver in their coverage area bypassing additional switches or hop-by-hop routing. Thus, assuming a larger number of receivers, broadcast technologies can significantly improve the overall efficiency of triple-play-enabled wireless back-haul networks.

Unidirectional broadcast links offer an attractive cost structure and deployment advantages, since only one higher power transmitter with a high gain antenna needs to be maintained, while the receivers can be based on a rather low-power design and use small antennas. Due to the asymmetric traffic demand in wireless back-haul networks, the coordinated use of UDTs can efficiently increase the overall, and in particular, the downstream capacity. In scenarios, such as the CARMEN 'Emergency Scenario' [38], this can simplify the deployment of satellite gateways, since many gateways can be operated in a rather low-power receive-only mode, while only a small subset of gateways needs to be transmit-capable. Similarly, it can be beneficial for the overall meshed wireless back-haul network connectivity and performance to utilize UDTs where the possible or affordable link budget only allows for a unidirectional physical link configuration, as shown in [39].

Typical traffic engineering approaches plan resources unidirectionally. Hence, the possibility to treat wireless links as unidirectional resources offers greater flexibility to deal with physical layer issues such as interferences, channel jamming, etc. since those are usually unidirectional phenomena often affecting only one end of the communication channel.

Challenges Supporting Unidirectional Technologies

A major drawback of UDTs is that a direct bidirectional communication with a link partner is not possible. This breaks many existing protocols or complicates higher layer protocol design since alternative return paths need to be discovered and maintained. Heartbeat mechanisms are no longer link local mechanisms and might become resource intensive since replies might have to be routed across multi-hop return paths using a considerable amount of the available bandwidth on such paths. Another consequence introduced by the extended return path are inaccuracies or asymmetric behavior of e.g. Round Trip Time (RTT) measurements, which may affect protocols such as the Transmission Control Protocol (TCP) and their congestion control algorithms. Hence, monitoring in networks containing UDTs may require special considerations.

While it is often advantageous to be able to avoid channel access coordination, this approach leads to exclusive wireless channels per transmitter. When this transmitter is idle, precious wireless resources are left unused, since temporary sharing of resources among transmitters can not readily be supported. Access schemes such as CDMA, which are currently used in 3G mobile networks, avoid the access coordination problem in the time domain, but are mainly suitable for constant bit rate applications such as telephony. Supporting high-speed burst-type traffic, e.g. Internet access, poses greater challenges for CDMA networks. Hence, a multiplexed downstream via Orthogonal Frequency Division Multiple Access (OFDMA) and a single transmitter upstream via Single Carrier Frequency Division Multiple Access (SC-FDMA) were chosen for the Long Term Evolution (LTE) architecture. Thus, both directions are operated as unidirectional wireless cells or links.

1.3 Summary

Utilizing UDTs has the potential to increase the capacity of wireless back-haul networks and provide an efficient distribution medium for broadcast content. However, most existing network protocols require bidirectional links to function properly. The amount of modifications required varies depending on the protocol and its layer in the protocol stack. The modifications to be introduced should perform well under bidirectional operation, but readily allow for the utilization of UDTs when present or beneficial for specific traffic or content types (e.g. broadcast) or physical channel conditions. Hence, our aim is *'To describe and integrate UDTs into the WiBACK architecture so that spectrum or capacity optimization algorithms can consider and utilize them'*.

As described in [4] and considering a wireless back-haul architecture based on the CARMEN project outcomes, the aspects that need to be considered to support UDTs are resource and capability descriptions, neighborhood, link and LSP monitoring, unicast and multicast LSP signalling as well as topology forming and maintenance. From those aspects, we have derived our research questions:

- How to identify and describe UDTs?
- How to perform monitoring on UDTs?
- How to signal paths in the presence of UDTs?
- How to include UDTs into Topology Management?

Designing the UDT-aware WiBACK architecture, our contributions address the above listed research questions by:

- Extending the Abstract Interface (AI) to properly describe and handle UDTs.
- Introducing the Technology Independent Monitoring (TIM) component to support passive UDT-aware receiver-side link and LSP monitoring.
- Integrating explicit routing via the *LinkVector* extension into the *Transport Service.*
- Designing the Pipe Management Protocol (PMP) to provide RSVP-TE-style path signalling *around* UDTs.

- Designing the Topology Management Function (TMF) which extends the CARMEN Self-configuration Function (ScF) with support for UDTs

In the following chapter, we evaluate the state-of-the-art regarding the above aspects with a focus on support for UDTs. We then describe the methodology followed throughout this study and then, in the following chapter, present the design of our proposed extensions. In the next chapter, we thoroughly evaluate our work and summarize our findings and contributions. Concluding, we critically review our work, discuss limitations and give an outlook and future directions.

Chapter 2

Literature Review

This chapter provides a thorough review of research activities addressing unidirectional link support in protocols applicable to our WiBACK architecture and evaluates existing approaches against the WiBACK requirements. Since the characteristics of a WiBACK network are more related to those of classical wired back-haul networks instead of WMNs, we will evaluate the literature for traditional routing protocols as well as for WMN-centric approaches.

First, we introduce our WiBACK architecture which is based on the consolidate outcomes of the EU FP7 project CARMEN, where a consortium of operators and research institutions had been studying, specifying and validating a heterogeneous multi-radio carrier-grade wireless mesh network architecture. The next section discusses the characteristics of UDTs and Unidirectional Link (UDL) support in wired as well as wireless networks. Following that, we evaluate UDL support by typically WMNs, Multi-Radio Wireless Mesh Network (MR-WMN) or Mobile Adhoc Networks (MANETs) protocols and discuss how far those approaches are applicable to our WiBACK architecture. Next, we investigate the impact of cell or link ranges, especially on multicast tree computations and then discuss Traffic Engineering (TE) and possible monitoring approaches covering UDTs. Based on this, we discuss the MPLS protocol suite, including protocols such as Resource ReSerVation Protocol (RSVP) as well as the Path Computation Element (PCE) architecture. Concluding, we will summarize the state of the art regarding UDT support, identify the open issues and present our tentative *design* of the integration of UDTs into the WiBACK architecture.

2.1 The WiBACK Architecture

The EU FP7 STREP project CARMEN[1] [40, 5, 41] aims at studying and specifying a cross-layer MR-WMN supporting carrier-grade triple-play services in future heterogeneous mobile/fixed network operator environments:

---

[1] http://www.ict-carmen.eu

" …Inferring that future mobile operator networks will be comprised of a common all-IP core network and several back-haul networks, the CARMEN back-haul network will support existing access technologies by exploiting cost-efficient mesh networking technologies. The project proposes to integrate heterogeneous wireless technologies (i.e. Wireless Local Area Network (WLAN), Worldwide Interoperability for Microwave Access (WiMAX), DVB and Universal Mobile Telecommunications System (UMTS)) in a multi-hop fashion in order to provide ubiquitous Internet access in a scalable and efficient manner…

…The CARMEN project is developing solutions to enable wireless mesh networks comprised of heterogeneous radio technologies to deliver carrier-grade services. The project is a 3 year project part funded by the EU: the project comprises of 8 European partners 2 network operators, 2 equipment manufacturers, 1 research institute and 3 universities. The project commenced in January 2008…

…An important issue being addressed within the project is a means to provide support for different radio interfaces having potentially quite different characteristics. This will be addressed through a so-called Abstract Interface (AI) which hides many of the complexities of the radio interface, making it easier to develop solutions for routing and network management for disparate radio interfaces.…

…CARMEN focuses on three aspects, namely technology, message transfer and self-configuration & management to ultimately provide a complete solution for setting up and maintaining a cost-effective carrier grade wireless mesh network… "

In the following subsections we will introduce our WiBACK architecture which is based on the consolidated outcomes of the CARMEN project and the results of this study, which addresses the issues related to UDT integration. Due to the involvement of the author in CARMEN from the preparation phase, the support for UDTs has been considered already in the early discussions while the actual work was performed during the course of the project and beyond.

2.1.1 Overview

The scope of the cross-layer WiBACK architecture is to provide or extend existing back-haul capacity which might range from single-hop long distance wireless connectivity to multi-hop connectivity with up to ten hops in urban and rural environments in developed or emerging regions [38]. The interface to external networks at WiBACK Gateway (WGW) or WiBACK Access Point (WAP) nodes can be realized via regular Internet Protocol (IP), Proxy Mobile IP (PMIP) [42] or MPLS trunking, see Figure 2.1.

The WiBACK architecture builds on proven technologies which have been extended to support heterogeneous wireless technologies. The control plane builds upon an extended IEEE 802.21 architecture, which allows for a hardware-independent and modular architecture design, see Figure 2.2. The data plane builds on Traffic Engineering (TE) principals incorporating MPLS-based forwarding, constraint-based path computation following the PCE concepts and a model to describe wireless channel resources. Those established technologies have been adapted to exploit the heterogeneity of wireless technologies including

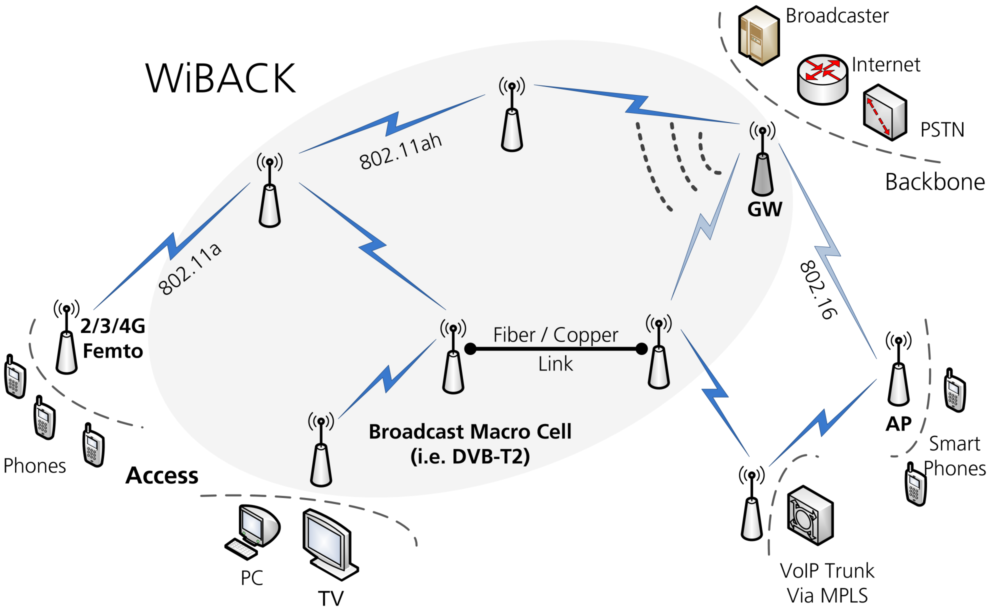

Figure 2.1: The WiBACK architecture integrates heterogeneous technologies supporting mobile or fixed terminals as well as trunked payload.

unidirectional broadcast technologies such as DVB. Hence, the WiBACK design addresses the requirements of a wireless back-haul architecture with a novel and wholistic approach incorporating proven protocols from the network operator domain, where possible.

The typical wireless network management components such a topology discovery, radio resource management, link monitoring, path computation and mobility management can be implemented as User Modules using the IEEE 802.21 messaging mechanism. This differentiates our approach from typical Network layer routing protocols, which integrate similar functionality in one protocol and are often agnostic to physical hardware capabilities.

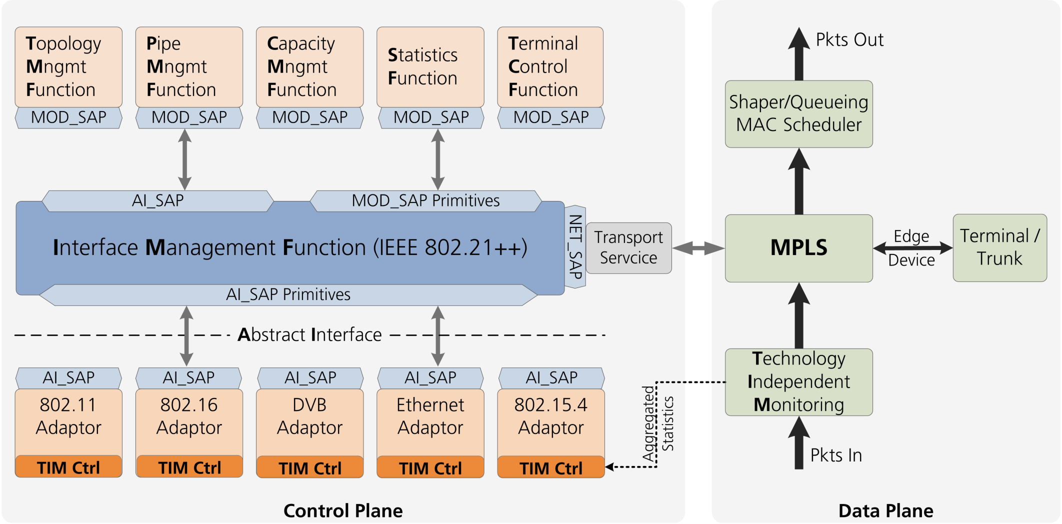

Figure 2.2: The WiBACK node architecture consists of an MPLS-based data plane and an IEEE 802.21-inspired control plane

The WiBACK architecture operates on so-called *Pipes* which are MPLS LSPs extended with dedicated per-hop resource allocation. *Pipes* are used as aggregates enforcing resource isolation as well as fairness among traffic classes or *Pipes* of the same traffic class. For each segment of an LSP, the end-to-end traffic class is mapped onto the respective per-hop Data Link layer traffic class.

Compared to traditional wired or fixed micro-wave-based operator back-haul networks, meshed WiBACK networks offer simplified deployment and maintenance processes due to their flexible self-management characteristics [30]. Those allow for the use of more cost-effective packet-switched equipment, such as IEEE 802.11, 802.16 or 802.22 and also support the integration with existing technologies such as DVB, point-to-point micro-wave, optical or even wired solutions, see Figure 2.1.

For the WiBACK architecture to be considered as an alternative for a rather over-provisioned operator back-haul network, it must meet similarly strict requirements such as guaranteed QoS, high availability and predictable behavior in high load situations in order to support the provisioning of the triple-play service mix todays customers expect. Broadcast services such as TV or radio can introduce a high load on capacity-constrained interference-sensitive wireless links [43]. To address this issue, our UDT-aware WiBACK architecture integrates broadcast technologies such as DVB. This allows the spectrum as well as capacity management components to dynamically shift such multicast traffic from the regular wireless to more efficient broadcast technologies, possibly depending on content, customer demand, as well as their density and distribution. Hence, the WiBACK architecture enables the re-use of the existing broadcast infrastructure exploiting the benefits of the usually longer range of such broadcast cells as well as their higher spectral efficiency [23].

In the following subsections we first summarize the IEEE 802.21 standard and present the amendments proposed by the CARMEN project. Then, we introduce the WiBACK control plane and its individual components. Following that, we present the data plane design.

2.1.2 IEEE 802.21

The WiBACK control plane is heavily inspired by the IEEE 802.21 standard. Although the purpose of this standard is to facilitate seamless handover between heterogeneous technologies, the concepts of media abstraction can easily be extended for other purposes as well. In the following, we will first introduce the IEEE 802.21 architecture and then describe the CARMEN amendments to support the management of heterogeneous wireless nodes and their interfaces.

The main goal of the IEEE 802.21 standard [44, 45, 46] is to facilitate a handover between heterogeneous access networks including wired and wireless technologies, as well as 802 and non-802 networks, by providing link layer intelligence for the upper layers and, thus, improving the user experience of mobile devices by enabling a seamless handover wherever this is supported by the underlying network environment. This includes, for example, a scenario where a user gets handed off from a IEEE 802.3 network to an 802.11 or a mobile terminal which is handed off from a 3G cell to a 802.11 network during an ongoing call. For this purpose, IEEE 802.21 defines a media-independent abstraction layer which provides a uniform interface to the higher layers allowing for implementing and designing upper layer modules in a technology-independent way.

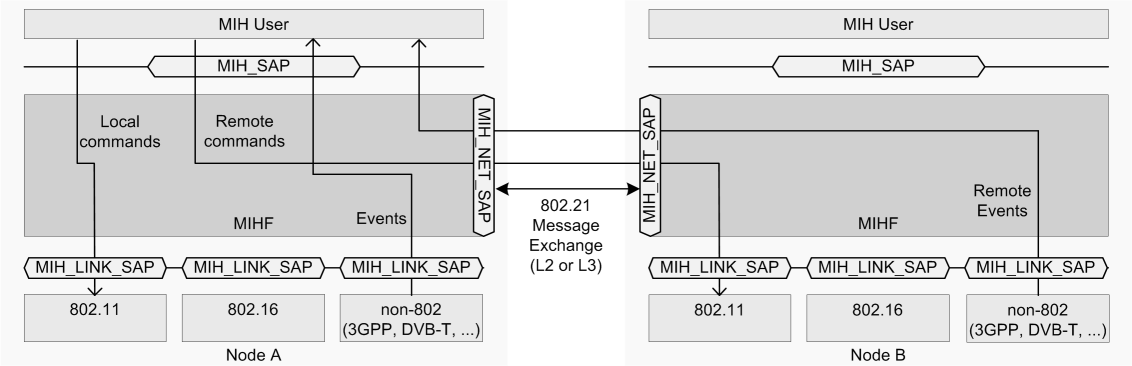

Figure 2.3: IEEE 802.21 architecture

Figure 2.3 depicts the general 802.21 architecture. The IEEE 802.21 standard specifies a framework consisting of two main components, namely the Media Independent Handover Function (MIHF) and the MIH Users as well as several Service Access Points (SAPs). The MIHF is the central entity in each 802.21-enabled device. It provides a common set of media-independent primitives to the upper layer, the MIH Users, via the MIH_SAP and maps those to the corresponding technology-specific link layer primitives via the MIH_LINK_SAP. In addition to that, 802.21 defines the MIH_NET_SAP which offers an interface of the MIHF providing resilient MIH message exchange with a remote MIHF utilizing Layer2 oder Layer3 transport mechanisms. It is important to note that the actual handover decision as well as the handover execution are done by the MIH User(s) and outside of the scope of the IEEE 802.21 standard. The purpose of 802.21 is to enable a (seamless) handover by providing appropriate information, event notifications and commands to an MIH User. In order to do so MIHF defines three main services:

- The Media Independent Event Service (MIES) provides a mechanism for classification, filtering and reporting of events related to changes in link status, link quality or link characteristics. Events originate at the lower layers or the MIHF. A MIH User needs to explicitly subscribe to a particular event in order to receive it. For this purpose the MIHF maintains an appropriate subscription list. A MIH User can also subscribe for remote events which are generated on a different node in the network. It should be noted that events are delivered asynchronously by the MIHF and the receiving MIH Users don't need to take any action when being notified of an event.
- The Media Independent Command Service (MICS) allows the upper layer to manage and control mobility related functions in the lower layers, e.g. initiate the handover or turning off a certain radio. Command primitives are sent from an MIH User either

to the lower layers or the MHIF itself and can be delivered locally or remotely. In the latter case, the MIH User sends the command to the local IMF which will transport it to the appropriate destination node. In contrast to the MIES, command primitives are transmitted sequentially and follow the request-response message exchange pattern.

- The Media Independent Information Service (MIIS) enables the MIH Users to obtain information on the serving network and neighboring networks in a geographical region. This even includes information on networks of other technologies than the network to which the mobile node is currently attached to, e.g. a node which is connected to a 3G network is able to obtain information on accessible 802.11 WiFi networks within the same geographical region. MIIS either supports a pull mode, which is primarily used by a mobile node to gather information from the network, or a push mode which gives, for example, the operator the possibility to push handover policies onto the mobile devices.

Besides IEEE 802.21, there are other approaches which provide similar functionality: WiOptiMo [47] for example provides a small Java application allowing for a handover particularly of client/server applications by providing a special socket that switches the underlying network invisible for the actual application. In contrast to that, Unlicensed Mobile Access (UMA) or Generic Access Network (GAN)[48] offer a proprietary solution for operators to allow access to their networks not only via cellular but also via unlicensed radio access technologies, such as IEEE 802.11, by tunneling the connections back to the operator networks through a secure connection. Thus, UMA provides an easy solution for offloading traffic from cellular networks such as GSM or 3G to a Wifi Hot-Spot. However, as opposed to those approaches, IEEE 802.21 aims at enabling fast hand-overs by providing appropriate Data Link layer mechanisms, which make it fairly flexible in terms of extending the foreseen use case as described in the following paragraphs.

The basic idea behind IEEE 802.21 can be described as a Media Independent paradigm since it hides the technology specificities from higher layers by providing a common set of primitives and, thus, reduces their complexity. The concept of media independence, however, is not limited to supporting hand-overs between heterogeneous networks, but can rather easily be exploited to simplify a larger variety of applications and operations in contemporary operator networks, such as bootstrapping, routing or network management.

CARMEN Amendments to IEEE 802.21

The CARMEN-based WiBACK architecture design adopts the general IEEE 802.21 [45] architecture. The main difference between the IEEE 802.21 standard and the WiBACK architecture is the respective target application. IEEE 802.21 focuses on media independent hand-overs of mobile terminals, while the WiBACK architecture aims at managing heterogeneous wireless networks in a media independent fashion. The majority of the primitives defined by IEEE 802.21 can also be utilized for non-handover related purposes, such as managing local and remote radio technologies in a media independent manner. This characteristic is a main requirement for the WiBACK architecture, which builds upon IEEE 802.21 primitives where possible and extends those with mesh specific requirements. A subset of these primitives has been discussed within the IEEE 802.21b working group and, eventually, included into the IEEE 802.21b standard.

As depicted in Figure 2.4, the Media Independent Handover Function++ (MIHF++) of the WiBACK architecture extends the IEEE 802.21 MIHF with primitives specific to wireless network management, therefore the name Interface Management Function (IMF) has been chosen to reflect its responsibilities which go beyond *Media Independent Handovers*. This amendment to IEEE 802.21 provides a single interface for realizing User Terminal (UT) handovers as well as building and managing a heterogeneous wireless networks.

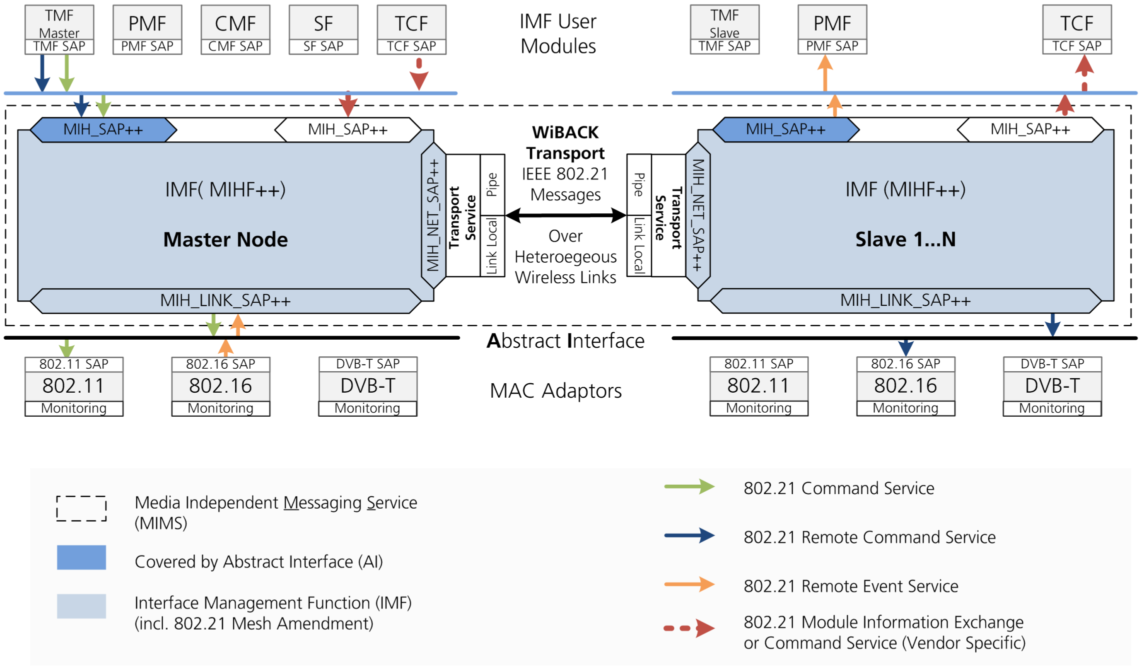

Figure 2.4: The IMF extends the IEEE 802.21 MIHF by User Module-to-User Module communication

In Figure 2.4, the dark colored MIH_SAP++ corresponds to abstract interface primitives of the WiBACK architecture, while the white one corresponds to module specific primitives. These primitives have been separated due to the fact that the module specific primitives of the WiBACK architecture are mainly related to wireless network management, while the Abstract Interface (AI) provides a more generic interface for managing lower layers. In IEEE 802.21 standard, the dark colored MIH_SAP++ would correspond to link events and commands and the white one would correspond to handover-related commands.

IEEE 802.21 does not foresee any direct communication between two User Modules via the MIHF. It is assumed that higher layer entities are already aware of each other and rely on vendor specific means of communication, such as IP-based protocols. In the WiBACK architecture it is assumed that the IMF is aware of the User Modules responsible for a specific message coming from a remote IMF. This mapping of message identifiers to the responsible module is maintained via the following module registration procedure between a module and the IMF.

Each module may provide a set of primitives. Combined, these primitives define the SAP of a module. Each primitive is identified by a unique message identifier (MID). During the registration process of a module to the IMF the module conveys the set of MIDs of its SAP. Thereby the IMF can forward an incoming message to the corresponding module by the MID. In this context, the MIDs can be seen as service identifiers where a module indicates to provide the service identified by a specific MID.

With this extension of User Module-to-User Module communication, the WiBACK architecture can rely on more complete a Remote Procedure Call (RPC)-like messaging mechanism among WNs, which is referred to as Media Independent Messaging Service (MIMS).

2.1.3 Terms and Definitions

In the following, we summarize the relevant terms that have been defined to help describe the WiBACK architecture.

- **WiBACK Node (WN)** - A WN is a node within the WiBACK network that is equipped with WiBACK capabilities. WNs are capable of forwarding and monitoring traffic among each other. A WN may also act as a WiBACK Gateway (WGW) to the back-bone or as a WiBACK Access Point (WAP) node to connect User Terminals (UTs) or as a WiBACK Coordinator (WC). WNs are aware of the payload QoS requirements and support four traffic classes: *best effort*, *video*, *voice* and *management/emergency*. WNs may be equipped with one or more heterogeneous radio interfaces, such as IEEE 802.11, 802.22, 802.15.4 or 802.16 and DVB interfaces.

- **WiBACK Coordinator (WC)** - A WiBACK network is managed by WC nodes which execute the Topology Management Function (TMF) and Capacity Management Function (CMF) *Master* entities. WCs may be replicated for redundancy purposes and may be connected via the Core Network or via *Management Pipes*

- **WiBACK Gateway (WGW)** - A WGW is a WN that provides connectivity to the back-bone network and is located at the boundary between the back-bone networks and the WiBACK network. The WGW has at least one standardized interface into the core network - typically, this is a wired interface using, for example, Ethernet, but it may also be a wireless connection.

- **WiBACK Access Point (WAP)** - A WAP is a WN with the capability to provide (wireless) access for UTs. WAPs may be equipped with one or more radio interfaces dedicated to UT access. Such interfaces do not carry WiBACK back-haul traffic and their capacity is not managed by the WiBACK network management system. The set of radio technologies employed on access links may be different from those used within the WiBACK network. For example, WAPs are envisaged to be equipped with GSM or 3GPP air interfaces while traffic generated by connected UTs is carried as regular WiBACK back-haul payload.

- **User Terminal (UT)** - A UT is an end-user device which uses the WiBACK network to obtain access to services. It can be both a fixed or mobile device and no specific form factor is assumed. As a minimum requirement, UTs are expected to be fully compliant with the standards of the respective access technology (radio or fixed). UTs may have advanced capabilities (e.g. IEEE 802.21 support, context-awareness functionality) to avail themselves of all of the services provided by the WiBACK network. Alternatively, if they do not have advanced capabilities, they can still use the WiBACK network as an access network and obtain an acceptable level of service.
- **Core Network** - The core network (or back-bone network) is an IP-based infrastructure which provides connectivity between WGW nodes as well as to external entities, i.e. the Internet at large.

Connectivity Definitions

The following terms have been defined to describe connectivity among WNs:

- **Back-haul Link** - A back-haul link is a connection between two interfaces of adjacent WNs and is often referred to as *link*. It is dedicated for traffic of one or a group of traffic classes. To support established traffic engineering approaches, a back-haul link is described as a unidirectional resource and bidirectional connectivity is reflected by a pair of unidirectional links. There may be more than one link between any two adjacent WNs on top of the same or different radio technologies.
- **Path** - A path is a sequence of links describing the way packets have to traverse across a WiBACK network.
- **Pipe** - A pipe is a path with an assigned identifier, installed forwarding states and allocated link resources on the outgoing interfaces of each link (e.g. traffic class, reserved bandwidth)

Identifiers

Entities and resources in the WiBACK architecture are identified by identifiers. The following identifiers have been defined:

- **NodeId** - An EUI64 address (derived as a hash of all local interface addresses)

- **InterfaceId** - A Layer 2 interface address encoded as specified by 802.21
- **LinkId** - A pair of Layer2 interface addresses encoded as 802.21 MIHLinkId
- **PipeId** - Consists of the NodeId of the ingress node and a locally generated/unique descriptor

2.1.4 Control Plane

In the following sections we summarize the functionality of the individual control plane components as depicted in Figures 2.4 and 2.2.

Abstract Interface

The Abstract Interface (AI) as designed by the CARMEN project [49], provides primitives to query and configure interface properties. Extending the MIH_LINK_SAP, additional primitives to extend IEEE 802.21 beyond support for Media Independent Handover (MIH) have been defined. In this work we build on the CARMEN-proposed AI primitives while carefully updating them where required to add native UDT support, see Table 2.1:

Primitive Name	**Description**
AI_EventSubscribe	Caller requests subscription to an event
AI_EventUnSubscribe	Caller requests un-subscription from an event
AI_LinkAllocateResources	Allocates LSP resource on an outgoing link
AI_LinkModifyResources	Modifies LSP resource on an outgoing link
AI_LinkReleaseResources	Releases LSP resource on an outgoing link
AI_RadioGetRadios	Returns a list of local InterfaceIds
AI_RadioGetProperties	Returns interface properties
AI_RadioGetEnvelope	Returns parameters describing the spectral envelope
AI_RadioJoinCell	Instructs interface to join a cell
AI_RadioLeaveCell	Instructs interface to leave a cell
AI_RadioCalibrateLink ⋆	Instructs interface to calibrate a link
AI_RadioSetupBeacon	Parameterizes the WiBACK beacon
AI_RadioBeaconScan	Triggers a beacon scan
AI_RadioChannelScan	Triggers a channel scan
AI_LinkDown	Indicates a LINK_DOWN event
AI_PipeDown ⋆	Indicates a PIPE_DOWN event
AI_RegulatoryEvent ⋆	Indicates a regulatory event

Table 2.1: AI primitives used by the TMF to manage heterogeneous wireless interfaces

The primitives AI_RadioCalibrateLink, AI_PipeDown and AI_RegulatoryEvent have been introduced as a results of this thesis.

The WiBACK architecture uses different identifiers for its resources, such as a NodeId, an InterfaceId, a LinkId and a PipeId. The first three identifiers have been taken from the IEEE 802.21 standard, with the NodeId being the equivalent of the *MIHFId*. InterfaceIds are generated from the hardware address of each interface and a LinkId uniquely identifies a link between two interfaces. NodeIds and InterfaceIds are considered to be unique across a WiBACK network and TMF *Masters* must verify this and reject *Slaves* with non-unique identifiers.

Interface properties can be queried using the AI_RadioGetProperties primitive. Among the properties reported by this primitive, as shown in Table 2.2, are the directionality, a list of supported channels and the operational mode.

Parameter	**Type**	**Description**
InterfaceId	INTERFACE_ID	Id of reporting interface
Directionality	ENUM	Rx, Tx, Duplex
Channels	LIST(CHANNEL)	Support Channels
OperationalMode	ENUMERATION	Infrastructure, Ad-Hoc, Broadcast
RegulatoryInfo	REG_INFO	Channel scanning and reaction times, etc.

Table 2.2: UDT-relevant interface properties reported by the AI_RadioGetProperties primitive

The AI provides the architectural support to address regulatory requirements, by providing the *RegulatoryInfo* object where an interface can report constraints such as minimum channel scanning times or maximum reaction times to free a channel can be described. The AI also provides the AI_RegulatoryEvent which may be triggered by radios detecting a collisions with a prioritized user, such as a weather radar in Unlicensed National Information Infrastructure (U-NII) band. Our Topology Management Function (TMF) design considers such regulatory spectrum allocation limitations and can be extended to address such issues either via external mechanism or databases.

Transport Service

The IEEE 802.21 messaging service provides node-local and, via the *NET_SAP*, remote messaging, while the actual transport is not defined by the IEEE 802.21 standard. The WiBACK architecture provides a *TransportService* via so-called *Management Pipes* or link-local multicast transmissions.

Topology Management Function

To manage such a network build upon heterogeneous wireless technologies, the WiBACK architecture introduces a Topology Management Function (TMF) to optimize the usage of scarce radio spectrum resources. The TMF may implement a ring-based master/slave approach where *Masters* located at WiBACK Coordinator (WC) nodes first bring up their own radio interfaces and determine the optimal radio configuration. This may be computed based on the capabilities of the radio interfaces and the ambient spectrum usage assessed by passive channel utilization analysis. Once this process is complete, a *Master* starts sending WiBACK beacons on all its active interfaces to inform adjacent *Slave* nodes about its availability.

Slave nodes determine their configuration during the bootstrap phase and then switch into the *beacon scan* mode in which they periodically scan all administratively permitted channels for WiBACK beacons sent by a *Master* node or already associates *Slave* nodes. Once they detect WiBACK beacons they will attempt to associate with the sending nodes according to a locally determined order. Since this decision on the node to associate with is based on local knowledge only, it may not be optimal considering the overall network topology or not fit with the TMF *Master's* overall optimization policies. Hence, a *Master* might reject the association request and suggest an alternative node or interface for association.

If a *Master* detects a node or link failure, the affected nodes or links will be marked as *down* and the *Master* will attempt to repair the remaining network according to its optimization criteria. Whenever *Slaves* detect a connectivity failure towards their *Master*, they will jump back into the bootstrap phase and attempt a new association.

Capacity Management Function

While the TMF *Master* is responsible for spectrum and channel management, the CMF is tasked with managing the capacity of the links or cells activated by the TMF. CMF entities are located at WiBACK Coordinator (WC) nodes. In order to quickly react to load variation or link degradations, the CMF typically runs at a faster, i.e. sub-second, timescale compared TMF. Figure 2.5 depicts the relationship between the TMF, the MAC Adaptors and the CMF concerning physical and logical link resource management, see [15] for a more detailed discussion.

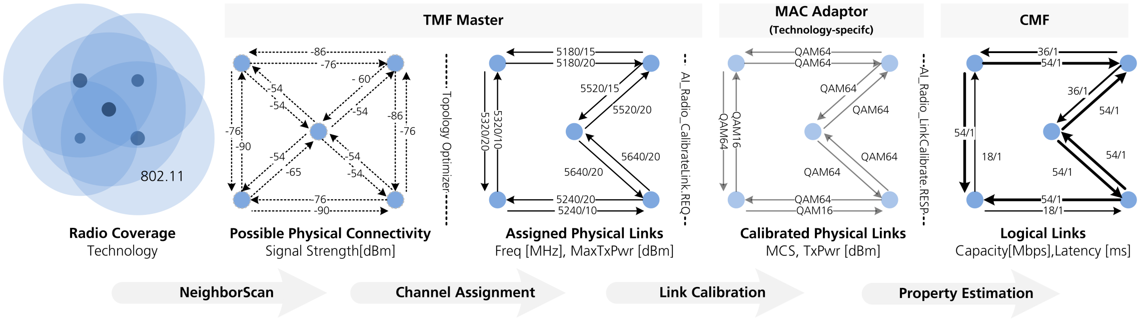

Figure 2.5: TMF detects physical connectivity among nodes, choses the optimal links and assigns orthogonal channels, where possible. Then TMF calls AI_Radio_CalibrateLink to determine the optimal radio configuration. Upon successful completion, the response message returns the resulting logical link properties and the parameters of the resource model. This information forms the basis of CMF's capacity management.

Upon association of new *Slaves*, the TMF computes the optimal channel configuration out of all possible physical links among the adjacent WiBACK nodes and their radio interfaces. Then TMF triggers the AI_Radio_CalibrateLink primitive and pushes the set of activated logical links and their properties to CMF for capacity allocations of *data Pipes* within the WiBACK network. In the multicast case, the CMF may leverage its knowledge about the number of multicast receivers per possible transmitting interface in order to exploit the benefits of Link Layer multicast transmissions and to utilize long range unidirectional broadcast cells where available.

The CMF design is based on the concept of a centralized stateful Path Computation Element (PCE) where the relevant messages of the Path Computation Element Protocol (PCEP) are mapped onto MIH-style primitives. For each link, the stateful CMF keeps track of the available as well as the currently allocated resources. In order to maintain an up-to-date view of the overall resource state of the *Pipes* under its control, CMF subscribes to AI_PipeDown events to allow for fine-grained reactions depending on the QoS requirements of the affected *Pipes*.

Pipe Management Function

The PMF, as proposed in this study, is tasked to set up, modify and remove *Management Pipes* or *Data Pipes* and supports regular downstream-assigned and upstream-assigned multicast LSPs. During a *Pipe* setup procedure, the PMP also allocates the associated *Pipe* resources at each outgoing interface along the path. This information can be used to monitor the proper QoS-handling of an LSP but may also be used to allocate MAC layer resources by, for example, configuring traffic shapers or configuring IEEE 802.16 service flows or IEEE 802.11e queuing parameters. *Pipes* are unidirectional resources, hence to form a duplex connection among any two WiBACK nodes a pair of *Pipes* is required.

Terminal Control Function

The Terminal Control Function (TCF) is an optional component of the WiBACK architecture which provides the functionality required to directly connect UTs to WiBACK nodes, thus providing WAP or *HotSpot* functionality. Depending on the implementation, TCF may provide multiple services, such as UT detection and hand-over control as well as local capacity management to match UT traffic demands with according back-haul capacity. The main task, however, is to keep UT/*Pipe* bindings between WAP and the WGW nodes the up to date.

The TCF may be implemented to support seamless terminal mobility via, for example, integration with Proxy Mobile IP (PMIP). If mobility is not a major concern, for example, due to rather fixed or nomadic UT usage patterns, less complex approaches such as our QoS-aware LAN Emulation (QLANE)-style mechanism may be implemented [12].

Statistics Function

The Statistics Function (SF) is typically located at *Master* nodes and can be instructed to collect interface, link or *Pipe* statistics from selected or all *Slave* nodes. This information can then be used to detect, for example, longer term usage or interference patterns.

2.1.5 Data Plane

The WiBACK data plane consists of an MPLS-based forwarding engine which utilizes underlying hardware-specific traffic shapers or MAC schedulers, where available. To collect local interface, link and *Pipe* per-packet statistics, the data plane also contains a measurement module.

MPLS-based Forwarding

The WiBACK data plane is built on top of a lightweight MPLS forwarding engine which supports unicast and 1-to-N multicast LSPs. As described above, the WiBACK architecture refers to LSPs and their associated resource allocations as *Pipes*. *Pipe* state can be configured with node-local function calls, which are typically performed by the PMF. When installing or removing a *Pipe*, the PMF also allocates or releases the associated resources on the outgoing link of the *Pipe* via the AI_LinkAllocateResources or AI_LinkReleaseResources primitives which are provided by the respective MAC Adaptor of the affected interface. The MAC Adaptors may map those primitives onto hardware-specific functions, such as internal resource allocations, DiffServ queue parameters, or traffic shaper configurations.

Technology Independent Monitoring

The WiBACK architecture separates per-packet statistics gathering from analysis of such data and configurable event creation. The Technology Independent Monitoring (TIM) component performs per-packet measurements of PHY, LINK or LSP objects and periodically pushes an aggregated statistics summary towards the node-local *TIM Ctrl* submodule, which is located within the respective MAC Adaptor. Here, this information may be aggregated further, evaluated in order to trigger events such as LINK_DOWN or PIPE_DOWN, or reported to the Statistics Function (SF).

2.2 Unidirectional Connectivity

Unidirectional network technologies are mainly present in the wireless domain, while virtually all wired network technologies such as Ethernet, SONET or TokenRing provide a bidirectional MAC layer and an interface would be considered broken if communication would fail into one direction and the affected interface card or wire would simply be replaced. Thus, the use of UDTs as such is not considered in wired networks. In the remainder of this section we therefore focus on UDTs in the wireless domain.

2.2.1 Reasons for Unidirectional Connectivity in Wireless Networks

Exposed unidirectionality of wireless technologies in wireless networks can be separated into two categories:

- Unidirectional behavior of malfunctioning bidirectional technologies (i.e. IEEE 802.11 in *Ad-Hoc* mode)
- Natively unidirectional technologies (i.e. DVB, ATSC)

Bidirectional Technologies

UDLs can be a frequent, temporary or permanent, phenomenon in typical WMN, Wireless Sensor Network (WSN) and especially MANET deployments [37, 39]. The aspects causing UDLs have been investigated in-depth for *ad-hoc* technologies and may occur when radios are operated within the so-called *Transitional Region* [37]. This may be due to, e.g. asymmetric transmit power settings, local interferences or an asymmetric drop in signal strength after channel reassignments due to, i.e. higher cable attenuation or lower antenna gain at the selected frequency, see Figure 2.6. *Infrastructure*-mode technologies do not expose this issue since they only operate in fully bidirectional connected mode and manage cell membership internally.

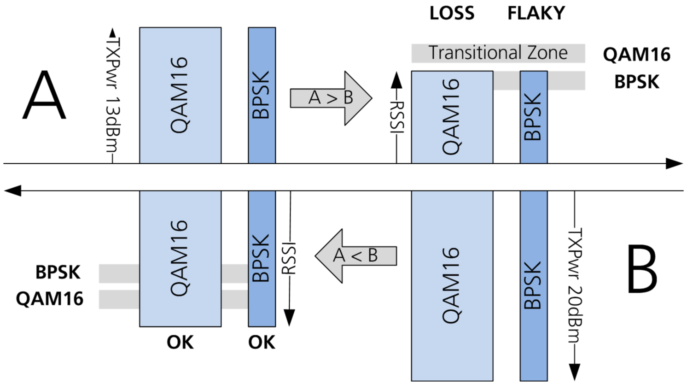

Figure 2.6: Typical scenario in *ad-hoc* networks rendering the link between nodes A and B effectively *unidirectional*

Figure 2.6 depicts a common scenario where an *ad-hoc* technology exposes unidirectional connectivity due to asymmetric transmit-power settings. Especially when the signal quality falls within the *Transitional Region*, the chance of frame corruption is rather high and increases significantly with an increasing frame size [37]. This might cause situations, where short and robustly modulated (i.e. using BPSK) management or control frames can still be exchanged, while QAM16-modulated data frames experience very high loss figures. In the worst case, even control or management frames can no longer be exchanged. While in the first case, the link would have become unidirectional from the data connectivity point of view, the MAC protocol might still be functional. In the latter case, the MAC protocol is no longer functioning correctly, since the predominant wireless technologies in such networks, IEEE 802.11, 802.15.4, expect bidirectional connectivity. Nonetheless, it might well be the case that some data frames might sporadically still be received.

Hence, the UDLs detected by MANET routing protocols are not physically unidirectional. Rather, the link budget in one direction is almost exhausted, so that the MAC protocols no longer function properly leading to non-deterministic behavior where, for example, some nodes might not sense ongoing traffic and therefore start sending themselves causing collisions on the channel. Most MANET routing protocols therefore attempt to detect such UDLs and often black-list the affected links, since, as Network layer protocols, they have no means to fix or adjust such flaky links.

Unidirectional link detection in MANET or WMN protocols typically works via layer three connectivity verification. If a link does not forward packets in a given direction it is marked as unidirectional, and depending on the protocol, avoided (e.g. Ad hoc On-Demand Distance Vector Routing Protocol (AODV), see RFC3561 [50]), or utilized if possible (e.g. Dynamic Source Routing Protocol (DSR), see RFC4728 [51]). The MAC layer uncertainties cannot be addressed and therefore pose potential trouble for all nodes sharing this channel, since channel access coordination can not be assumed to work reliable. This might be acceptable for *best effort* solutions, but such unpredictable channel access renders such a link unusable for carrier-grade networks. Such UDL conditions should therefore be detected by the monitoring system the of WiBACK architecture and the affected link be reconfigured or disabled.

Unidirectional Technologies

Natively unidirectional technologies considered in this work are mainly the broadcast technologies standardized by, for example, the DVB consortium or the ATSC. Such broadcast cells consist of one logical transmitter and a potentially unlimited number of receivers. In certain operational modes such as Single Frequency Network (SFN), multiple transmitters might be active transmitting the same signal, effectively enlarging the coverage area. Transmitters and receivers have distinctively different properties and can easily be identified as such.

DVB

Digital Video Broadcast (DVB)[2] refers to collection of open standards for digital television, which are maintained by the DVB Project. DVB systems may broadcast data via different physical media, such as:

- satellite via DVB-S, DVB-S2 and DVB-SH
- cable via DVB-C, DVB-C2
- terrestrial via DVB-T, DVB-T2, DVB-H

The main difference among such DVB broadcasting systems are the deployed modulation schemes and error correcting codes, which have been chosen to best address the different physical constraints. In the context of this study, the main focus is on terrestrial wireless networks, hence DVB-T and preferably its successor DVB-T2 are considered.

---

[2] http://dvb.org

Conceptually, the WiBACK architecture could also support the cable or satellite counterparts, where the technology specific differences would be addressed within the MAC Adaptor.

In addition to digital multimedia content, DVB also supports data broadcasting and can optionally support return channels via i.e. DVB-RCS[3] which are mainly used in satellite-based scenarios.

DVB-T only supports a constant modulation and coding scheme and relies on the Multi Protocol Encapsulation (MPE) or the Unidirectional Light Encapsulation (ULE) [52] on top of MPEG *Transport Streams* to support i.e. Link layer Protocol Data Unit (PDU) broadcasting. DVB-T2, on the other hand, supports per-packet modulation and coding adaptations in Advanced Coding and Modulation (ACM) mode, and supports the more flexible Generic Stream Encapsulation (GSE)[4] to broadcast i.e. IEEE 802.3 Ethernet frames.

In contrast to packet-based wireless technologies, such as IEEE 802.11 or 802.16, a DVB transmitter provides a permanent *carrier* which receivers maintain a *lock* on. If no actual data is available for transmission, so-called *stuffing frames* are sent for receivers to maintain the *lock*, thus minimizing the PHY synchronization overhead compared to i.e. per-packet preambles used by technologies such as IEEE 802.11.

DVB can broadcast multiple streams via so-called *multiplexes* over a single carrier. The structure and contents of such a *multiplex* is periodically announced via information tables such as a Service Description Table (SDT) or Program Map Tables (PMTs), which among other information list the Packet IDs (PIDs) of each stream. In order to detect available services, a receiver passively scans all available channels and interprets the received SDT and PMT tables.

In the WiBACK context, receivers can either filter out *WiBACK* services reported in the SDT or may assume a well-known PID to be used for WiBACK streams. Such streams can then be returned as the result, in terms of the channel and PID, of the AI_RadioBeaconScan.Response primitive. The contents of the WiBACK beacon may be encoded as *private data* within the PMT or the beacon may be sent as a regular broadcast data frame. On the transmitter side, it depends on the equipment used as well as regulatory constraints, which parameters may be adjusted via the AI. Such parameters include the modulation and coding scheme, the channel as well as the TxPower. The specifics would have to be implemented within the respective MAC Adaptor.

ATSC

Similar to the DVB Project, the Advanced Television Systems Committee (ATSC) [5] maintains a set of standards for digital television broadcasting over terrestrial, cable, and satellite media. ATSC specifies different modulation schemes and error correcting codes and

---

[3] http://dvb.org/technology/fact_sheets/DVB-RCS_Factsheet.pdf

[4] http://dvb.org/technology/fact_sheets/DVB-GSE_Factsheet.pdf

[5] http://atsc.org/cms

different channel bandwidths. Analogue to MPE or GSE in the DVB context, the ATSC A/90 Data Broadcast standard defines the encapsulation and broadcasting of data frames.

The specific differences between the DVB and the ATSC standards are outside the scope of the study, which focuses on the conceptual aspects integration of such UDTs, while technology-specific details would be addressed by the respective MAC Adaptor.

Identification of Unidirectionality

Typical IP-based routing protocols can not reliably determine the directionality of an interface. Hence, protocols such as OSPF [32] rely on mechanisms such as HELO messages in order to detect if links can be considered bidirectional or unidirectional. Those mechanisms can not differentiate between the two cases of unidirectionality we have identified. They would typical just black-list connectivity provided via DVB-T, for example.

Most WMN routing protocols have been developed with a focus on single-radio nodes and therefore assume that the wireless interfaces have been pre-configured and have no means to modify such configuration. Hence, they aim to utilize the discovered topology most efficiently.

The goal of the WiBACK architecture, however, is to provide mechanisms to enable radio planning and proper channel assignment following administered optimization policies. Precise knowledge about the directionality of the underlying technology can therefore be exploited by the optimization algorithm to detect possibly malfunctioning bidirectional links, in order to either disable or to reconfigure them.

2.2.2 Virtual Return Channels

In the following we discuss two different approaches to provide bidirectional connectivity for UDLs. While the Link Layer Tunneling Mechanism (LLTM) addresses terrestrial or satellite DVB links, Bidirectional Routing Abstraction (BRA) focuses on UDLs in WMNs.

LLTM

Unidirectional Link Routing (UDLR) [53] provides a mechanism to emulate full bidirectional connectivity between all nodes that are directly connected by a unidirectional link. The nodes on the receive-only side use a tunneling mechanism to forward link layer datagrams back to send-only node via a separate bidirectional IP connectivity. A typical tunneling protocol used in combination with UDLR is Generic Routing Encapsulation (GRE)[54], which is an IP protocol. Since the tunnel encapsulates data link layer frames, UDLR is considered transparent to higher layer protocols.

However, UDLR may cause problems with mechanisms such as IPv6 Duplicate Address Detection (DAD) which sends out Neighbor Solicitation packets to detect if other nodes already use the probed address. The standard DAD mechanism has been designed for regular link-local broadcast domains, such as an Ethernet segment. It assumes that only nodes other than itself will receive and respond to this solicitation packet, while in the

UDLR case, it will receive its own solicitation packet and may therefore assume the address to be already allocated.

UDLR uses the Dynamic Tunnel Configuration Protocol (DTCP) embedded in its downstream UDL to announce to receive-only nodes at which IP address the tunnel endpoints for the return channel can be reached. Each node with a send-only interface can signal where the tunnel endpoints for a specific unidirectional link can be found. DTCP allows for multiple tunnel endpoint addresses to be announced, but can not control, which end-point is eventually used by a receive-only node.

Since UDLR transparently provides bidirectional link layer connectivity between the nodes, normal routing protocols could be used without a need for modifications incorporating unidirectional link support. The main use case for LLTM is to provide *best-effort* virtual return links across foreign network clouds, such as the Internet. However, such a transparent return link poses a major problem where QoS resource allocations are required, since the higher layer protocols are not aware of the special nature of such a link which often crosses numerous heterogeneous networks and wireless cells until reaching the tunnel endpoint, see Figure 2.7. It is therefore not readily possible to perform per-hop resource reservations or proper *link* monitoring. Hence including UDLR links in QoS-constrained path computation algorithms or signalling protocols would require major adaptations. A possible approach to support QoS signalling was, for example, studied in [55], where the signalling was implemented based on the Next Steps in Signalling (NSIS)[56] architecture.

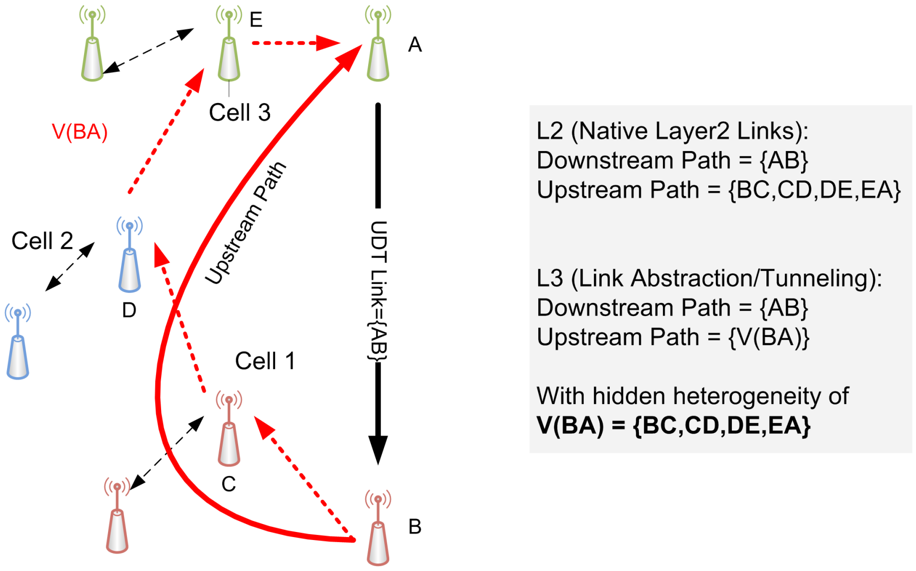

Figure 2.7: Link Layer Tunneling hides the underlying heterogeneity of the virtual return path V{BA}, while using native LSPs, all links are accounted for.

In the context of the WiBACK architecture, where the network is centrally controlled by the TMF or CMF entities, return links for UDTs including proper QoS allocations can readily be computed. Hence, the extra protocol layer introduced by LLTM can be avoided.

BRA

Bidirectional Routing Abstraction [39] uses a reversed distance vector algorithm to detect UDLs in MANETs. BRA provides an abstraction layer so that higher layer protocols only see bidirectional links. Support for QoS-constraint path computation suffers from similar implications as LLTM due the hidden heterogeneity of the multi-hop return link as described for UDLR.

[39] have done an analysis for MANETs where they identified three different causes that might lead to bidirectional links becoming highly asymmetrical in throughput or even unidirectional. Although the study was done for MANETs, the results should also be applicable for more static WMNs, since node mobility or nodes joining or leaving the mesh were not considered.

In their study, the authors have identified three sources of unidirectional behavior, which corresponds with findings by [37]:

- random signal irregularities
- external radio sources creating interfering signals at one node
- diversity of transmission power

The study shows that, depending on the scenario, up to 15% of the links in the investigated MANET scenarios might become unidirectional. Hence, if UDLs are not considered the connectivity of the mesh drops considerably. The authors also point out that the connectivity distribution is rather heavy-tailed which might lead to a sudden steep loss of connectivity if the ratio of UDLs is too high. Considering UDLs, the overall mesh connectivity increases and the distribution is less heavy-tailed.

The results of this study suggest that UDLs should be considered by single-radio mesh routing protocols to enhance the overall performance and resilience. Indeed, a number of extension for MANET routing protocols have been proposed to extend existing protocols to better utilize unidirectional links, see, for example [57]. UDL support of exemplary MANET or WMN protocols is discussed in Section 2.4.

2.3 Unidirectional Links in Wired Networks

Standard Internet routing protocols are not designed to support UDLs since they are very uncommon in the wired Internet infrastructure. In this section, mainly Interior Gateway Protocols (IGPs) such as OSPF and Routing Information Protocol (RIP) are considered, since a WiBACK network forms a rather interior network instead of a global backbone where Exterior Gateway Protocols (EGPs) such as BGP[58] are used.

2.3.1 Distance Vector Routing Protocols

Distance vector routing protocols such as RIP[59] inherently rely on bidirectional links to propagate and update their local distance vectors. RIP is a dynamic routing protocol which was initially defined in 1988 and belongs into the category of IGPs. The protocol has been enhanced several times to overcome limitations such as lack of authentication or support for Classless Inter-Domain Routing (CIDR) [60]. A modified version called Routing Information Protocol next generation (RIPng) [61] to support IPv6 has also been defined. All versions of RIP use a maximum hop limit of 15. In addition to the lack of a concept for routing areas this renders RIP not suitable for larger deployments. Since the hop count to a destination is the only routing metric, RIP can not be used to route depending on resource demands and availability. To overcome those limitations, Cisco Systems had implemented a proprietary protocol called Interior Gateway Routing Protocol (IGRP), which was replaced by the also proprietary Enhanced Interior Gateway Routing Protocol (EIGRP).

Especially in the wired Internet routing domain, distance vector routing protocols have been obsoleted by more capable link-state routing protocols such as OSPF or OSI's Intermediate system to intermediate system (IS-IS). For small-scale wireless ad-hoc networks, however, distance vector protocols are actively being developed because of their ability to establish communication among nodes without requiring any manual configuration, thus supporting the ad-hoc notion of such networks. Distance vector WMN routing protocols will be discussed in Section 2.4.

2.3.2 Link State Routing Protocols

OSPF [32] is a hierarchical link-state routing protocol designed for routing inside an Autonomous System (AS) [58], and as such it belongs to the IGP family of routing protocols. OSPF provides routes by maintaining a link state table of the complete topology of the network and running the Dijkstra algorithm to compute the shortest route to a destination. The main metric defined in OSPF is the cost, which usually is the path minimum cost to the destination. All nodes have the same view of the network, hence OSPF provides a centralized but distributed routing mechanism throughout the network. The topology information, which is maintained by each node in a Link State Database (LSDB), must be identical for all nodes in order for the individual OSPF nodes to make coherent routing decisions. This requirement is especially crucial in the case of Open Shortest Path First - Traffic Engineering (OSPF-TE) [62] where information regarding link utilization is distributed and used for Traffic Engineering [63] purposes. Here, even just a temporary incoherence of link state or utilization information might cause suboptimal routing decisions or partial network overload.

An OSPF network can be decomposed into smaller networks, called routing areas. Area 0 is also called the backbone area and interconnects all other routing areas, yielding a two level routing hierarchy. The backbone area is the core of the network and provides connectivity between different areas. OSPF addresses heterogeneous network technologies by defining different types of links to interconnect OSPF nodes taking into account the different characteristics of the specific technologies being used. Among the defined link types, the following may be suitable for WMNs:

- The *Point to Point* link type is the standard link type used by OSPF. A bidirectional link is required for this link type to be used.

- The *Broadcast* link type describes a link to which several nodes are attached to. Each node has the capability to address a single broadcast message to all the routers connected to the link. In such a network all nodes sharing a link are assumed to be able to communicate directly and the HELO protocol takes benefit of the broadcast capabilities to reduce the amount of signalling. IEEE 802.11 or Ethernet are examples of such a link type, where the HELO protocol selects a Designated Router (DR) among the nodes sharing a link which plays a central coordinator role. The DR performs two main functions for the routing protocol: It generates a network-LSA (Link State Advertisement) on behalf of all the nodes on the broadcast link, which provides information about all nodes attached to that link. The DR is also in charge of distributing this information to the rest of the routing area. The DR becomes adjacent to the rest of the nodes in the network. Since all Link State databases are synchronized in the network, the DR plays a central coordinator role in the synchronization phase. The use of the broadcast link type poses some problems when used in a typical WMN environment, since it assumes fully bidirectional communication between all nodes sharing the same link. As described in Section 2.2, this can not readily be assumed in, for example, IEEE 802.11 *ad-hoc* networks. Therefore, the DR designation algorithm may cause partitioning and two or more DRs may be selected and, thus, routing will not converge.

- The *Point to Multipoint* link type is usually used in Non-broadcast Multiple Access (NBMA) mode. OSPF treats links configured as Point to Multipoint as a collection of Point to Point links connecting nodes attached to the same link, hence allowing multiple Point to Point OSPF links to be aggregated on one physical interface. In this case, no DR is designated and the network is not advertised with a single network-LSA as in the case of the broadcast link type. In this link mode, each router creates an adjacency with all other nodes attached to the network so the signalling required by this link type is higher than in broadcast links. The higher signalling overhead is considered a main problem especially for WiBACKs networks where each link variation or resource reservation would have to be shared with all other nodes in an area, see [64]. As with the Point-to-Point link type, the Point-to-Multipoint requires bidirectional links.

- The IETF is considering the MANET Designated Router (MDR) extension for OSPF in order to better support operation within MANETs [65]. The MDR extension is based on an adaptation of the broadcast link type to allow for only partial connectivity which can occur in MANETs due to, for example, the hidden node problem. While the MDR extension might increase the stability of OSPF in single-radio MANETs, it does not address the requirements of a WiBACK which would require hidden nodes to be eliminated by e.g. network reconfiguration.

OSPF is a very popular IGP. Numerous extensions have been defined to address operator demands such as IPv6 support, traffic engineering extensions or security extensions. On the contrary, standard OSPF is an IPv4 centric routing protocol and major modifications were required and introduced with OSPF version 3 to support IPv6.

A more generalized protocol is the OSI IS-IS (ISO/IEC 10589:2002 Second Edition) protocol, which operates at an Open Systems Interconnection (OSI) layer between the data link layer and network layer, which makes it independent from the network layer for its management signalling purposes. IS-IS is similar to OSPF in that it is also a link state protocol that belongs to the IGP family. It also relies on the Dijkstra algorithm to compute the shortest path to a destination and provides an extended support for routing areas. IS-IS is often run in operator networks together with MPLS. If used for Traffic Engineering it is impacted by the same coherency issues as OSPF-TE.

Both, OSPF and IS-IS perform a bidirectional link check and disable links if the test fails. Hence, neither protocol supports UDLs. The underlying Dijkstra algorithm, however, considers unidirectional connectivity for its *shortest path* computation. Hence, conceptually, link state protocols could consider UDLs, if implemented accordingly and as long as each node has at least one transmit and one receive interface [66]. Here, the authors show that UDLs can be supported in link state routing protocols as long as an *inclusive cycle* exists. Following this approach the WiBACK TMF would have to determine such *inclusive cycles* via neighboring WNs in order to provide bidirectional control connectivity for such a node. In [66], already established satellite links where considered as the main use case while the issues of channel assignment and neighbor discovery in multi-channel environments where not considered.

Due to their complete topology knowledge, link state protocols are capable of computing optimal routes supporting TE. Their distributed design, however, poses a major obstacle in networks with volatile link and QoS allocation states when the distributed protocol state can no longer converge fast enough to ensure coherent routing or resource allocation decisions, see [64].

2.3.3 IP Autoconfiguration

In the context of this work, we focus on the interfaces that are used for intra-WiBACK communication. The access interfaces towards the UTs are expected to be implemented via bidirectional technologies providing user access via the standard IPv6 protocol suite.

Auto-configuration of network interfaces for Internet Protocol (IPv4) or IPv6 requires bidirectional connectivity, since discovery packets are sent out and responses are expected on the same interface.

Additionally, the Address Resolution Protocol (ARP) protocol and the Dynamic Host Configuration Protocol (DHCP) protocol for IPv4, as well as the IPv6 stateless and stateful auto-configuration mechanisms such as Stateless Address Autoconfiguration (SAA), DAD or router and neighbor solicitations require link-local broadcast or multicast capabilities and expect to receive replies on the same interface.

To support UDTs, the relevant IPv4 parameters such as interface configurations, ARP entries of local neighbors and default gateways could also be configured manually or be provided via a specially tailored configuration mechanism. In IPv6 this is also still possible, but requires even more intrusion into mechanisms that have been designed for automated operation.

Using multiple interfaces in one node, e.g. one for receive-only and one send-only interface, might also introduce problems related to multi-homing, if the interfaces are located in different subnets. One possible approach to provide IPv4/IPv6 support for unidirectional interfaces has been proposed with the standardization of LLTM[53] within the IETF. The implications of such an approach for the WiBACK architecture have been discussed in Section 2.2.2.

2.4 Unidirectional Links in Wireless Networks

In the following we evaluate UDL support in popular MANET or WMN protocols. Among the numerous protocols and extensions that have been proposed, only a small subset has actually been implemented or reached Request for Comments (RFC) status. As discussed before, such mostly IP-based protocols are unaware of the actual underlying technologies and their specific characteristics. Moreover, channel assignment cannot be conceptually supported. Hence, such protocols simply run on top of the links resulting from the current radio configuration aiming at maintaining an optimal connectivity given the discovered topology. Most WMN protocols implicitly assume volatile link and even node states or assume an underlying IEEE 802.11 MAC layer in *ad-hoc* mode.

Another family of WMN or MANET protocols, that has been proposed, is located at the MAC layer of technologies such as IEEE 802.11. The most well-known and actually implemented and wide-available protocols is the IEEE 802.11s mesh extension which builds on top of a hybrid reactive and locally proactive approach.

As an example of the above, we discuss the rather simple reactive AODV, the more complex DSR, the pro-active Optimised Link State Routing (OLSR) as well as IEEE 802.11s. We do not review protocols such as batman [67], or OLSR extensions such as *fish-eye*, since non of those protocols address core WiBACK requirements such as spectrum or capacity management. Hence, we focus on highlighting the different concepts of addressing UDLs in order to evaluate if such mechanisms can be applicable to the WiBACK architecture.

2.4.1 AODV

AODV[50] is a reactive routing protocol that provides dynamic, self-starting, multi-hop routing between participating nodes. AODV allows nodes to quickly obtain routes for new destinations, while routes for inactive destinations do not need to be kept. AODV is inactive as long as both endpoints of a communication connection have valid routes to each other. AODV operates loop-free by using destination sequence numbers, and by avoiding the Bellman-Ford *counting to infinity* problem, provides quick convergence upon topology changes.

AODV defines three message types, Route Requests (RREQs), Route Replys (RREPs), and Route Errors (RERRs) which are used to communicate via User Datagram Protocol (UDP) with other nodes. When a route to a new destination is needed, the source node broadcasts a RREQ to find a route towards the destination. A route can be determined when the RREQ reaches either the destination itself, or an intermediate node with an up-to-date route to the destination. The propagation range of RREQs can be limited by setting the Time to live (TTL) in the IP header accordingly.

An up-to-date route is considered a valid route entry for the destination if the associated sequence number is greater or equal as the one contained in the RREQ. A route is made available by unicasting a RREP back to the origination of the RREQ. Each intermediate node maintains a route back to the originator of the request, so that the RREP can be unicast from the destination along a path to the originator.

When a link breakage in an active route is detected, a RERR message is generated indicating destinations and possible subnets which are no longer reachable via the broken link. The RERR is sent to all neighbors in the so-called *precursor list*, which contains the IP addresses of all neighbors that are likely to have active routes across the broken link.

AODV nodes may request ACKs when sending RREQs over links they suspect to be unidirectional or broken otherwise. AODV can not differentiate the cause, but would black-lists such links if no ACKs are received, therefore effectively ignoring native UDTs. This issue has been studied, for example, in [57]. The proposed modifications, such as *HELO-ACK*, may improve the UDL-detection of AODV and allow it to utilize UDLs, but might significantly increase the protocol overhead. Moreover, as described in the Section 2.2, the proposed mechanisms can not differentiate between a *flaky* IEEE 802.11 ad-hoc links or links provided by a natively unidirectional technology, such as DVB-T.

2.4.2 DSR

Popular MANET routing protocols such as AODV assume bidirectional links ignoring non-bidirectional links by, for example, *black-listing* them, so as to avoid a protocol malfunction. DSR[51] takes a more advanced approach towards unidirectional links in that it tries to overhear traffic from other nodes when possible and maintains a local routing cache. Through cache lookups, it could learn that a link is functional unidirectionally and utilize it in that direction. As described in Section 2.2, for wireless technologies such as IEEE 802.11 in *ad-hoc* mode this would mean that such links are operated on top of a MAC layer that is in a *flaky* and nondeterministic state and is therefore not suitable for a carrier-grade back-haul network.

2.4.3 OLSR

OLSR[68] is a proactive routing protocol, based on classical link state routing, optimized for use in WMNs. Unlike reactive routing protocols such as AODV, OLSR constructs routes in advance of any traffic forwarding request. Therefore, routes are instantly available and the forwarding delay is lower compared to reactive protocols. OLSR utilizes a technique called multi-point relaying to limit the scope of control message flooding thus yielding a scalable link state routing protocol. Each OLSR node performs two-hop neighbor discovery via a periodic exchange of HELO messages. Based on this information each node then selects a set of one-hop neighbors as Multipoint Relays (MPRs). MPRs are selected such that there is a bi-directional path to each of a node's 2-hop neighbors via a potential MPRs. Each node informs its neighbors about its MPR set in further HELO messages. Received HELO messages are parsed by each node to maintain information about the set of neighbors that have selected it as an MPR.

The basic forwarding rule in OLSR is that an MPR node only forwards a HELO message if it is the first time that it has received the message and if the previous hop has designated it has an MPR. In order to disseminate the neighbor information throughout the network, each MPR forwards Topology Control (TC) messages indicating the set of nodes for which it is designated as an MPR. By limiting the number of nodes that can rebroadcast control messages to a subset of neighbor nodes designated as MPRs, OLSR greatly reduces the number of transmissions required to flood a message to all nodes in the network. OLSR differs from other link state protocols by flooding only partial topology information, e.g. sets of MPRs. Each node uses the partial topology information to compute the shortest path to any other destination in the network. It should be noted that OLSR is beneficial mainly in dense wireless networks in which each node has a large number of neighbors and then only selects a subset of these to act as MPRs. In a sparse network, OLSR may select all neighbor nodes as MPR and therefore will perform like a classical link state protocol.

As specified in RFC 3626, OLSR performs a bidirectional link check and excludes UDLs from the set of possible links. To the best of our knowledge no extensions for OLSR have been proposed which would introduce UDL support.

2.4.4 MAC Layer Mesh Routing Protocols

Multiple MAC layer mesh routing protocols have been proposed, see [69] for an in-depth discussion. Being MAC layer protocols, the proposals are tailored for a specific technology and cannot readily be applied to the heterogeneous WiBACK architecture, since they may use technology specific addressing, message formats or make assumptions about he specific MAC protocol. Analogous to the IP-based variants, capacity allocations or hot-standby backup path are not considered. To the best of our knowledge, UDL issues have not widely been discussed. Since the proposed protocols are conceptually variants of the above discussed IP protocols, similar issues and solutions are to be expected.

IEEE 802.11s

The IEEE 802.11s [70] amendment introduces support for mesh networking by defining how IEEE 802.11 devices may create a mesh network that can be applied to either static or ad-hoc topologies. Contrary to the protocols discussed above, IEEE 802.11s provides a framework which can utilize different routing protocols at its core. Hybrid Wireless Mesh Protocol (HWMP) is designated as the default routing protocol, but numerous alternatives and QoS-considerations have been discussed [71, 72, 73, 74]. Being an IEEE 802.11 centric framework, it cannot readily be extended to support natively unidirectional technologies.

2.4.5 Multi-Radio WMNs

Research on MR-WMNs focuses mainly on routing metrics and optimization algorithms for IEEE 802.11-based systems in order to increase the network capacity by, for example, reducing inter-channel interference [75] or by avoiding busy channels and external interferences, see [76], [77] or [78]. Both centralized and decentralized schemes have been discussed and, for example, in [79] the impact of packet loss and queuing delays on QoS-support is considered, while, conceptually, per-path capacity allocations are not supported. In [80] an IEEE 802.11-specific multi-radio *Infrastructure*-mode approach has been implemented which forms a mesh rooted at a gateway node. This approach supports channel assignments to minimize interferences, but does not consider TxPower nor the coverage class optimization. Due to the operation in *Infrastructure*-mode this approach is not required to consider UDLs, since they would not be exposed.

To the best of our knowledge, no heterogeneous approach to facilitate topology forming and maintenance supporting channel assignments and link optimization for multi-radio multi-channel networks including UDTs has been proposed.

2.5 Cell and Link Ranges

In the WiBACK architecture, the Topology Management Function (TMF) may, via the AI, adjust the transmit-power level or Modulation and Coding Scheme (MCS) settings of any wireless interface that supports such configuration options. Hence, any radio that allows its transmit configuration to be altered in a wider range, may, logically, act as a micro, standard range or macro cell. The resulting range or coverage area and, therefore, the number of possible receivers may be an important criterion when computing optimized multicast *Trees*, but also plays an important role when optimizing point-to-point link configurations. Those considerations, which have been discussed in [10], are not particularly related to the integration of UDTs, but effect the efficient deployment of, for example, DVB-T2 cells. Considering our research questions, the aim was therefore to evaluate if the CARMEN Abstract Interface (AI) provides proper support for cell or link range determination in order for the TMF to determine an optimal configuration of the respective wireless technologies.

2.5.1 Cell Range Considerations

Macro or overlay cells have been studied in the literature for cellular networks where they might increase the system capacity[81, 82, 83], but also in the context of WMNs[84, 85, 86, 87], here mostly with the focus to break with the single-radio-per-node *ad-hoc* forwarding paradigm and its limitations regarding throughput and predictable QoS support. In the WiBACK architecture, bidirectional macro cells are natively supported and exposed to the spectrum and capacity management modules as regular links between WNs. The potential advantage of such macro cell-based links, namely the direct link-local connection between nodes, needs to be balanced against the lower bandwidth-per-area density compared to smaller mesh cells where SDMA and frequency re-use may be exploited, see Figure 2.8. Hence, smaller cells yield higher unicast throughput, while larger cells may reach larger groups of receivers with a single isochronous broadcast transmission. Such macro cells are therefore well suited for the distribution of multicast traffic or specific network management or synchronization tasks. Due to their minimal MAC overhead for broadcast transmissions, we propose to provide such macro cells via robust unidirectional broadcast technologies such as DVB-T2.

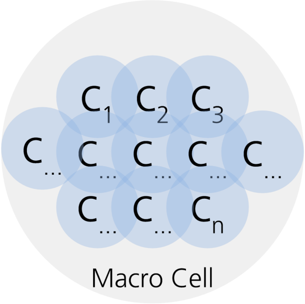

Figure 2.8: Compared to standard range or even micro cells, macro cells provide a higher range, but a lower dedicated bandwidth density

2.5.2 Multicast Tree Forming Considerations

The WiBACK architecture assumes that most multicast use cases can be addressed using 1-to-N *Trees*. Where multiple or mobile senders are required, they may be configured to send their datagrams via unicast to the multicast tree root, which would then reflect them back out into the tree. This approach may increase the delay for some receiving WN, but can easily be integrated into the WiBACK QoS management, mobility and forwarding schemes.

Depending on their configured cell range and the receiver distribution, macro cells may only partially cover the WNs forming a multicast *Tree*. Therefore further in-mesh multicast forwarding may be used to reach all receivers, see Figure 2.9. If supported by the underlying technology, the management modules may adjust the transmit power and MCS configuration to control the cell range and the resulting amount of required in-mesh forwarding. As depicted in Figure 2.10, a lower MCS yields a lower spectral efficiency, but

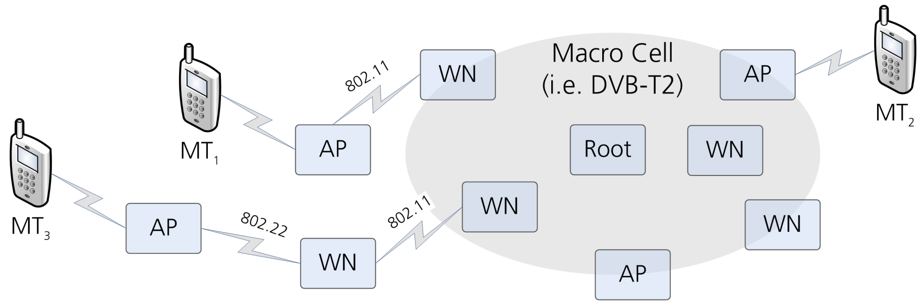

Figure 2.9: A macro cell partially covering a multicast tree of a WiBACK network

an increased cell range, and vice versa. Hence, in order to efficiently utilize longer range broadcast cells, the WiBACK capacity allocation algorithm must address the trade-off between in-mesh forwarding of traffic versus the range of broadcast cells. This becomes especially important for the distribution of multicast traffic with a larger number of receivers. Here, single-source 1-to-N multicast routing within a WiBACK network with macro cells may be configured in different ways depending on operator policies, receiver distribution and QoS requirements.

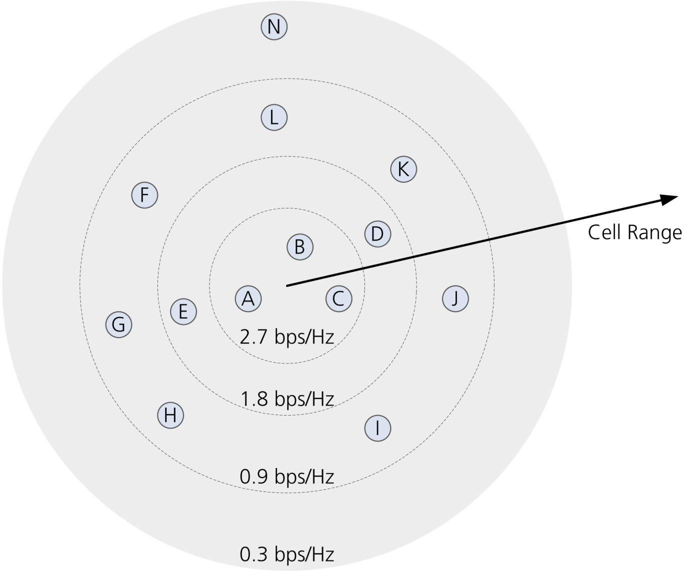

Figure 2.10: Effective spectral efficiency of an IEEE 802.11a cell depending on the MCS configuration and the resulting achievable range or coverage area

2.5.3 Multicast Tree Cost Calculation

In order for the capacity management module to calculate the optimal multicast forwarding tree, we propose a tree-wide cost function which combines the information provided by the wireless cell resource model, topology graph, the *Pipe* database as well as probing results from the potential receiving WNs. From the topology graph of logical links and the path database the topological receiving node distribution of a multicast tree can be determined. The result is a set of individual hop-by-hop forwarding trees, since in most cases multiple options will exist to form a tree covering all nodes. Based on the interface capacities reported by the AI, the algorithm can determine which destination WNs could be reached with a single link-local broadcast transmission. Additional probing would be required to determine which MCS must be used to reach the WN with the weakest link conditions in the group, where a lower MCS yields a lower spectral efficiency (E). To calculate the costs of the resources to be allocated for a *Pipe* segment, the number of receiving nodes (N) in the respective cell, the costs of its resources (C), e.g. bandwidth, and the scheduling or channel access overhead (O) are required. The latter varies heavily depending on the technology, its MAC layer design and the payload characteristics. For example, the IEEE 802.11 MAC is very inefficient when small (e.g. Voice-over-IP (VoIP)) datagrams are sent [88], since before sending each datagram the contention-based channel access procedure must be executed, which often requires more time than the actual datagram transmission. Hence, we propose to express the costs of *Pipe* resources allocated in a wireless cell C_{LSP} as:

$$C_{LSP} = \frac{C \cdot O}{E \cdot N}$$

C and O are constant for a given *Pipe* and its payload's characteristics. For example, for IEEE 802.11a, E has an effective lower bound of $E_{min} = 0.3$ bits/s/Hz and an effective upper bound of $E_{max} = 2.7$ bit/s/Hz. E may therefore vary depending on the receiving node distribution in the macro cell. Since E is bound and N may raise to ∞, C_{LSP} decreases reciprocally proportional to the number of nodes, for $N >> \frac{E_{max}}{E_{min}}$. If N is in the order of $\frac{E_{max}}{E_{min}}$, however, adding one distant node, such as node N in Figure 2.10, can significantly decrease E and thus increase C_{LSP}.

The above considerations need to be applied to each cell of the tree. The, the total costs of a tree C_{tree} can then be expressed as the sum of the costs of the n cells traversed:

$$C_{tree} = \sum_{i=1}^{n} C_{LSP_i}$$

Here, we assume that the individual per-cell cost consideration do not affect other links or cells which are part of the multicast tree or even the back-haul network as a whole. A simple multicast tree computation algorithm therefore has to consider two optimization criteria, meeting the end-to-end QoS requirements of the payload while minimizing the total costs C_{tree} of the resulting multicast tree. Since receivers may join or leave a tree over time, the tree topology may become suboptimal and a re-computation of the tree may free up mesh resources.

2.5.4 Discussion

While cell or link range considerations are not particularly related to UDTs, they play an important role when computing topology or multicast *Tree* optimizations. Due to unknown signal attenuation possibly introduced by cables or antennas or due to None Line of Sight

(NLOS) conditions, the theoretical range of a link or cell may differ significantly from the actual achievable range. Based on the current radio configurations, possible inter-WN connectivity can be determined via the AI_RadioBeaconScan or AI_RadioChannelScan primitives. To support cell or link range optimizations, an additional mechanism, such as the AI_RadioCalibrateLink primitive as proposed in Section 4.2.1, may be required. Moreover, such a primitive may need to consider multiple strategically chosen receivers, either in parallel or sequentially, in order to allow for an efficient calibration of a cell in the case of native Link layer multicast transmissions.

2.6 Monitoring in Heterogeneous Networks

In this section we evaluate existing approaches regarding monitoring in wireless networks focusing on support for UDTs and heterogeneity, where the main requirements of the WiBACK architecture are the detection or possibly prediction of link or *Pipe* failures as well as statistics gathering and neighborhood scanning. Since a transmitter on a wireless UDL has no means to detect reception problems on the receiving side, the monitoring system must be capable of detecting link or *Pipe* failures solely via passive monitoring on the receiving side.

2.6.1 Traffic Engineering

RFC 3272 [63] states that *"Traffic Engineering (TE) is concerned with performance optimization of operational networks with the goal to achieve efficient and reliable network operations while simultaneously optimizing network resource utilization"*. Additionally, RFC 3272 [63] suggests that the individual QoS assurances given to LSPs be monitored. The recently published RFC 6374 [89] specifies protocol mechanisms to measure and monitor packet loss, one-way or two-way latency, as well as delay variation and throughput in order to verify that, for example, Service Level Agreements (SLAs) among providers are met.

Hence, the TE-related tasks such as monitoring, measurement and statistics gathering as well as the analysis of the gathered data and possible event creation are also a crucial aspect of the WiBACK architecture.

TE relies on monitoring data for proactive off-line and reactive dynamic approaches, see [90]. The TE concepts relying on MPLS have mainly been developed for wired networks with orthogonal point-to-point links based on reliable technologies such as optical fiber or Ethernet. The breakage of such a medium can quickly be detected and fail-over times of about 50ms are typical in Synchronous Optical Networking (SONET) networks. Multiple schemes have been proposed to allow MPLS FRR [91] to provide similar fail-over times [92], while the standard RSVP HELO interval support a minimum of 1 second.

2.6.2 Monitoring in Wireless Networks

In the context of wireless networks, the definition of a broken link is rather fuzzy and very technology- or even implementation-dependent. Hence, additional parameters need to be considered in order to reliably evaluate link quality and reliability and in the literature a plethora of schemes have been studied, such as [93], [94] and [95]. Often, the proposed solutions combine measurement with analysis and event creation.

The wireless technologies currently considered in the WiBACK architecture range from satellite (i.e. DVB-S) over DVB-T to IEEE 802.16 and 802.11. Therefore, the nominal characteristics, as well as the parameters that can be analyzed may vary substantially. Moreover, due to the dynamic nature of wireless links caused by temporary fading or interferences as well as the often very dynamic per-frame transmitter configurations, the performance indicators to be evaluated must be carefully chosen.

Our work, as proposed in [11], provides a framework that measures and separates statistics gathering from analysis, interpretation and adaptable overall network state aware event creation. Additionally, multiple criterion functions, so-called *Rating Agents*, may be deployed interchangeably depending on specific technology or payload characteristics.

Due to this volatile nature of radio links and the tight QoS-requirements of, for example, VoIP and multimedia services the monitoring component of the WiBACK architecture is crucial to assess the overall network state. Hence, in [11], we have presented a modular passive multi-layer monitoring architecture addressing radio, link and well as LSP monitoring. The proposed architecture extends the CARMEN monitoring subsystem by LSP monitoring and, in particular, considers the monitoring of UDTs. We have shown that WiBACK monitoring can be implemented with a very low overhead using piggy-back injection of measurement headers to aid with delay and loss measurements while minimizing the transmission of additional monitoring frames, such as RSVP HELOs.

2.6.3 Statistics Export

IP Flow Information Export (IPFIX) is the successor of the NetFlow protocol developed by Cisco Systems and has been generalized to support statistics collection. The IPFIX protocol, as specified in RFC 5101 [96], specifies how IP Traffic Flow information gathered by so-called Metering Processes is transmitted across a network by so-called Exporting Processes to so-called Collecting Processes. To facilitate the export of IP Traffic Flow information, RFC 5101 specifies a common representation of flow data and standardized means of communicating such data. With RFC 6313 [97], support for hierarchical structured data and lists of Information Elements in data records was added. IPFIX considers network flows as unidirectional streams of packets which are identified by their source and destination IP addresses, the IP protocol and, optionally, the source and destination ports. IPFIX Metering processes may distinguish flow by their MPLS label, but does not readily support Link layer addresses, such as the WiBACK LinkId.

Conceptually, the IPFIX protocol could be adopted to support statistics exports within the WiBACK architecture, for example to push statistics data from the *Slave* nodes to their *Masters*, but seems too complex for intra-node statistics data exchange, which might be required to provide statistics data to analysis modules.

2.7 MPLS Protocol Suite

The Multi Protocol Label Switching (MPLS) or Generalized Multiprotocol Label Switching (GMPLS) protocol suite provides the required protocols and mechanisms to support Traffic Engineering (TE) and therefore constraint-based path computation, resource allocations and signalling as well as resource isolation among LSPs and forwarding along centrally computed paths. Since the WiBACK architecture adopts an MPLS-based data plane, in the following sections, we discuss possible issues regarding support for and integration of UDTs.

2.7.1 Multi Protocol Label Switching

Unlike hierarchical prefix routing which is typically associated with IP packet routing in the Internet, Multi Protocol Label Switching [98][99] tags packets with a 20 bit wide label based on which hop-by-hop forwarding is then performed in intermediate routers, see Figure 2.11. MPLS operates at an OSI layer which is often referred to as Layer 2.5, between the data link and the network layer and MPLS labels are typically positioned between link layer headers and the headers of the encapsulated packet; usually a layer 3 protocol header, see Figure 2.12. The name MPLS refers to its independence from both layer 2 and layer 3 protocols, which makes MPLS an interesting candidate for data forwarding in a MR-WMN that integrates heterogeneous technologies such as IEEE-based ones and those developed by 3GPP or DVB, as well future emerging technologies. MPLS paths are unidirectional, hence MPLS could be used on top of UDL without any need for modifications. Three bits in the MPLS header are designated for QoS prioritization and congestion notification which is especially important in a capacity-constrained network such as a WiBACK network.

Figure 2.11: MPLS label stack entry (MPLS shim header)

When a packet enters its ingress router, which is called Label Edge Router (LER) in MPLS terminology, the MPLS header is prepended to outgoing datagrams. Or, in MPLS terminology, a label is pushed onto the label stack of the packet. Downstream routers simply look up the first label in the stack of the packet in a table that maps labels to outgoing interfaces and therefore next hop routers. Forwarding is then performed based solely on the table entry referenced by the label. In order to avoid scalability limitations of globally unique labels, intermediate routers generally rewrite the label of a packet to

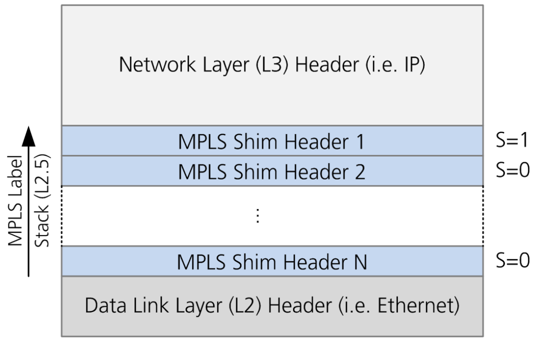

Figure 2.12: MPLS label stack embedded into the protocol stack headers

be forwarded with another label which has only local significance for the next hop. This exchange is referred to as label switching.

When a packet reaches its egress router in the MPLS network, the label is removed. Or, in MPLS terminology, the last label is popped from the label stack, and further packet handling must be performed based on the header information of the actual packet. MPLS can carry numerous traffic types, incl. IP, Asynchronous Transfer Mode (ATM), or IEEE 802.3(Ethernet)[100]. As many flows of packets might leave through a certain egress router, a mechanism called penultimate hop popping allows the second-last router to empty the label stack prematurely in order to relieve the egress router. The paths, which are given by the label-based forwarding states in all routers between ingress and egress routers, are called LSPs. LSPs are conceptually similar to virtual connections in ATM, although unlike ATM, MPLS supports variable packet lengths. To set up LSPs in IP networks, Constraint-based Routing Label Distribution Protocol (CR-LDP) or Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE) have been defined by the IETF. MPLS can be used to support and enforce Traffic Engineering decisions when forwarding data along PCE computed paths.

MPLS Fast Reroute (FRR)

MPLS Fast Reroute (FRR) [91] can provide protection against link breakage in a LSP by a backup path that circumvents the broken links. Each LSP can have a dedicated backup path or all LSPs sharing a network segment can also share the same backup path. Another possibility is to protect each individual link of an LSP, which requires backup paths between any two consecutive nodes, see Figure 2.13. To protect an individual node, a backup path from the upstream to the downstream neighbor of the to be protected node is required, see Figure 2.14. Maintaining backup paths introduces higher protocol overhead, but provides faster recovery in case of node or link failures when compared to higher layer protocol recovery, which is particularly useful for real-time traffic. In a capacity constrained WiBACK network, backup paths might block precious capacity needed for regular payload traffic and the trade-offs need to be considered carefully. Statistical backup capacity sharing among multiple backup LSPs might be used to reduce the amount of blocked but not utilized resources [101]. Another alternative could be to use the backup capacity for best-effort traffic.

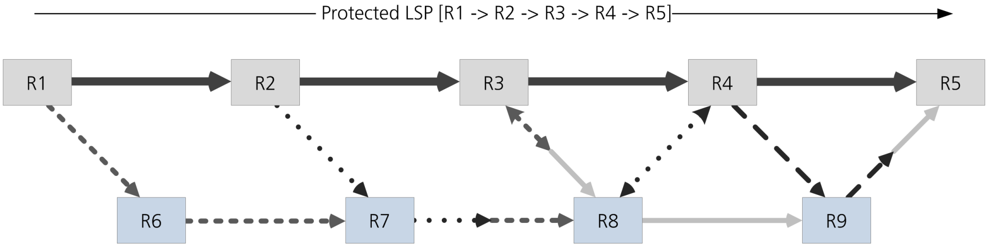

Figure 2.13: The MPLS one-to-one backup protects individual LSPs or segments thereof

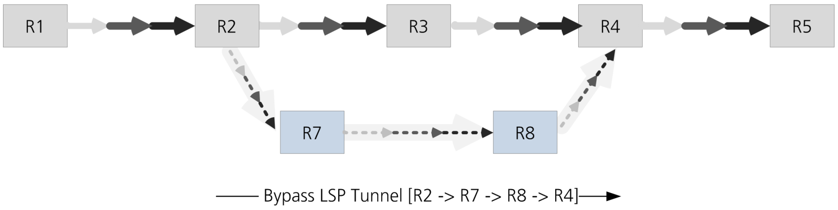

Figure 2.14: The MPLS facility backup protects a set of LSPs

MPLS Pseudowire

Ethernet *pseudowires* provide a virtual IEEE 802.3 Ethernet connection over an MPLS network by tunneling Ethernet PDUs via LSPs and allow service providers to offer *emulated* Ethernet services over MPLS networks. RFC 4448 [100] specifies the encapsulation of Ethernet/802.3 PDUs over a *pseudowire* as well as the procedures for using *pseudowires* to provide a 'point-to-point Ethernet' service. The *pseudowire* header, the so-called *control word*, mainly contains a 16bit-wide sequence number which can be used to ensure frame ordering or to detect loss. QoS tagging can be supported either by using the PRI field of the Virtual Local Area Network (VLAN) tag header of the encapsulated Ethernet frame, or by using the traffic class field of the MPLS header, see RFC 5462 [56].

2.7.2 CR-LDP

CR-LDP [102, 103] is an extension of the Label Distribution Protocol (LDP) as specified in RFC3036 [104], and extends LDP to allow for constraint-based routing. For example, LSPs can be set up based on explicit route constraints, QoS constraints, and other constraints such as MPLS-based Virtual Private Network (VPN) setup. As of February 2003, the IETF MPLS working group deprecated CR-LDP and decided to focus purely on RSVP-TE[105]. Hence, LDP is not considered further in the context of this work.

2.7.3 RSVP

RSVP [106] was initially designed to reserve resources inside an IP network to guarantee certain QoS for a particular flow. Later on, RSVP was extended to establish LSPs.

This extended protocol is called RSVP-TE [107] and is the preferred signalling protocol for LSPs. In order to reserve network resources for a given unidirectional flow, RSVP first discovers and sets up the path of the flow inside the network, and then reserves the resources along the RSVP-aware routers forming such path. PATH messages are employed for path setup, whereas RESV messages are employed for resource reservation. This split of the reservation mechanism in two phases allows RSVP to be applied to both unicast and multicast flows, so each receiver of a multicast group is able to reserve resources independently, according to their own requirements. This reservation mechanism for multicast flows is controlled by the receiver and the sender is not aware of the actual resource reservations performed by each receiver. While this mechanism is in-line with typical Internet-wide multicast session scenarios, it contradicts multicast resource planning in controlled and managed networks, such as the one provided by the WiBACK architecture.

To resolve errors and dynamic network changes both flow setup and resource reservation are handled by RSVP routers as soft-state, thus the source and the receiver must periodically send PATH and RESV messages to renew the flow state, otherwise the flow state will age until it is removed by all the intermediate RSVP routers. In order to quickly deallocate resources, RSVP also defines PATH Tear and RESV Tear messages. If an error occurs in the path setup or resource reservation process, RSVP routers may send PATH Error or RESV Error messages to notify the flow's source and destination respectively. Therefore if the topology or the available resources of the network change, the affected RSVP routers can quickly notify both ends of the flow.

To set up a flow, the source of the flow creates a PATH message identifying the data flow and its traffic characteristics. It then sends a PATH message towards the flow's destination IP address. All RSVP-aware routers in the flow's path intercept and process this PATH message, create path specific state and forward the message downstream until the PATH message arrives at the destination. At this point, flow state has been installed in the path, but no resources have been committed. To commit the requested resources, the receiver sends a RESV message towards the source which is forwarded backwards hop-by-hop along the same path until it reaches the source. After performing admission control and policing, each node commits the resource reservation for the flow. To support multicast flows, RSVP may merge several reservations sharing the same resources.

RSVP messages are transported directly on top of IP and may include the IP Router Alert Option in order to be processed by intermediate routers, while the RSVP message, i.e. a PATH message, is addressed to the destination node or multicast group. All RSVP messages have a common header, which specifies the message's type (e.g. PATH, RESV, Tear, Error), followed by several RSVP Objects encoded in Type-Length-Value (TLV) structures. Among the most relevant RSVP objects are the following:

- Sender Template and Session Class objects which specify the flow's source and destination (e.g. by IP address and port or LSP)
- Time Values object which specify the refresh interval of PATH and RESV messages

- Hop class which has different objects to notify the address of the previous or next RSVP router
- Adspec object that describes the intermediate hops that compose the path
- Flowspec object which specifies the characteristics of the flow as a simple Traffic Class or with more detailed Token Bucket parameters

Traffic Engineering Extension

The TE extension for RSVP introduce additional objects to support label distribution. In particular PATH messages may include a Label Request Object, whereas Label-Switched Routers (LSRs) add a Label object to the RESV message. To support constrained-based routing, the ingress router may use an Explicit Route Object (ERO) [108] to specify the full path of an LSP defining each hop of the path. Since an Explicit Route Object may include loose hops (e.g. not all the hops but just some intermediate points in the path), and in order to avoid loops, it is recommended to add a Record Route Object, so RSVP routers can check if they have already processed a RSVP message. Finally the Session Attribute Class allows defining more Traffic Engineering parameters, such as the priority of the LSP to solve the cases of LSP preemption. To correctly account for resource allocations in shared-medium subnets, RSVP employs a centralized admission controller, the so-called Subnetwork Bandwidth Manager (SBM). This way, when a RSVP router connected to the subnet receives a PATH message, instead of forwarding it to the next downstream router, it is handled by the SBM that fills the Hop Object with its own address. Accordingly, the RESV message will be addressed to the SBM and it is able to keep track of the subnet resources and accept or deny further reservations.

The RSVP-TE extension adds support for MPLS downstream-assigned unicast label distribution via the *PATH* and *RESV* messages. Support for multicast LSPs was added with RFC 5331[109] and RFC 5332[110] which introduced upstream-assigned labels. These consist of two MPLS labels, with the first label containing the node context, while the second label is interpreted as the actual label. The node context is used to allow a receiving node to distinguish among possibly identical labels assigned by different upstream node.

Acknowledgements

RSVP is a soft-state protocol and therefore relies on the periodic retransmission of messages to maintain its state and does not use Message IDs or sequence numbers nor does it provide an explicit *ACK* or *NACK* service. In [111], it was shown that RSVP and, in particular the *Graceful Restart* mechanism, perform poorly over links with higher loss probabilities, which are to be considered in the WiBACK context. Reliable messaging via Message IDs was introduced with RFC 2961[112] which allows for Acknowledgments as well retransmissions of lost messages on a per hop basis, which should significantly improve the protocol performance in the case of packet loss, see Figure 2.15.

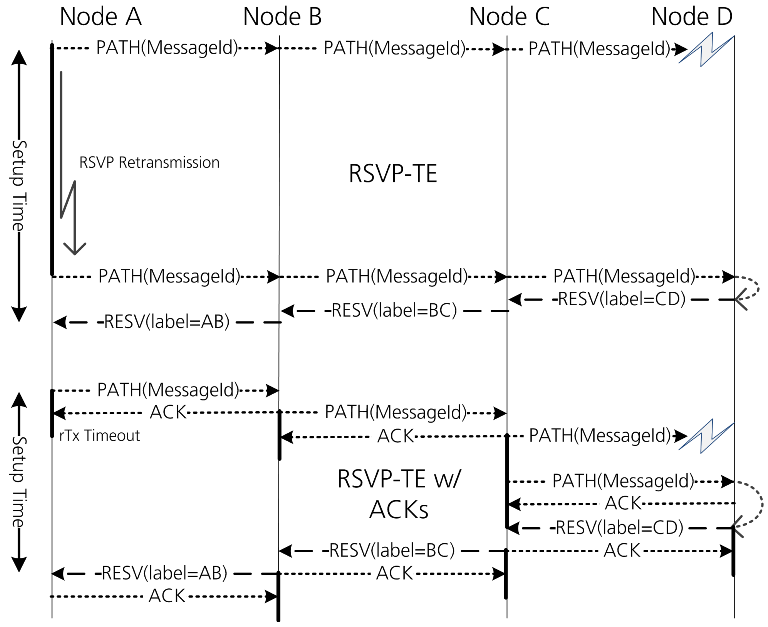

Figure 2.15: In case of packet loss, standard RSVP relies to periodic end-to-end refresh messages, while the RFC 2961 extensions allow for hop-by-hop ACKs and explicit retransmissions

The WiBACK IMF supports reliable messaging through the use of transaction identifiers and the optional *AckService* component which can be deployed on a per-transaction basis, for example, when communicating over an unreliable transport. The underlying IEEE 802.21 architecture does not support, however, that messages are intercepted and processed by intermediate nodes, hence an RSVP-TE-like mechanism would require the use of nested hop-by-hop Request/Response transactions in order to realize a conceptually similar signalling mechanism. This issue will be described in detail in Section 4.4.1.

RSVP messages are typically forwarded via regular IP routing. In order to support TE, forwarding along pre-computed paths can be enforced using the Explicit Route Object (ERO) which describes the hop-by-hop route of RSVP messages. Hops can be specified as IPv4 or IPv6 addresses, while support for unnumbered links was added with RFC 3477[107] which allows for signaling over links without IP addresses in combination with the ERO. Since RSVP assumes bi-directional links, the *RESV* message would be sent back by reversing the path described in the ERO. Hence, this mechanism can not readily be applied in the presence UDTs, see Figure 2.16, where RSVP-TE would attempt to send a RESV message directly from node C to node B, since it is not aware of the underlying UDT.

RSVP describes individual flow QoS resources via rather flexible *FlowSpec* objects, while WiBACK describes its *Pipes* as flow aggregates via *TrafficSpecifications* specifying the QoS resources in terms of bandwidth, maximum latency and maximum loss as well as the *TrafficClass*. LSP payload type signalling is out of scope for RSVP, while in the WiBACK architecture it is crucial to, at least, differentiate between *Management Pipes* and *Data Pipes*.

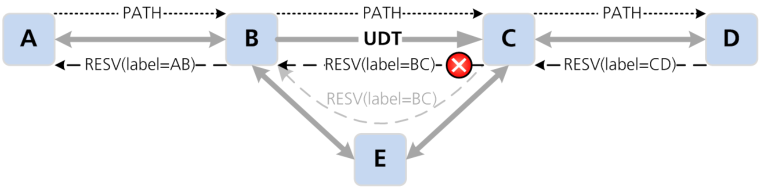

Figure 2.16: The ERO allows RSVP-TE to signal along a predetermined path. The return path is determined by reversing the path described by the ERO. This mechanism does not work in the presence of UDTs where an alternate path around the UDT would have to be used.

2.7.4 PCE

In order to perform Traffic Engineering and support optimized constraint-based path computation the Path Computation Element (PCE) concept has been introduced [113] where a set of centralized PCEs maintain the complete link and resource state of the routing area they control. If a new path is to be configured a request is sent to the PCE which may provide a set of possible paths meeting the requested QoS requirements. Forwarding along the provided path is usually enforced by setting up an MPLS LSP while RSVP-TE is used as the signalling protocol. In parallel to PCEs, either OSPF or IS-IS are operational to facilitate the proper routing of the network's signalling and management traffic while the actual user payload is forwarded along the PCE computed LSPs. This implies that additional network-wide routing state must be maintained in order to support the communication between the PCEs and the WNs as well as the RSVP-TE signalling. The issues and possible solutions to support UDLs in link state routing protocols have been described in Section 2.3.2, while UDL support for RSVP-TE has been discussed in the previous section.

The PCE architecture allows different approaches to implement PCEs by specifying a centralized or a decentralized operation. Furthermore, a PCE may be stateful and perform book-keeping on allocated resource or stateless and rely, for example, on the RSVP resource broker to eventually deny an LSP set up due to insufficient resources. RFC 5440 [114] defines the Path Computation Element Protocol (PCEP) which must be used by Path Computation Clients (PCCs) to request a path computation, but is also used among PCEs to accomplish, for example, the computation of a path crossing domains controlled by multiple PCEs.

The PCE concept has been adopted by the CARMEN project and the WiBACK CMF is designed as a centralized stateful PCE. TE considers links as unidirectional resources, where a bi-directional link is represented as a pair of unidirectional links. Hence, adopting those well-established TE principals, the CMF can readily describe UDTs and perform resource allocations. The relevant messages of the PCEP are mapped onto IEEE 802.21 style messages.

2.8 Summary

Summarizing this review of the literature we can state that support for UDTs has not been widely discussed by the networking research community. This is mainly due to the fact, that UDTs are very uncommon in contemporary IP networks. Issues involving UDTs have mainly been studied in two areas, namely satellite-based networks or wireless *ad-hoc* networks. In the first case, UDTs are exposed due to the implementation of many satellite systems and, for example, the Link Layer Tunneling Mechanism (LLTM) has been proposed to integrate UDTs into *best effort* IP networks. In the latter case, bidirectional wireless technologies in *ad-hoc* mode may effectively expose unidirectional behavior when being operated in the so-called *transitional region*. All well-known MANET or WMN routing protocols can detect such unidirectional connectivity and depending on the protocols either avoid such links or try to utilize them to increase the overall mesh connectivity.

We have shown that IP-based protocols can not distinguish between the two cases of unidirectionality and therefore might either avoid native UDTs or utilize malfunctioning or flaky connectivity provided by *ad-hoc* technologies operated within the *transitional region*, which might be acceptable for a *best effort* MANET or WSN, but poses serious issues for a carrier-grade network due to the unpredictable behavior of such connectivity.

MAC layer WMN protocols have only been proposed for specific bidirectional technologies, such as IEEE 802.11, and are therefore not required to support native unidirectional technologies. Moreover, to the best of our knowledge, no standardized generic interface exists such that, i.e. WMN protocols could query or modify the configuration of the underlying wireless technology.

With regard to our identified research questions, we have identified the following issues which, to the best of our knowledge, have not been addressed previously:

- **How to identify and describe UDTs?**
  The CARMEN project has introduced the Abstract Interface (AI), which offers a common interface to configure heterogeneous wireless interfaces. Through the AI, the topology and capacity management protocols can query interface capabilities, and request or set the actual interface configuration. This interface description could be extended to include an indication about the directionality of an interface which could then be exploited by other components.

- **How to perform monitoring on UDTs?**
  The monitoring concepts utilized by existing IP-based WMN routing protocols can not consider PHY or MAC layer information. Furthermore, most protocols combine measurement and analysis in one monolithic block and often rely on bidirectional HELO message exchanges which cannot readily be applied to UDTs. Monitoring aspects studied within the MPLS context may be applicable, possibly requiring adaptations to the wireless domain and UDTs, in particular.

- **How to signal paths in the presence of UDTs?**
  RSVP-TE can be considered as the de-facto standard protocol to signal LSPs in MPLS networks. While RSVP-TE supports explicitly source routed signalling to indicate the path of the LSP to be set up, it assumes bidirectional connectivity for each link in the path and can therefore not readily support UDTs. Moreover, RSVP-TE expects an IP routing protocol to be operational. Hence, applying the RSVP-TE concept to the IEEE 802.21 -based WiBACK control plane might require larger modifications, in particular when considering support for UDTs and signalling without IP routing state or without uniquely assigned IP addresses.

- **How to include UDTs into Topology Management?**
  Architectures to support self-managed forming and maintenance of heterogeneous wireless networks have mainly been discussed for IP-based MR-WMNs, where mostly IEEE 802.11 technologies are assumed. Hence, support for natively unidirectional technologies has, to the best of our knowledge, not been considered. The WiBACK TMF, based on the CARMEN ScF and our extended Abstract Interface (AI), can be seen as providing a novel approach in this field. The AI as well as the ScF protocol will require enhancements to detect, describe and utilize UDTs.

Concluding the above, we can state that support for native UDTs in meshed multi-radio wireless networks and the integration into the WiBACK architecture, in particular, require novel approaches beyond the state of the art.

2.9 Tentative Design

Based on the *problem awareness* developed in Chapter 1 and the *suggestions* derived from the literature review, we present our tentative design integrating UDTs into the WiBACK architecture, which can be broken down into five aspects. Figure 2.17 depicts the architectural role of our contributions.

Addressing the above, we will:

- extend the Abstract Interface (AI) to properly describe and handle UDTs, which may require the addition of new primitives or the modification existing primitives.
- design the Technology Independent Monitoring (TIM) component to support passive UDT-aware receiver-side link and LSP monitoring.
- integrate explicit routing via the *LinkVector* extension into the *Transport Service.*
- design the hard-state Pipe Management Protocol (PMP) as an IMF *User Module* to provide RSVP-TE-style path signalling *around* UDTs building upon the *Transport Service.* This will include the design of an adjustable reliability mechanism in order to support reliable signalling under varying loss and latency conditions.

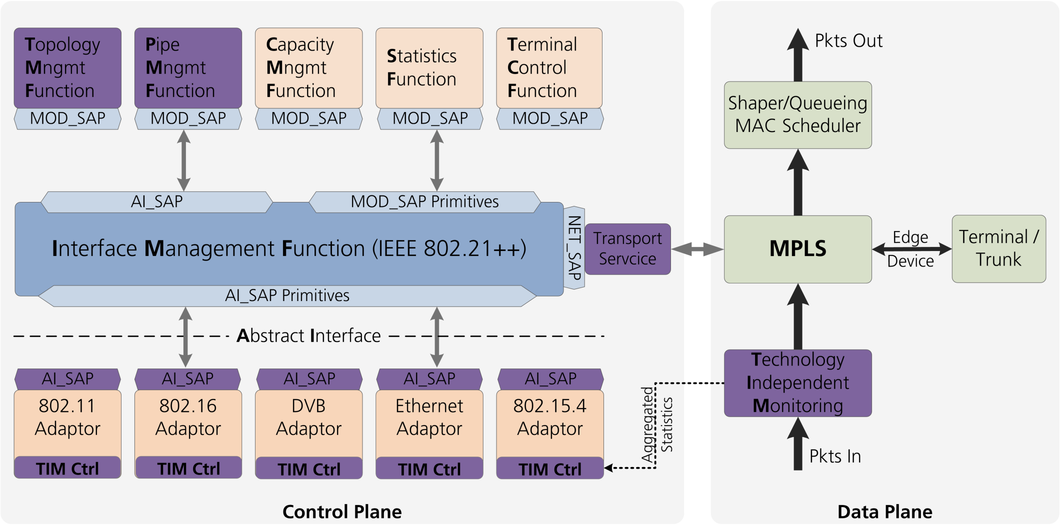

Figure 2.17: The dark purple items of the WiBACK node architecture represent the contributions of this study

- design the Topology Management Function (TMF) as an IMF *User Module* extending the CARMEN ScF with support for UDTs. Additionally, the level of possible UDT involvement during the topology forming phase will be investigated.

We will produce distinctive artifacts to evaluate the TIM component as well as the PMP with its adjustable reliability mechanism and the TMF. With regard to the AI extensions as well as the *LinkVector* extension, we expect to validate them via individual unit tests and implicitly during the artifact evaluations, since they rely on this core functionality.

In the following chapter, we describe and justify our methodology followed to design the required protocols and mechanisms integrating UDTs into the WiBACK architecture.

Chapter 3

Methodology and Design

In this chapter we will describe and justify the methodologies followed throughout this study to achieve our objectives and to address the derived research questions. In Chapter 1, we have have identified our or main objective *'To describe and integrate UDTs into the WiBACK architecture so that spectrum or capacity optimization algorithms can consider and utilize them'*. Based on this main objective we have identified open issues with regard to UDT support in the WiBACK architecture from which we have derived our research questions.

First we introduce quantitative research and the *positivistic* research perspective which assumes that numerical data exposes the underlying state of the object under investigation. This view is shared by engineers and many researchers in natural and applied sciences [22]. We then compare this view to *Design Science Research* (DSR), the research methodology followed throughout this work.

3.1 Quantitative Research and Positivism

Quantitative research is widely used in social sciences, while research in natural sciences such as physics is, by definition, quantitative. Qualitative methods, on the other hand, produce information only on the particular cases studied, while any further conclusions are only hypotheses. Quantitative methods can be used to verify which of such hypotheses are true. The process of measurement is central to quantitative research because it provides the fundamental connection between empirical observation and mathematical expression of quantitative relationships, see, for example, [115] for a detailed discussion.

Quantitative investigation of the world has been performed since people first began to record events or counting objects. The modern concepts of quantitative processes have their roots in Auguste Comte's positivist framework. Positivism emphasizes the use of scientific methods through observation to empirically test hypotheses explaining and predicting the occurrence of phenomena. Positivist scholars believed that only scientific methods rather than previous spiritual explanations for human behavior could advance science [116].

Positivism assumes that there is a single objective reality which is separate from our knowledge of it (separation of subject and object). The positivist position is grounded in the theoretical belief that there is an objective reality that can be known to the researcher, if the correct methods are used and applied in a correct manner [117, 118]. Table 3.1 lists the philosophical assumptions of the *positivistic* research perspective and also compares them to the underlying assumptions of the *Design Science Research* perspective [22]:

Basic Belief	*Positivist*	*Design*
Ontology	single reality, knowable, probabilistic	multiple, contextually situated alternative world states, socio-technologically enabled
Epistemology	objective, dispassionate detached observer of truth	knowing through making, objectively constrained construction within a context, iterative circumscription reveals meaning
Methodology	observation, quantitative, statistical	developmental, measure artificial impacts on the composite system
Axiology	Truth: universal and beautiful, prediction	control, creation, progress (improvement), understanding

Table 3.1: Philosophical assumptions of the different research perspectives [22]

3.2 Design Science Research

Design science research (DSR), a derivative of Design Research (DR), is a methodology often used in the field of information technology and here most notably in the Engineering and Computer Science disciplines. It focuses on the outcome and offers guidelines for evaluation as well as iteration throughout a research project. DSR emphasizes the development of *designed* artifacts and their performance. DSR explicitly intends to improve the functional performance of an artifact, where the artifact may be, for example, the protocol or algorithm under investigation. A more elaborate discussion of DSR and its relation to the *Positivistic* or *Interpretive* research perspectives can be found in [22].

In [119] the authors have presented a set of guidelines for DSR within the discipline of Information Systems. Design science research requires the creation of an innovative, purposeful artifact for a special problem domain. The artifact must be evaluated in order to ensure its utility for the specified problem. In order to form a novel research contribution, the artifact must either solve a problem that has not yet been solved, or provide a more effective solution. Then, based on the newly gained knowledge, the theory would have been enriched. Both the construction and evaluation of the artifact must be performed rigorously and the process clearly documented to allow for proper peer reviews.

The general Design Science Research methodology, as depicted in Figure 3.1, consists of five stages:

- Awareness of Problem - A problem has been encountered and is typically expressed in

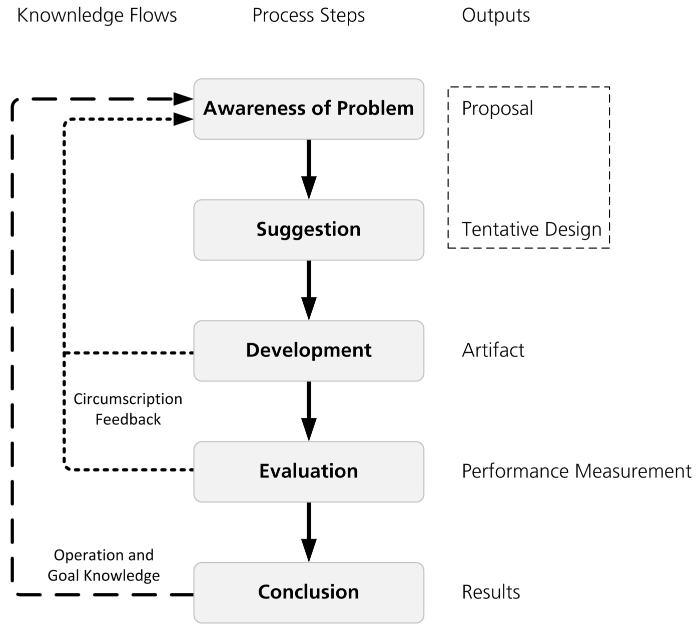

Figure 3.1: The general methodology of Design Science Research [22]

the form of a *Research Question*, such as 'How to integrate UDTs into the WiBACK architecture?'

- Suggestion - Suggestions to address the research question are collected yielding the tentative artifact design
- Development - The artifact, i.e. a protocol or algorithm, is implemented according to the tentative design
- Evaluation - The artifact is first validated and tested for suitability. Then the functional performance is evaluated in order to assess if and to what degree the designed solution addresses the research question
- Conclusion - Evaluation results are critically analyzed and a conclusion is drawn.

A crucial aspect of the DSR methodology are *knowledge flows* in the form of *Circumscriptions* where new findings discovered during the development, evaluation or conclusion stages are fed back into the problem awareness stage, typically yielding a more refined or precise tentative design. Hence, this mechanism generates knowledge. The evaluation phase also contains an analytic sub-phase in which hypotheses are made about the behavior of the artifact. Such initial hypotheses concerning behavior of the artifact are often revised and the evaluation phase results and additional information gained in the construction and running of the artifact are brought together and fed back into the Suggestion phase. This feedback process may be repeated multiple times, until the results can be considered *good enough*, that is the artifact behavior deviates only slightly from the hypothetical predictions [22]. At this point, the results are either considered *firm* and are summarized in a report or paper, or they may be considered as *loose ends*, possibly considered as subject of further research.

The DSR methodology has been adopted for this work, since it seemed to best match the task of this study, which focuses on the design of network architecture extensions to include UDTs. We have presented the tentative design in Chapter 2. In each case, the knowledge gained through feedback during the artifact development and evaluation stages has improved the final design, as presented in Chapter 4, significantly. After some initial iterations, intermediate results had been concluded and where published and discussed at scientific conferences [11, 7, 6]. The feedback received during those discussions as well as the experiences gained during the development and evaluation stages have been fed back, thus eventually yielding the final design and results as presented in this thesis and in articles submitted to relevant journals in the field of wireless networking [1, 2].

3.3 Overall Methodology and CARMEN

The majority of the work leading to this thesis has been performed within the EU FP7 CARMEN project, where broadcast technologies were to be integrated into the to be developed wireless back-haul architecture in order to support the distribution of broadband multimedia content. The initial problem awareness developed already during the proposal phase and initial suggestions have been made. The design went under numerous refinements either due to overall project design decisions or due to specific UDT-related issues encountered during the development and evaluation stages.

In an early stage of this work we realized that our methodology should consider two levels of detail, a more high level, overall theme addressing the main objective *'To describe and integrate UDTs into the WiBACK architecture so that spectrum or capacity optimization algorithms can consider and utilize them'* and more detailed subtasks addressing the individual research questions derived from the overall objective:

- How to identify and describe UDTs?
- How to perform monitoring on UDTs?
- How to signal paths in the presence of UDTs?
- How to include UDTs into Topology Management?

Our work focuses on the aspects regarding UDT integration into the wireless back-haul architecture as it is being specified by the CARMEN project. While the CARMEN project discusses and designs the overall architecture, our emphasis is on UDT integration during both, the discussion and the design phase of the project.

At a very early stage in the CARMEN project we have prepared the foundation for UDT integration by supporting the TE-based approach of treating network resources as logically unidirectional. Moreover, during the requirements phase, UDTs have been considered when connectivity among nodes was described.

Following the functional requirements phase of the project, there was a long and elaborate debate on the overall management paradigm. Should a distributed approach similar to OSPF-TE be followed or a rather centralized master/slave approach, more in line with typical operator network planning and management? While the CARMEN project has mainly considered scalability issues due to the expected link resource volatility [64], we have taken a different viewpoint in this discussion via a study [3] considering at which level in the protocol stack the proposed architecture would operate and what implications this would pose for the integration of UDTs.

While both, an integration at or below the network layer, was found to be possible, our study concluded that UDTs could more readily be supported in a centralized approach and that this would then rather operate below the network layer, for example, leveraging the MPLS protocol suite, which is considered a layer 2.5 technology. Eventually, the design decision was taken to base the CARMEN architecture upon a centralized master/slave management paradigm. This decision was adopted for the WiBACK architecture and at this point we broke down the overall research objective into separate subproblems which yielded the above listed research questions.

3.4 Research Questions

For each research question we analyzed the requirements and compared them against the state-of-the-art in the respective field of research. Where possible, we have aimed for an application of existing concepts to the artifact design.

3.4.1 How to identify and describe UDTs?

To address this research question, we will analyze the related aspects of the WiBACK architecture, such as the Abstract Interface (AI) taking into account requirements from modules such as the TMF or the PMF. We will propose extensions to the AI. The conceptual output of this question, the suggestions, will be integrated into the reference implementation of the AI and the control plane, which could be seen as the artifact regarding this research question. We expect feedback from the development stage, while the evaluation may mainly consist of a functional validation concluding that adding UDT-awareness has not broken the overall design.

3.4.2 How to perform monitoring over UDTs?

We will analyze the capabilities of Tx-only and Rx-only interfaces with regard to interface, link or LSP monitoring and compare them against the requirements of the WiBACK monitoring subsystem. Where necessary we will suggest possible extensions or modifications to the WiBACK architecture to support monitoring on UDTs. The proposed design will lead to an artifact development and an evaluation phase. The conclusions should be fed back into the WiBACK architecture.

3.4.3 How to signal paths in the presence of UDTs?

Addressing this question,we will research existing concepts and compare them to the requirements of the WiBACK architecture taking into account the centralized management approach as well as the 'below the network layer' signalling decision. We will then investigate and suggest how a suitable signalling mechanism can be realized based on the WiBACK control plane. The tentative design of the protocol will then be developed into an artifact and throughly evaluated under varying loss and latency patterns. We expect multiple iterative steps until a final and stable design has been accomplished.

3.4.4 How to include UDTs into Topology Management?

We will take the findings of the first two question and study possible UDT involvement during the topology forming phase as well as UDT support by AI primitives and general TMF mechanisms. The suggested extensions or modifications will then lead to an artifact whose performance can be evaluated under varying scenarios. We expect rather numerous iterations to incorporate feedback from the development and evaluation stages, since the conceptual level of UDT involvement may vary. Concluding, we will discuss which level of UDT involvement during the topology forming phase should be adopted by the overall WiBACK architecture and how UDTs can be supported by the TMF.

3.5 Quantitative Evaluation

Network protocol evaluation is a rather difficult task and depending on the protocols, their complexity, the size of the scenario and the evaluated aspects, no optimal solutions exist [120]. Where feasible, analytical methods may be used. If the problem is too complex to be solved analytically, such as networking protocols, simulations or emulations are usually relied upon to provide numerical results. Performed properly, such evaluations of an artifact provide quantitative assessments.

Moreover, certain simulation and especially emulation environments allow for the injection of real-time traffic into the system which allows well-established external measurement tools to be deployed in the validation or evaluation process. By also enabling the injection of recorded traffic patterns, such simulators or emulators are particularly useful in allowing the network designers to test new networking protocols or to change existing protocols in a controlled and reproducible manner [121].

Both, simulation and emulation, rely on models to describe the respective physical entities, such as wireless channels. The quality and proper application of such models may significantly impact the results, see [122] for an in-depth discussion. Additionally, in an emulation, the system is executing along a continuous time axis and computing overheads or other latencies of the emulator are exhibited to the evaluated protocol.

A simulation, on the other hand, executes along a non-continuous time axes, where idle periods of the system may simply be compacted and the time advances to the next pending future event. This may significantly speed up the simulation sessions. Likewise, if the computation of a model, e.g. a wireless channel, can not be performed in real-time, this overhead can be hidden within the non-continuous time axes and would therefore not affect the protocol evaluation. However, under such assumptions, real world traffic can not be considered by the simulator.

3.5.1 Evaluation Framework Alternatives

Simulation frameworks like ns-2, ns-3 or OPNET have become a frequently used method to evaluate protocols or algorithms. Since they can provide surrogates for real networks, simulations are commonly used when developing prototypes of new protocols and mechanisms [123, 124]. However, creating reliable and credible simulations is not an easy task, which requires a lot of additional effort since an adequate model and an appropriate analysis of the simulation output data is essential [125]. Besides that, depending on the simulation tool, running the same code as the real-word application without any modification might be not possible. This is a crucial issue for our WiBACK evaluation since, on one hand, we require deterministic scenarios to evaluate our protocols in terms of performance and for scalability. Here we do not evaluate the exact impact of technology specific MAC and PHY layers, but rather verify the scalability of the higher layer WiBACK protocols. On the other hand, in order to evaluate, for example, the QoS-assurance provide to a *Pipe*, we rely on real hardware radios, or a combination of emulated and real hardware nodes. The latter allows us to create larger scenarios, where only the data path to be evaluated must consist of real nodes. Hence, a real-time emulation seems more suitable for our requirements than a simulation tool.

The above considerations concern the evaluation concept of the overall WiBACK architecture, which is to be extended to support and integrate UDTs. To solely validate and evaluate the UDT-related aspects, simulations could have been relied on, as well.

Multiple toolkits are available which allow for creating executable program code as well as the integration of such code into an emulation or simulation environment. The *Click* framework [126] seemed to be the most promising candidate for our use-case. It provides a modular, fine-grained software architecture which allows for creating flexible and configurable routers. A *Click* router is build by connecting packet processing modules called elements which implement simple functions such as queuing, classification of packets or just interfacing with a network device. In order to assemble a router a user needs to connect a collection of elements into a directed graph where the edges are called connections and represent the possible paths between elements. To extend the functionality, new elements can be implemented or existing elements can be composed in new ways. The main organizational principle of *Click* is a packet flow and, thus, fine-grained elements responsible for simple tasks are generally preferred over coarse-grained elements allowing more complex features. This, however, might be appropriate for rather simple protocols such as regular IP routing but more complex and stateful problems might not be easily divisible to fit into this packet flow model.

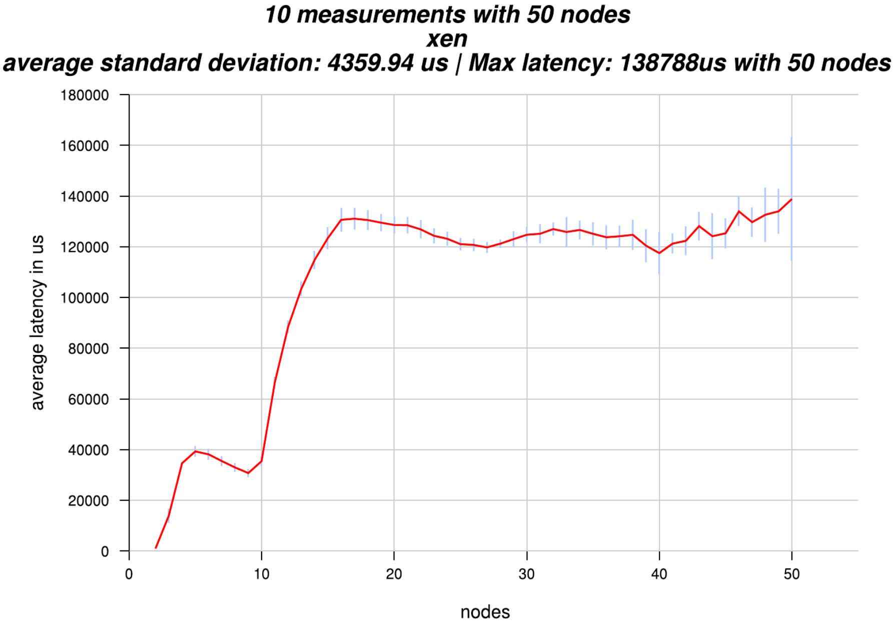

Figure 3.2: Local multicast packet reception latency over number of receiving nodes emulating the reception of a frame within a wireless cell

Another commonly used approach is to use virtualization techniques in order to emulate nodes as well as the connectivity among them, see, for example, [127]. Since Virtual Machines (VMs) usually emulate complete PCs the footprint in terms of CPU load, required RAM, etc. is relatively large. Additionally, VMs may introduce considerable latencies when emulating the transmission of a frame within an emulated wireless cell, where, in reality, a frame sent by a sender is received by receiving nodes at almost the same time. Depending on the aspects to be analyzed, larger latencies could introduce significant inaccuracies rendering the obtained results useless. Particularly if protocols with strict timing requirements are considered this becomes a critical issue. In our WiBACK architecture we have various timeouts in the order of some 100ms. Hence, in order to keep the potential emulation-introduced inaccuracy below 1%, the frame delivery latency and especially its variance should not exceed a low millisecond figure. Figure 3.2[1], depicts the latency for a Linux/XEN-based scenario on an eight-core Xeon machine and shows that the latency increases from roughly 20ms for a small scenario to about 140ms for a medium sized 50-node scenario, which is one or two orders of magnitude higher than our targeted maximum latency. Ongoing research on this topic confirms our findings [128].

---

[1] Those measurements were performed by A.Gillert at Fraunhofer FOKUS as part of his Diploma thesis which provided a more detailed insight regarding emulations on Linux-based system and investigated how the Linux kernel could be better tuned for such tasks.

For a different use case, the SelfNet project [129] used a VirtualBox[2]-based approach. Since the requirement was to only emulate about 10 nodes and latency requirements were not so strict, this seemed like an easier solution. During the course of the project it was discovered that the CPU requirements would not allow the researchers to run the foreseen scenario in real-time, even on relatively fast multi-core machines. After converting the implementation to a NetEMU-based approach, as described in the following section, the CPU requirements dropped by about 80%, see Figure 3.3. Note that this comparison was performed on a 2.8 GHz quad-core system, hence the maximum CPU utilization reported by *top* would be 400%. During the measurements we utilized the SelfNet Dynamic Protocol Composition Framework (DPCF), with a simple forwarding rule set. Each node had to process five hundred packets per second with a payload of 1400 Bytes each.

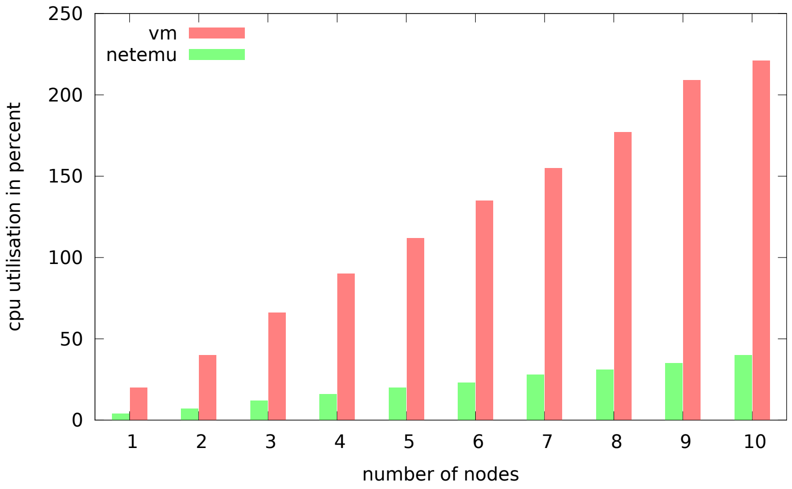

Figure 3.3: Comparison between VMs and NetEMU regarding the CPU utilization

Since neither the simulation approach nor the currently available emulation and virtualization based frameworks meet the requirements towards our WiBACK development framework, we have developed our own emulation and development framework, called NetEMU, which allows us to evaluate the same source code in emulation or on embedded hardware. Where possible, both emulation and testbed measurements will be performed to triangulate the results, thus improving their statistical significance as well as validating their applicability to real-world scenarios.

3.5.2 NetEMU Emulator Considerations

The artifacts built for evaluation will be based upon our SENF-based[3] low-latency NetEMU framework [9] which also provides a real-time network emulator component. This framework allows us to evaluate the same binary code on emulated or real embedded nodes. For emulated interfaces random packet loss and a fixed link latency can be introduced and the

---

[2] http://www.virtualbox.org

[3] http://senf.berlios.de

transmission range of an interface is emulated considering the frequency-dependent free space loss as well as a varying transitional zone.

Since the real-time emulations are executed on a multi-core Linux host, the operating system introduces slight scheduling latencies as well as variances thereof. In a 60-node scenario, those latencies and the emulation overhead have been shown to be less than 1 ms [9] per emulated link, which is a typical latency for loaded IEEE 802.11 links. It therefore depends on the requirements of each measurement if such latencies or variances are rather beneficial since they introduce real-world aspects or if they introduce extra uncertainties potentially rendering an evaluation useless.

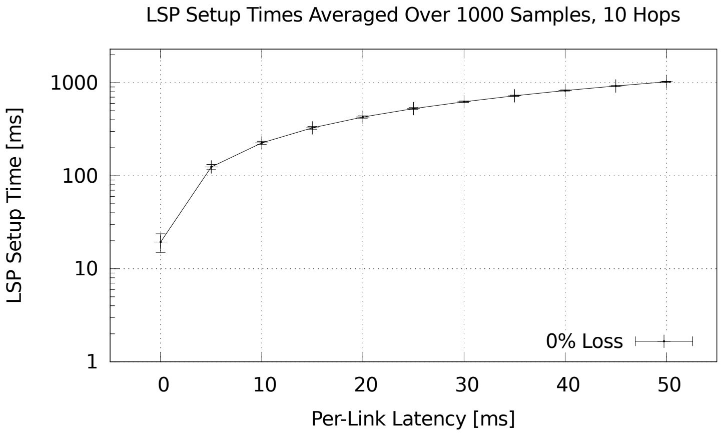

Figure 3.4: Example of a measurement set exposing emulator and protocol computation overhead as well as variances thereof

An example for the random latency variances introduced by the emulator environment as well as the rather fixed protocol computation overhead is shown in Figure 3.4, which depicts the path signalling times in a 11-node IEEE 802.11-based WiBACK scenario where. In Chapter 5, the results of those measurements are shown in linear scale to show the dependencies of the LSP setup time on per-link latency and loss figures. Here we focus on the data samples for the measurements with loss and latency set to 0% and 0 ms, respectively. Those exhibit the accumulated latency introduced by the emulator environment as well as the internal WiBACK protocol processing among the Pipe Management Function (PMF) *User Module*, the Interface Management Function (IMF) and the MAC Adaptors.

In Figure 3.4, we only depict the results of the loss free case and show them in a logarithmic scale to determine the possible impact of the emulation overhead. In this scenario, with the emulated link latency set to 0 ms, the accumulated latency introduced by the internal protocol and emulation processing amounts to about 19 ms with a standard deviation of about 4 ms accumulated over eleven nodes and 20 hops, ten in the downstream and ten in the upstream path. Hence, the total average overhead, broken down per link, in this scenario is in the order of 1 ms with a standard deviation of about 200 microseconds.

For the evaluation tasks performed in the context of this work, latencies or variances in the order of 1 ms should not negatively impact our measurements, since they can be considered negligible compared to our protocol timing requirements which are mainly in the order of hundreds of milliseconds or higher. Moreover, they will introduce latencies and variances that would also be present in real-world testbeds, where the WiBACK protocol processing would be performed by embedded processors.

Chapter 4

Integration of Unidirectional Technologies

Based on the wireles back-haul architecture designed by the CARMEN project, we first discuss possible approaches to integrate Unidirectional Technologies (UDTs) so that they can be made available as regular back-haul interfaces. Considering our research questions defined in Chaper 1, we will highlight advantages and discuss the consequences of the approaches assuming as the minimum requirement on a participating node that it provides one receive-capable and one transmit-capable interface. In most real-world use cases, a node will be equipped with bidirectional and unidirectional interfaces. Hence, any additional complexity potentially required by the UDT integration should be kept at a minimum.

Following this discussion and addressing our research questions, we present the final design of our contributions which is based on the tentative design presented in Chapter 2. The architectural role of our contributions within the WiBACK architecture is depicted in Figure 2.17. We first describe our contributions to the core components such as the extended UDT-aware Abstract Interface (AI), the Technology Independent Monitoring (TIM) component which introduces passive receiver-side monitoring and the *LinkVector* extension of the *TransportService* enabling explicit source routed message forwarding among IMF entities.

Building on those core components, we then present the UDT-aware User Modules, the Pipe Management Function (PMF) and the Topology Management Function (TMF). The PMF is located at each WN and executes per-Pipe Pipe Management Protocol (PMP) instances, which provide RSVP-TE-like *Pipe* signalling among WNs while supporting signalling *around* UDTs by traversing all nodes in the path. Our proposed TMF supports UDTs with different levels of integration. It may either detect, configure and report them to the CMF for capacity allocation, or it may actively utilize them during the topology forming phase in order to establish management connectivity. The implication of the latter will be discussed.

4.1 Integration Alternatives

In this section we discuss at which level in the protocol stack a centralized WiBACK architecture could operate and what implications this would pose for the integration of UDTs.

The goal of integrating UDTs into the WiBACK architecture is to allow higher layer services to transparently utilize UDTs when they are beneficial to address physical radio or deployment conditions, given payload characteristics or receiver distributions. The following fundamental WiBACK architecture design decisions of the CARMEN consortium need to be considered when integrating UDTs:

- Control plane and hardware abstraction are based on an extension of IEEE 802.21
- Unidirectional interfaces can be identified and their capabilities described
- Topology and resource management are performed by centralized entities
- Individual flows are aggregated into QoS-aware *Pipes* of four different traffic classes
- Pipes are unidirectional resources established between any pair of nodes
- In the multicast case, a *Pipe* is extended into a 1-to-N tree

Since payload, either management or data, is forwarded via unidirectional QoS-aware *Pipes* across a WiBACK network, the data plane should be agnostic to UDTs as long as the chosen tunneling protocol provides the required QoS-differentiation. In the following sections, we therefore focus on the control plane and its signaling and management protocols such as monitoring, topology and resource management, connectivity among WiBACK nodes as well as *Pipe* signalling.

The WiBACK control plane builds on and extends the IEEE 802.21 messaging service which does not specify a specific message transport for inter-node communication, but suggests that data link communication as well network layer communication may be used. We have identified two approaches for our IEEE 802.21-based control plane signalling in a WiBACK network. The *below the network layer*, or L2.5, approach assumes no routing state to be present among the WNs, while the *at the network layer*, or L3, approach would depend on an operational routing protocol for its signalling purposes. In the following sections we will discuss the advantages and implications of the two approaches for the centralized WiBACK architecture and the integration of UDTs in particular.

4.1.1 Below the Network Layer

This approach considers signaling at and above the Data Link layer and integrates UDTs where they occur as a result of MAC protocol design decisions or physical characteristics of the radio technologies. The QoS-aware *Pipe* concept could be supported via, for example, MPLS-based forwarding on the data plane which also inherently supports the forwarding along centrally pre-computed paths. Since MPLS is considered a layer 2.5 technology, the approach proposed here would not be a pure Data Link layer approach, so the name *Below the Network Layer* approach was chosen. The use of MPLS on the data plane would complement the support for heterogeneous technologies, with an IEEE 802.21-based control plane providing support for local as well as remote management of heterogeneous nodes and their radio interfaces.

Due the centralized WiBACK architecture design, management connectivity is mainly required between the WiBACK Coordinator (WC) nodes and its associated WNs. Direct messaging between WNs may be used for MPLS FRR signaling or to optimize User Terminal (UT) mobility among neighboring WiBACK Access Point (WAP) nodes. Connectivity between the WC nodes and its WNs could be facilitated by installing MPLS LSPs among the centralized WC nodes and each WN during the association procedure. Disjoint redundant backup paths could be installed using the MPLS FRR feature to provide resilience to intermediate WN or link failures. Since the topology information to compute such paths is maintained at the WC nodes, this information would be reused, thus avoiding potential duplication and coherence issues which might be incurred by running a separate Network layer routing protocol for signalling purposes.

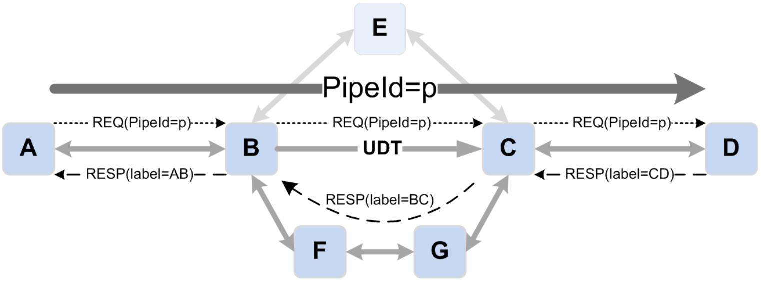

Figure 4.1: PMP uses the *LinkVector* extension to signal around a UDT via source routing

MPLS path setup is typically implemented via Network layer protocols such as RSVP-TE, which also provides the so-called Explicit Route Object (ERO) to enforce the LSP setup along centrally computed path. A similar mechanism would be required in this approach and could be implemented via explicit source routing among the IMF entities along the path to be configured. This could be facilitated by using, for example, encapsulation MIH messages describing the path to ensure that such messages would be processed at each hop along the path of the LSP to be configured. If a link provided by a UDT is present in the path, the confirmation message would have to be sent on a different path *around* the UDT while still traversing the downstream path in reverse order, see Figure 4.1 for an example. As described in Section 2.3.3, local neighbor discovery mechanisms cannot be relied upon in the case of UDTs, Hence, both the Data Link layer source and destination addresses would need to be explicitly specified.

Similar approaches to provide Link layer connectivity among non-neighboring nodes are widely used, such as the IEEE 802.1d STP. Or, the IEEE 802.11s standard, which transparently hides IEEE 802.11s mesh clouds from the Network layer.

4.1.2 At the Network Layer

This approach considers signaling at the Network layer which already addresses Data Link layer heterogeneity in order to integrate different Data Link layer technologies. The support for UDTs could be seen as an extension of that concept. In the case of IP-based protocols, the Data Link Layer technologies are typically assumed to provide bidirectional means of communication, hence an integration of UDTs would require modification to proven IP-related protocols, such as interface auto-configuration, neighbor discovery, routing or flow signaling.

Alternatively, the Link Layer Tunneling Mechanism (LLTM) could be deployed to provide tunneled return channels for UDTs to provide virtually bidirectional interfaces. Following the LLTM concept, an IP-tunneled return channel would be detected as one hop by the routing protocol even though it may span across a large set of heterogeneous links and wireless cells. As described in [55], this would complicate QoS considerations. However, this issue could be ignored, since the tunneled return links could only be used for management traffic, while user payload would be forwarded via regularly allocated data plane *Pipes*. Under the assumption that a WiBACK node always has at least one bidirectional interface, UDTs could be ignored at the cost of potentially lower mesh connectivity, as shown for BRA [39]. This is the default behavior of common WMN protocols such as AODV or link state protocols such as OSPF. In this case, nodes without a bidirectional interface could not be supported.

As a mesh network has no default routing hierarchy, a routing protocol would be required in order for non-neighboring nodes to communicate with each other for management purposes. This is required independently from the centralized TMF and CMF entities. Hence, the routing protocol would have to rediscover the topology made available by the TMF and would also be required to track any modifications administered by TMF as a reaction to link or network state changes. During transition periods, the routing protocol might not be synchronized with the TMF state and signaling attempts may fail. To optimize the coexistence, proven routing protocols could be adapted to accept hints from TMF or a special routing protocol for the WiBACK use case could be developed.

To support the QoS-aware *Pipe* concept, MPLS-based forwarding on the data plane could be utilized. Alternatively, *host routes*, IP-in-IP tunnels or forwarding based on IPv6 flow labels could be used. The use of flow labels might, though, collide with other services already relying on the flow label information. Relying on *host routes* would require a signalling mechanism to push such *host routes* into the network along the designated path. A similar mechanism would be required for IP-in-IP tunneling to enforce forwarding along the centrally pre-computed path instead of the relying on the regular routing information.

RSVP-TE could be used to set up LSPs and the ERO could be used to enforce centrally computed paths. This object may need to be extended so that it could hold the downstream and upstream path, which would need to be specified for UDTs where no LLTM return tunnel is configured.

4.1.3 Discussion

Conceptually, signalling and UDT integration either at or below the Network layer would be possible, with each approach requiring a distinct set of modifications and considerations regarding the overall architecture. Taking the aforementioned WiBACK design decisions into account, we propose an integration below the Network layer. This appears to be the more natural fit, since UDTs can be seen as a Link layer or Physical layer phenomenon and can be described and handled by the WiBACK Abstract Interface (AI) as such.

Since the WiBACK architecture is build upon a centralized management approach, the WC nodes already maintain all required topology information in the form of a topology graph data structure and a replication via Network layer routing protocols would require extra state to be kept and may cause coherence issues to be addressed. Moreover, the TMF is the authority for radio planning and channel assignments, hence Network layer protocols would have be extended to interface with the TMF in order to synchronize upon topology alternations.

Resource allocation and book-keeping as well as path computation involving UDTs can readily be supported in the *below-the-network-layer*, since UDTs are fully integrated and represented as regular links, identified by their source and destination Data Link layer addresses. In the Network layer approach, UDTs would require special handling with a varying complexity depending on the level of integration and interaction between the Network layer routing protocol and the centralized path computation module as well as the radio and topology management module.

Traffic Engineering is usually performed based on logically unidirectional resources. Likewise, MPLS forwarding is a unidirectional task. Therefore, the main WiBACK components that require modifications to fully support the utilization of UDTs are the WiBACK monitoring module, as well as the Pipe Management Function (PMF) and the Topology Management Function (TMF), both of them also requiring a source-routed signalling mechanism within the IMF's *TransportService*.

4.2 Abstract Interface Extension

The IEEE 802.21-based WiBACK Abstract Interface (AI) has been introduced in Section 2.1.4. Here, we discuss LINK_SAP modifications and extensions to the AI proposed as a result of this thesis. Those primitives introduce link calibration, per-*Pipe* monitoring events, an updated WiBACK beacon and support for regulatory events.

4.2.1 Modified Primitives

The AI primitives that have been introduced or modified as a result of this thesis are shown in Table 4.1. The AI_RadioSetupBeacon an AI_RadioCalibrateLink primitives will be described in the following subsections, while the Indication primitives will be presented later in the monitoring Section.

Primitive Name	**Description**
AI_RadioSetupBeacon	Parameterizes the WiBACK beacon
AI_RadioCalibrateLink	Instructs interface to calibrate a link
AI_PipeDown	Indicates a PIPE_DOWN
AI_RegulatoryEvent	Indicates a regulatory event

Table 4.1: Additional AI primitives used by the TMF to manage heterogeneous wireless interfaces

Link Calibration

The main role of this primitive within the WiBACK architecture is to separate the physical link parameters such as channel and maximum TxPowerLevel which are handled by the TMF from the abstract logical link properties which form the basis of the CMF's capacity allocation and bookkeeping, see Figure 2.5. The underlying assumption of this concept is that the PHY/MAC layer implementation of a specific technology should know best how to tune itself to optimally adjust to the actual link characteristics. The rather abstract TMF should handle channel assignments and transmit-power computations, while the technology specific functionality within or even below the MAC Adaptor should leverage its knowledge to optimize the radio within the channel and MaxTxPower limits specified by the TMF. The resulting technology specific settings are not relevant to the abstract WiBACK control plane. Therefore, we have designed the abstract LogicalLinkProperties object, which is returned as the result of a successful invocation of the AI_Radio_CalibrateLink primitive. This object describes the resulting link with technology agnostic properties such as nominalBandwidth, nominalLatency and ResourceModels. Such properties can readily be interpreted by the CMF which can now perform resource allocations and bookkeeping across a back-haul network consisting of heterogeneous technologies with locally optimized radio and therefore link characteristics. The specifics of this resource model as well as capacity management are outside of scope of this study. A more detailed discussion can be found in [15].

The AI_Radio_CalibrateLink primitive may be used to trigger an interface to calibrate a link to a neighboring WN. Typically, the TMF *Master* will trigger a link calibration upon establishment of a new link or in an attempt to repair a link if the monitoring subsystems report a link malfunctioning. Note that link calibration is defined as a unidirectional procedure, thus to calibrate a bidirectional link, two individual calibration requests must be triggered.

This primitive is defined sufficiently coarse to allow a mapping onto already existing mechanisms of certain wireless technologies, such as, IEEE 802.16 or 802.22. For IEEE 802.11, the MAC Adaptor may provide support to, for example, calibrate the MCS configuration, the actual TxPower level as well as the coverage class in order to automatically adjust the MAC timing. The required messaging is assumed to be handled directly among the *MAC Adaptors* via media specific communication.

Table 4.2 lists the parameters of the AI_Radio_CalibrateLink.Request primitive which

Parameter	**Type**	**Description**
LinkId	MIH_LINKID	Link to be calibrated
MaxTxPower	UNSIGNED(4)	Max. TxPower level in dBm
Distance	UNSIGNED(4)	Distance in m, if known
Mode	ENUMERATION	Capacity, Robustness
ReturnLinkVector	LIST(MIH_LINKID)	Optional return link vector to support UDTs

Table 4.2: Parameters of the AI_Radio_CalibrateLink.Request primitive

may be sent to any associated WN via the *Transport Service*. The LinkId specifies the link to be calibrated at that node. The radio planning or optimization algorithms within the TMF *Master* may have determined a MaxTxPower level to be used for the given link. If supported by the underlying technology, the respective *MAC Adaptor* is free to assign any TxPower level up to the specified maximum in order to optimally configure the given link. The distance between the radios forming the link may optionally be specified. If given, it may speed up the calibration procedure. If the distance setting is required by the given technology, the respective calibration mechanisms must determine it via RTT measurements, or other technology specific means. The Mode parameter may be used to control the calibration algorithm to i.e. prefer capacity over robustness.

Once the link calibration procedure has completed, the *MAC Adaptor* must determine the logical properties of this link, namely the nominalBandwidth, the nominalLatency as well as the parameterization of the resource model.

Typical UDTs such as DVB-T can be assumed to be statically configured and per-link calibrations are not supported. In this case the logical link properties based in the current settings are computed and returned. Since the calibration signalling is assumed to be implemented via technology specific means, support for DVB-S2 or DVB-T2 in ACM mode requires a return channel between the two involved MAC Adaptors during the calibration phase. This could be provided via DVB-RCS or via the WiBACK architecture installing a dedicated *Pipe* for the duration of the calibration procedure. Alternatively, the *LinkVector* extension of the *TransportService* could be utilized to provide a return channel via source routed messaging along the path specified by the ReturnLinkVector TLV.

This primitive supports the calibration of a point-to-point link. To optimize the transmitter configuration for native Link layer multicast among WNs, the *links* between the transmitting interface and strategically chosen receivers could be individually calibrated. The most conservative configuration among the calibrated *links* may then be applied.

WiBACK Beacon

The WiBACK beacon is used to periodically broadcast information about a WN's state within the broadcast domain of its Tx-capable interfaces. Similar mechanisms are used, for example, by IEEE 802.11 or DVB which broadcast information about their cell using beacon frames or information tables. Hence, to avoid the overhead of additional periodic transmissions, the respective *MAC Adaptor* will, where possible, extend such management frames, with WiBACK specific information, the so-called WiBACK beacon. If such mechanisms do not exist, the WiBACK beacon may be sent as a regular link-local multicast or broadcast frame. The specifics are implemented in the respective *MAC Adaptor* and the TMF may control the WiBACK beacon content via the *AI_RadioSetupBeacon* primitive, see Table 4.3:

Parameter	**Type**	**Description**
NetworkId	UNSIGNED	ID of WiBACK network
MasterId	MIHF_ID	TMF master ID
MasterTStamp	TIMESTAMP	TMF Master time stamp
SenderId	MIHF_ID	ID of sendign node
Coordinates	COORDINATES	GPS Coordinates
Distance	UNSIGNED	Hop distance to Master

Table 4.3: Parameters of the AI_RadioSetupBeacon primitive

WiBACK beacons are sent by associated WNs to provide initial information for freshly joining nodes. For example, if such a node receives beacons from multiple associated WN it may use the hop distance and the signal quality the beacon was received at, as a criterion to choose the most suitable WN in order to associate with a WiBACK network. WiBACK beacons are also interpreted by the neighboring WN's monitoring component which are using them as a heartbeat signal and to differentiate between WNs and interfering non-WiBACK transmitters. The *NetworkId* is used to separate multiple WiBACK networks. A TMF *Master* only accepts *Slaves* configured with its own *NetworkId.*

WiBACK beacons are also sent by unassociated nodes during the *well-known-channel* scan mode. In this case the *MasterId* is set to *None* to indicate the not-associated state. This mode is explained in more detail in the Section 4.6.

4.3 Technology Independent Monitoring

Monitoring in the WiBACK architecture operates at multiple network stack layers, from the Physical Layer to the MPLS layer, which is often referred to as Layer 2.5. The monitoring module performs the following tasks:

- Neighborhood monitoring and per-frame analysis
- Maintenance of neighboring node statistics
- Creation of link *up* or *down* events
- Provision of LSP end-to-end performance statistics
- Creation of QoS network state aware *Pipe* events

As presented in [11], we developed the UDT-aware Technology Independent Monitoring (TIM) extension of the CARMEN monitoring subsystem which introduces LSP monitoring and stresses the separation of measurement and statistics gathering from analysis and adaptable events creation.

Figure 4.2 depicts the architecture of the WiBACK monitoring module where TIM extension is located above a link monitoring component and the *Rating Agents* are logically located on top of the TIM extension. The *Rating Agents* operate on local neighbor, link or LSP statistics, but are controlled by and perform tasks for the topology or capacity management modules. The communication between the local monitoring subsystem and the centralized management modules is implemented via extended IEEE 802.21 Command and Event Services. While many kinds of events could be created, in this work we focus on a PIPE_DOWN event which can be issued at each hop on a per LSP basis taking the individual QoS requirements, technology-specific characteristics and overall network state into account.

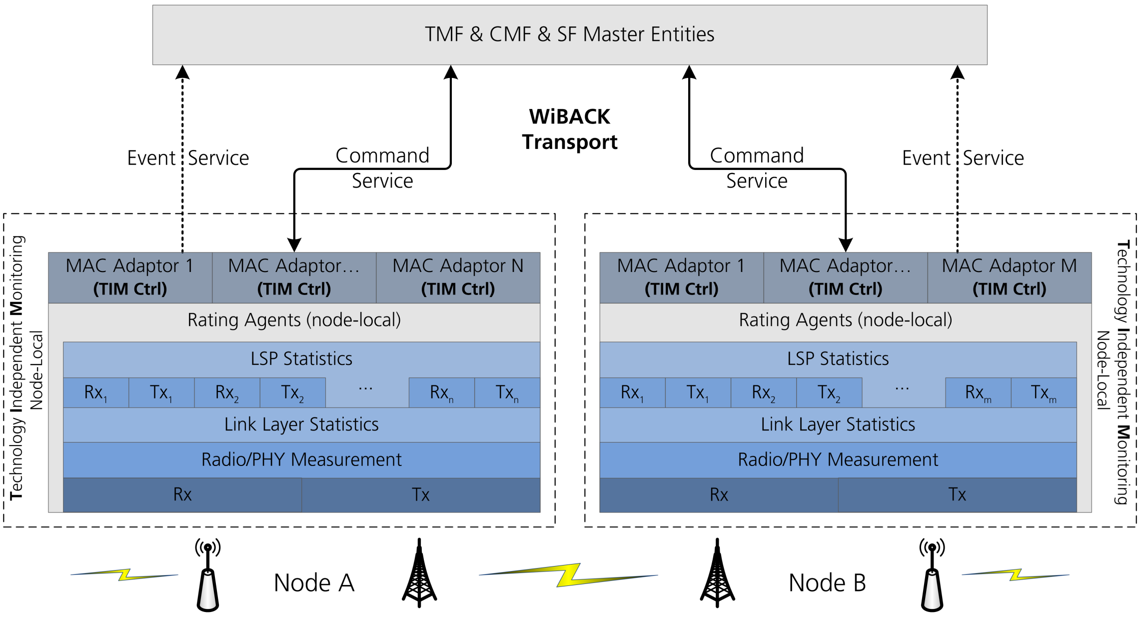

Figure 4.2: The multi-layer monitoring architecture signals via MIH messages

4.3.1 Receiving and Transmitting Side Monitoring

A link can be monitored from the transmitting side, the receiving side or from both. The data measured, its accuracy and interpretation as well as the possible events that could be triggered vary depending on the underlying technology. Hence, certain data should be monitored on a particular side of a link. For example, a periodic or constant signal needs to be present on a link in order for the receiving side to detect that a transmitter is alive. Figure 4.3 depicts one-hop LSP segments sharing a physical link and the parameters that can be measured on the transmitting side or receiving side of each LSP segment.

As suggested in RFC 3272[63] our WiBACK architecture performs end-to-end moni-

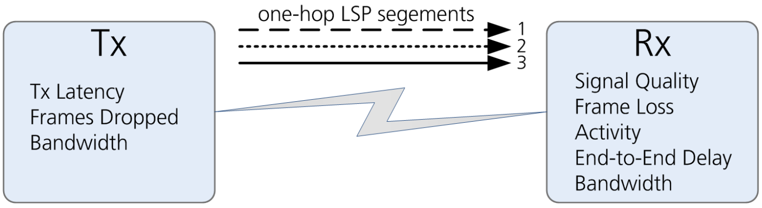

Figure 4.3: Possible transmitting and receiving side monitoring parameters per LSP segment

toring of individual LSP statistics. In addition to the actual bandwidth utilization, we also maintain loss, signal quality, delay and activity statistics which can indicate wireless link stability with a varying significance depending on the QoS requirements of the payload. This receiving side monitoring measures the actual end-to-end characteristics of an LSP and is therefore mandatory to verify if an LSP receives the agreed end-to-end QoS handling. The per-LSP QoS requirements, described by the *TrafficSpecification* record, are installed at each node on the path during the LSP setup procedure together with the LSP forwarding state. This information is therefore available to transmitting as well as receiving side monitoring.

Transmit side monitoring can provide valuable information about the per-hop QoS handling of an LSP. Depending on the wireless technology a node can detect on the transmitting side if the forwarding latency of a frame is within the guaranteed bounds or if frames are dropped due to traffic shaper policies. Also, it can measure the burst and average bandwidth utilization over various intervals. However, transmitter side loss monitoring relies on receiver feedback via, for example, a Link Layer Automatic Repeat reQuest (ARQ) mechanism. Therefore it can not detect transmission errors when frames are sent with a unidirectional link configuration. This includes multicast or broadcast frames since they are usually sent without any *ACK*.

In cases of link degradation, for example, the *rating agent* may create a PIPE_DOWN event, which needs to be signaled to the upstream Point of Local Repair (PLR) so that it may redirect the traffic onto a pre-configured backup LSP. Ensuring the delivery of the PIPE_DOWN event to the PLR can be a difficult task in a situation of unstable or natively unidirectional links. The proper calculation of backup LSPs and provision of reliable signaling paths is the task of the capacity management module and outside of the scope of this study.

As detailed above, end-to-end receiving side monitoring is mandatory to assess the QoS handling an LSP receives. Optionally, transmitting side monitoring can be used to gather additional node-local data regarding an LSP which can help with anomaly detection and faster PIPE_DOWN event delivery.

4.3.2 Technology Independent LSP Monitoring

Our monitoring approach, TIM, provides loss, delay, activity and signal quality statistics per LSP or its segments over physical links implemented by heterogeneous technologies.

This data is evaluated by *Rating Agents* to create LSP-specific events. In order to reduce wireless resource consumption, TIM tries to minimize the number of extra frames being sent or the header overhead introduced by aggregating LSP measurements and exploiting technology-specific features where possible.

TIM defines a header which can be seen as a technology independent extension of the MPLS shim header in order to provide a protocol, payload and technology independent mechanism to perform feedback-free end-to-end loss and delay measurements per LSP. As depicted in Figure 4.4, the TIM header is 32 bits wide and is always located behind the MPLS label stack, which in the WiBACK context usually only consists of one label, unless temporary backup paths via tunneling are in affect. Hence, the header parsing complexity per hop is minimal.

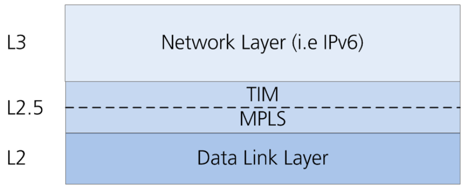

Figure 4.4: Position of the TIM header in the protocol stack *extending* the MPLS header

The TIM header as depicted in Figure 4.5 is inserted at the MPLS ingress node and thus is piggy-backed along the LSP and removed at the egress router together with the MPLS label stack. Should multiple MPLS labels be added to the stack due to FRR or other means of traffic engineering, the TIM header remains untouched in order to still allow for end-to-end LSP measurements.

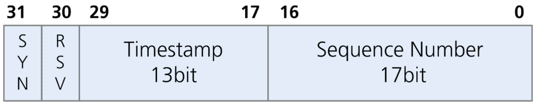

Figure 4.5: The TIM header is 32 bits wide

The ingress node maintains a 17bit wide cyclic *sequence number* per outgoing LSP, which is copied into the TIM header of each outgoing packet of the LSP. The *sequence number* wraps around after 131072 frames per LSP which can cover a sufficiently long period to detect packet loss or reordering. To identify even larger burst losses, a correlation with the time stamp field can be used to still detect the loss, while a precise quantification is no longer possible. Packet reordering should not occur since forwarding strictly follows the LSP, but might occur for short periods during a LSP fail-over process. Therefore, reordering-aware loss calculation algorithms should be used and the degree of acceptable reordering for a specific LSP should be derived from its QoS requirements. Longer periods of packet reordering would indicate a network misconfiguration, a misbehaving link or node.

The 13 bit wide *time stamp* field is filled with the current time of the ingress node when sending out the packet. The resolution is 1ms and thus allows for a maximum delay of 8192 ms to be measured. This is sufficient to cover even, for example, shared Digital Video Broadcast - Satellite (DVB-S) satellite return links with a typical latency between some hundred milliseconds and multiple seconds. The *SYN* bit is used to indicate that the ingress nodes uses a clock source synchronized to a global time base such as the one provided by the Global Positioning System (GPS) and that the implementation guarantees an accuracy of < 1ms. If the egress node's clock is also synchronized to the same time base, the time difference between the *time stamp* and the egress node's clock can be assumed to be the actual packet end-to-end forwarding delay with a +/- 2ms tolerance. If the clocks are not synchronized, the delay variation of subsequent packets can be interpreted. RFC3393 [130] provides further details regarding Packet Delay Variation (PDV) measurements with unsynchronized clocks.

Bandwidth measurements can be performed on an per-LSP basis by calculating, for example, a sliding average of the actual bandwidth consumption of the LSP. Signal quality and activity analysis could be performed similarly to the bandwidth analysis by just interpreting frames of a specific LSP. This approach would ignore valuable information, since here we are only interested in information regarding the state of the underlying link regardless of the LSP a frame belongs to. Hence, in Figure 4.3, any frame sent from node A for any LSP that is received by node B, can be interpreted as activity. Also, the signal quality and radio parameters of such a frame can be evaluated. Care needs to be taken if dynamic transmitter configurations, for example, regarding the transmit power or MCS are being used by the transmitting node. Activity and signal quality analysis are mainly of node-local importance for the *Rating Agent* to evaluate link reliability.

Intermediate WNs may examine the TIM header similarly to the egress node to maintain loss and delay statistics between the ingress node and themselves. Those statistics could be evaluated by *Rating Agents* or explicitly queried for debugging purposes to locate under-performing segments. Likewise, the TIM header may be evaluated by each WN in MPLS 1-to-N multicast trees to determine loss and delay between itself and the root.

4.3.3 Rating Agents

A *Rating Agent* can be seen as a plug-in and consists of a list of dynamically chosen and individually configured criterion functions which can create arbitrary events related to a link or an LSP such as exceeded burst bandwidth, signal quality degradation, inactivity or loss above a certain threshold. *Rating Agents* are invoked periodically after a new set of aggregated *StatisticsData* has been pulled from the data plane's measurement modules.

4.3.4 Adaptive AI_PipeDown Indication

In the context of this work we focus on AI_PipeDown indications which trigger a two-fold recovery procedure. As a quick and temporary fail-over solution, the use of a precomputed MPLS FRR backup path will be signaled to the upstream PLR of the under-performing link. In parallel, the capacity management module will receive an event about this incident.

Using its complete topology knowledge, it may decide to compute and install a new primary LSP as well as a backup LSP. Once the new LSPs have been set up, the traffic is routed onto this new path. The capacity management modules relies on such feedback from established LSPs as well as basic link monitoring to maintain overall network link state and utilization statistics.

IEEE technologies such as 802.11 or 802.16 determine a link to be *up* by monitoring the channel for periodically broadcasted frames, such as beacon frames or super frames. Broadcast technologies such as DVB permanently send out a carrier which receiving nodes acquire and maintain a *lock* on. If no actual data is available, so-called *stuffing frames* are transmitted, but are usually filtered out by the receiving hardware and can therefore not be used to detect transmitter liveliness. In the case of DVB, regularly broadcasted mandatory tables such as the PMT could be seen as the equivalent of a beacon and longer periods without a received PMT would indicate a broken or flaky link. In both cases the detection process might be too slow or not accurate enough for a WiBACK network, since it might take one second or longer until the upper layers of a node are informed about a link breakage. Additionally, the threshold when a link is considered *down* depends heavily on the QoS requirements of the affected LSPs.

Although the transmission of extra *ALIVE* frames to determine the link state should be avoided due to the additional overhead they are still necessary in some cases, especially when monitoring the availability of an idle LSP. The number of such frames should be minimized by taking into account existing traffic and by aggregating the *ALIVE* frame interval, so that it covers all LSP in questions. In Figure 4.6 different *ALIVE* intervals for the active LSPs have been configured. The actual *ALIVE* interval used is the smallest one among all intervals. The default *ALIVE* interval is initially determined by a fixed value per traffic class. It can be adjusted by the management component to optimize overall network performance.

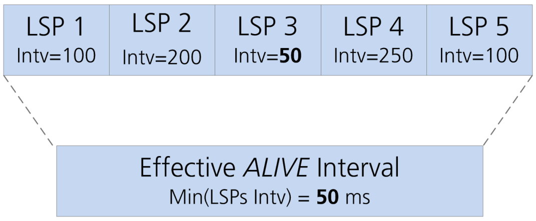

Figure 4.6: Individual per-LSP and effective *ALIVE* Interval

Most recent wireless technologies support the use of different MCS configurations per destination. For multicast or broadcast frames, typically the most robust MCS is used to ensure that all nodes can receive such frames. When interpreting regular data frames as an aggregated *ALIVE* indication, it needs to be ensured, that such frames are sent using an MCS that can be decoded by all receiving nodes. For that reason, *ALIVE* frames shoud be sent as broadcast frames at the lowest commonly receivable MCS. If such MCS can not be established the most robust MCS of the affected wireless technology should be used.

4.3.5 AI_RegulatoryEvent Indication

A *regulatory* event might be created on a link if a prioritized user for the current channel has been detected. This may for example be the case for IEEE 802.11a radios operating in the 5 GHz U-NII band, where Dynamic Frequency Selection (DFS) is mandated in most regulatory domains to, for example, avoid collisions with weather radars systems.

The initial detection is usually performed within the PHY layer and the MAC Adaptor of the respective interface must detect such events. To avoid false positives, additional filtering or pattern matching might be applied before an AI_RegulatoryEvent.Indiaction is created.

The topology management module must subscribe to AI_RegulatoryEvent indications and reassign affected channels within the time frame mandated by the regulatory agencies, as reported by the AI_Radio_GetProperties primitive.

4.4 Transport Service

The WiBACK control plane builds on and extends the IEEE 802.21 messaging service which does not specify a specific message transport for inter-node communication, but suggests that data link communication as well network layer communication may be used. In the WiBACK architecture, those aspects are handled by the *TransportService* which implements a link-local multicast and a *Pipe*-based transport. To allow for explicitly source-routed message delivery, we describe below the design of our proposed *LinkVector* extension of the WiBACK *TransportService*.

4.4.1 Link Vector Extension

Regular MIH Messages can be exchanged between IMF instances and are identified by their source and destination *MIHFIds*, which the WiBACK architecture refers to as NodeIds. The *TransportService* alone is responsible for the delivery to the destination IMF. The IEEE 802.21 standard does not specify a mechanism to pass extra routing information via the *NetSAP* to the *TransportService*. Hence, to comply with the standard as closely as possible, we introduce special Encapsulation primitives for Request, Response and Indication message which consist of an outer MIH header followed by a *LinkVector* TLV object holding the source routed path, similar to the ERO in RSVP-TE. The actual payload of those Encapsulation messages is the encapsulated original MIH message contained in a special Encapsulation TLV, see Figure 4.7.

MAC Adaptors or IMF User Modules, such as the PMF, may now send explicitly source-routed Request, Response or Indication messages using so-called *LinkVectors*, which contain the LinkIds of the links to be traversed. Such LinkIds consist of the source and destination Data Link layer addresses, which allows a specific link to be explicitly specified and no further address lookups, such as ARP are required. This is crucial to support UDTs where such lookups can not readily be supported.

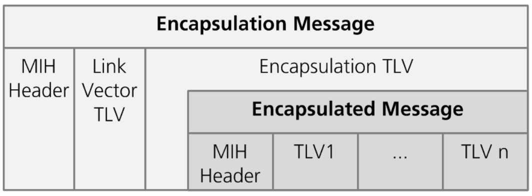

Figure 4.7: The MIH_Encapsulation primitives consist of an outer MIH header, a *LinkVector* TLV and an encapsulation TLV holding the encapsulated message

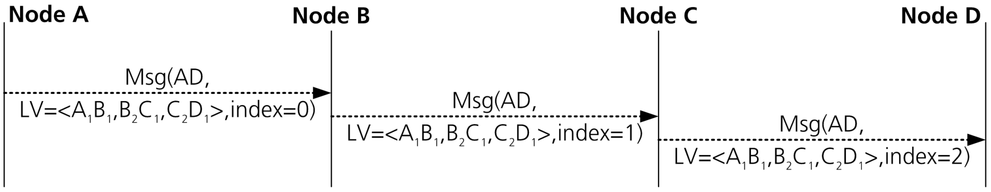

Figure 4.8: Message sent from node A to node D along the link vector $< A_1B_1, B_2C_1, C_2D_1 >$. At each node the *index* is incremented. When the *index* points to the last vector element, the message has reached the destination node.

The *LinkVector* object maintains an *index* variable pointing to the current LinkId. Hence on a sending node it refers to the outgoing link while on a receiving node, it refers to the incoming link. The originating node initializes the *index* to 0 before sending the message, while each intermediate node increases the *index* by 1. Each receiving node verifies that a message was received via the link pointed to by the *index*. In case of a mismatch the *LinkVector* is considered inconsistent and the message is dropped. If a receiving node receives a message with the *index* pointing to the last entry in the *LinkVector*, it considers itself as the destination node. Now, the *TransportService* removes the outer header and verifies that the destination NodeId specified in the original message matches its own NodeId and passes this original message on to its IMF for regular processing or delivery to the destination User Module or MAC Adaptor. Hence, the destination IMF is not aware of the source routed transport and no further modifications are required. See Figure 4.8 for an example for a message sent from node A to node D long the link vector $< A_1B_1, B_2C_1, C_2D_1 >$.

This extension of the WiBACK *TransportService* is, however, not supported by the standard IEEE 802.21 *AckService* since it can not interpret the *LinkVector* TLV, nor would it be able to compute a return path for *ACK* messages, especially in cases were UDTs are present. Therefore, where a Response message is to be sent as a reply to a Request message, it is assumed that the path is determined by application specific means.

If the vector only contains one element, this mechanism provides an explicit link local delivery, where the local sending as well as the remote receiving interface are specified via the LinkId. In this case, the remote address may be a multicast address to reach all nodes associated with the specified source interface. This allows multicast messages to be sent out on a specific interface only, while regular IEEE 802.21 multicast messages are sent on all transmit-capable interfaces.

4.5 Pipe Management Function

Our Pipe Management Function (PMF) is designed as an IMF User Module which is present at each WN and can execute multiple Pipe Management Protocol (PMP) instances for individual *Pipes* identified by their PipeId. The PMP builds upon the *LinkVector* extension of the WiBACK *TransportService* as described in the previous section. This extension allows the PMP to send source-routed MIH encapsulation messages specifying the exact path of MIH messages towards the destination node, which, in turn, allows the PMP to implement RSVP-TE-style *Pipe* signalling.

4.5.1 Pipe Management Protocol

The Pipe Management Protocol (PMP) builds upon proven RSVP-TE concepts where possible and heavily utilizes the *LinkVector* extension of the WiBACK *TransportService* in order to support hop-by-hop signalling from the ingress node toward the egress node of a *Pipe*, see Figure 4.9.

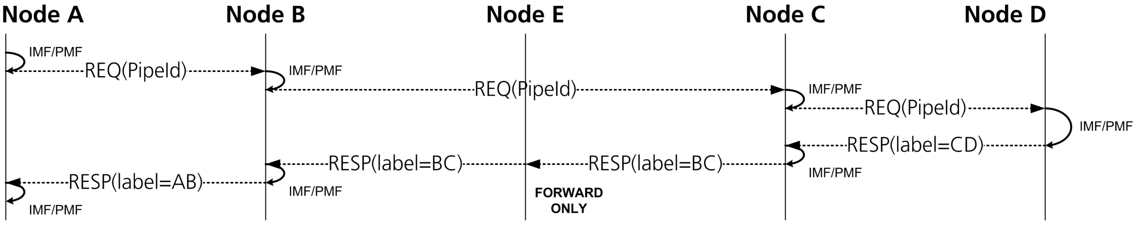

Figure 4.9: MSC of scenario shown in Figure 4.1 depicting the use of single-hop *LinkVector* message forwarding on the downstream path and partial multi-hop *LinkVector* forwarding via node E on the upstream path to circumvent the UDT between node B and C

As depicted in the MSC in Figure 4.9, PMP uses single hop forwarding on the downstream path, since the Request messages need to be actively processed at each WN along the path. On the return path, the Response message might take an alternative route *around* UDTs, see Figure 4.1. Here multi-hop *LinkVector* forwarding may be used, since the Response message only needs to be actively processed by the WNs in the signalled path. In Figure 4.1, nodes G and F would be such forwarding WNs, which are not part of the actual *Pipe* being signaled, and therefore merely act as forwarding hops for the response message to circumvent the UDT. Apart from the depicted return path via nodes G and F, a path via node E would also have been possible. The decision on the exact paths is made by either TMF or CMF, while PMP executes along those chosen paths. If UDTs are present, the return path must be explicitly specified. If no UDTs are present it may be specified or, alternatively, is derived by reversing the downstream path.

Our proposed PMP supports four tasks, which are described in detail in the following subsections:

- *Pipe* Setup
- *Pipe* Removal

- *Pipe* Modification
- *Pipe* Failover

For each task IEEE 802.21-compatible Request/Response respectively Indication primitives have been defined, which contain the relevant *Pipe* state and QoS resource allocation information, similar to RSVP-TE. The respective pairs of Request and Response primitives allow for a nested hop-by-hop signalling similar to RSVP-TE using the Explicit Route Object (ERO). The source-routed signalling mechanism is described based on the *Pipe* setup procedure, which is the most comprehensive task, since here new state is installed along the signaled path, while the removal and modification tasks operate on the state installed during the *Pipe* setup phase. *Pipe* fail-over signalling via an MIH Indication is a separate task which does not rely on nested hop-by-hop signalling, but may also utilize the *LinkVector* extension if no management connectivity exists towards the destination node as it may typically be the case for PLRs.

Pipe Setup

The equivalents of RSVP *PATH* and *RESV* messages have been defined as PMF_PipeSetup. Request and PMF_PipeSetup.Response primitives. See Tables 4.4 and 4.5 for a list of parameters contained in the two primitives:

Parameter	**Type**	**Description**
PipeId	PIPE_ID	PipeId of this new *Pipe*
TrafficSpecs	TRAFFIC_SPECS	Traffic specification of the *Pipe*
Type	ENUMERATION	Primary, Backup or Multicast
PayloadType	ENUMERATION	i.e IEEE 802.21, Ethernet, IP
Labels	LIST (LABEL)	Upstream-assigned labels
DownstreamLinkVec	LIST (LINK_ID)	Actual path to be configured
UpstreamLinkVec	LIST (LINK_ID)	Signalling return path, optional
Epoch	TIMSTAMP	Node Timestamp/Epoch
Parameters	PMP_Parameters	Individual PMP configuration

Table 4.4: The PMF_PipeSetup.Request primitive contains the above TLV-encoded parameters

Parameter	**Type**	**Description**
Status	STATUS	IEEE 802.21 Status codes
Labels	LIST (LABEL)	Up/Downstream Labels
FailedNodeId	NODE_ID	NodeId of a failed Node
MaxTransmitUnit	UNSIGNED	MTU discovered for this *Pipe*
Statistics	PMP_Statistics	*Pipe* signalling statistics

Table 4.5: The PMF_PipeSetup.Response primitive contains the above TLV-encoded parameters

The actual path of the *Pipe* to be installed is specified by the *DownstreamLinkVec* TLV, while the return path may either be explicitly specified using the *UpstreamLinkVec* TLV, or if that TLV is not present, it is derived by reversing the *DownstreamLinkVec*, see Figure 4.9.

This source routed signalling along the data path provides an implicit test of each link along the path, which aids TMF in detecting potential link instabilities during the bootstrapping phase. The PMP_Statistics TLV of the Response message holds more detailed information such as total setup time as well as the total number of sent Request and Response messages indicating potential link stability issues.

PMP can be instructed to set up either unicast *Pipes* or 1-to-N multicast LSP *Trees*. In the first case, PMP generates and distributes regular downstream-assigned labels while in the latter case PMP generates and distributes upstream-assigned labels. According to RFC 5331, the context must, at least, be unique among directly adjacent WNs. Within the WiBACK architecture, the 20bit context ID can easily be derived from a unique NodeId. Analogously to the mapping described in RFC 5332, the destination multicast MAC addresses for each segment of a multicast LSP is derived from the respective upstream-assigned label of this segment. Hence, no further address lookup or negotiation is required. PMP maintains 1-to-N multicast *Trees* by successively adding to or removing branches from the tree, while the computation of the *Tree* is performed by CMF.

In addition to label assignment and LSP state configuration, PMF also allocates the associated *Pipe* resources with the respective MAC Adaptor of each outgoing link along the path by locally triggering the AI_LinkAllocateResource primitive of the respective MAC Adaptor. The MAC Adaptor may use the provided information, mainly the *TrafficSpecifications* of the respective *Pipe*, to optimally configure the underlying wireless technology. Since WiBACK manages resources centrally, it is envisaged that this information is used to program traffic shapers, monitoring thresholds or scheduler and queue parameters. Contrary to the RSVP SBM, wireless cells in the WiBACK architecture are not considered authoritative for resource allocations, but rather monitor and report deviations from the centrally computed configuration. The details of WiBACK capacity handling, which are outside the scope of this thesis, are discussed in [15].

Another service provided by the PMP is a per-Pipe Maximum Transmit Unit (MTU) discovery. Upon a successful setup, the PMP reports the discovered end-to-end MTU via the *MaxTransmitUnit* TLV of the Response primitive. This value may vary on a per-*Pipe* basis depending on MTU of the underlying technologies in the path, the size of the MPLS label stack, the presence of additional monitoring headers or possibly required padding of encryption mechanisms.

For each *Pipe*, the PMF maintains a so-called *PipeState* object at each traversed WN, see Figure 4.10. *Pipes* are identified by a PipeId which consists of the NodeId of the ingress WN and a 32bit-wide descriptor assigned by the PMF instance at the ingress WN. A PipeId serves as a network-wide unique identifier and is present in each PMF_PipeSetup.Request, PMF_ModifyPipe.Request, PMP_PipeRemove.Request or AI_PipeDown.Indication message. A *Pipe* identifies an LSP and its associated QoS resources given in the form of a *TrafficSpecifications* record.

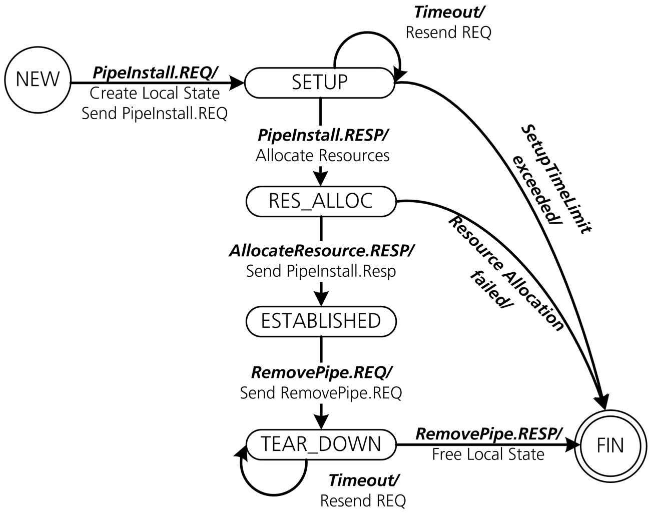

Figure 4.10: Simplified per-*Pipe* state machine maintained by the PMP traversed WNs

PMP is required to provide a robust *Pipe* signaling mechanism that quickly and reliably executes *Pipe* setup, modification or remove requests from either TMF or CMF, even under suboptimal link conditions. Therefore, the retransmission behavior can be parameterized on a per *Pipe* basis, possibly depending on the wireless technology being used or the current channel conditions. If a setup or remove procedure fails, PMP uses the *failedNodeId* TLV of the Response primitives to indicate the first node on the downstream path causing the error. This information can be examined by either TMF or CMF in order to take appropriate corrective actions. A default PMP parameterization for typical use cases will be determined in Section 5.

RSVP provides an epoch field that denotes the creation time of a node. This information is used to detect stale state in the case a node has been restarted, for example, after a crash or network outage. In the WiBACK architecture the epoch check among WNs is a task of the TMF, but PMP states and messages also maintain the epoch time stamp to allow independent consistency checks by the PMF *garbage collector*. Inconsistent state might be created due to link failures or network partitioning when established *Pipes* are considered broken, or when the setup or remove procedures do not complete successfully. In such a case, TMF or CMF may remove the affected *Pipe* and its allocated resources from their internal graphs or tables, while stale *Pipe* state is handed over to the *garbage collector*, which will asynchronously attempt to remove stale *Pipe* state, either by partially re-initiating the nested removal sequence or by explicitly removing it from affected nodes.

Pipe Removal

The *PathTear* message has been implemented via a PMF_PipeRemove.Request primitive, which for PMP is explicitly confirmed with a PMF_PipeRemove.Response primitive. See

Tables 4.6 and 4.7 for the parameters of those primitives. Optionally, an *UpstreamLinkVec* TLV may be present in the Request primitive indicating an alternative signalling return path to be used. This might be required, for example, if UDTs are present in the *Pipe* and the initial return path is no longer valid due to topology changes.

Parameter	**Type**	**Description**
PipeId	PIPE_ID	PipeId of this new *Pipe*
UpstreamLinkVec	LIST (LINK_ID)	Signalling return path, optional

Table 4.6: The PMF_PipeRemove.Request primitive contains the above TLV-encoded parameters

Parameter	**Type**	**Description**
Status	STATUS	IEEE 802.21 Status codes
FailedNodeId	NODE_ID	NodeId of a failed Node

Table 4.7: The PMF_PipeRemove.Response primitive contains the above TLV-encoded parameters

Pipe Modification

The PMF_PipeModify.Request and PMF_PipeModify.Response primitives allow the resources associated with a *Pipe* as specified via the *TrafficSpecifications* TLV to be altered. This procedure may be used to, for example, increase the bandwidth of a VoIP *Pipe* to reflect an increased demand. See Tables 4.8 and 4.9 for the parameters of those primitives.

Parameter	**Type**	**Description**
PipeId	PIPE_ID	PipeId of this new *Pipe*
TrafficSpecs	TRAFFIC_SPECS	Traffic specification of the *Pipe*
UpstreamLinkVec	LIST (LINK_ID)	Signalling return path, optional

Table 4.8: The PMF_PipeModify.Request primitive contains the above TLV-encoded parameters

Parameter	**Type**	**Description**
Status	STATUS	IEEE 802.21 Status codes
FailedNodeId	NODE_ID	NodeId of a failed Node

Table 4.9: The PMF_PipeModify.Response primitive contains the above TLV-encoded parameters

Fail-over Signalling

If a failure of an underlying link or a PIPE_DOWN is detected by the monitoring component, the detecting node may trigger an AI_PipeDown.Indication primitive to be sent to the *Pipe's* Point of Local Repair (PLR) via the IEEE 802.21 event service in order to trigger a *Pipe* fail-over, similar to the MPLS FRR extension. See Table 4.10 for the parameters of this primitive. In cases where the PLR is not the TMF *Master* node, no *Management Pipe* might exist between the triggering WN and the PLR node. Hence, the *LinkVector* extension is relied upon to deliver the *Indication* message. In this case, the path must be pre-computed and installed during the backup LSP setup phase. Multiple disjoint paths may be provided in order to increase the chance of successful *Indication* message delivery in the presence of network errors. The decision if a *Pipe* is to be protected with a backup is out of scope of the PMF and might depend on configurable policies of the TMF or CMF entities.

Parameter	**Type**	**Description**
InterfaceId	INTERFACE_ID	Id of the reporting (receiving) interface
PipeId	PIPE_ID	Id of the affected *Pipe*

Table 4.10: The AI_PipeDown.Indication primitive contains the above TLV-encoded parameters

4.5.2 Adjustable Reliability Mechanism

Depending on the wireless technology and its configuration, especially during the bootstrap phase, wireless links may be subject to relatively high loss figures compared to, for example, optical fiber links. Analogously to RSVP using the ERO and the *MessageId* extension, PMP implements nested hop-by-hop MIH transactions and adjustable timeout handling to achieve robust hop-by-hop *Pipe* signalling under loss conditions.

For regular MIH transactions, the IMF's end-to-end *AckService* can be deployed to provide message acknowledgements and to trigger retransmissions of lost or late messages. This mechanism is transparent to the IMF's transaction manager and can therefore simply resent messages between the *AckService* instances at the source and destination IMFs. However, MIH messages sent via the *LinkVector* extension are sent from IMF User Modules or MAC Adaptors and each request message creates a new MIH transaction with the local IMF while a remote IMF would only accept one response message in return for a delivered request. Hence, neither request nor response messages can simply be *retransmitted* by the respective module.

Instead, on the module level, a new MIH transaction must be created by the originating module in order to resend a previously timed out transaction. Old transactions must be closed in order to free the associated *transactionId*, which is a rather limited resource in the IEEE 802.21 messaging system. The WiBACK IMF divides the maximum number of 4096 *transactionIds* in two segments of 2048 for each direction. These limitations must be considered when designing a reliability mechanism for PMP.

The goal of the PMP reliability mechanism is to support fast, robust and confirmed *Pipe* signalling. Hence, PMP needs to trade-off between a low maximum signalling time and the resources required, either in terms of total signalling packets sent or in open MIH transactions.

An initial study considered an approach with only one open transaction [7], but, for a ten-hop scenario, was found to yield rather long setup times of up to 50 seconds under higher loss conditions. In this study, we present a slightly more aggressive mechanism which may use multiple parallel MIH transactions in order to more quickly recover from message loss, see Figure 4.11. This approach does not close pending parallel transactions of an active setup procedure which might still be open due to late responses because of higher link or processing latencies instead of packet loss. Each node controls its own transaction resend timers with a capped exponential back-off, independently from its position in the path setup chain. In contrast to the initial study, transactions are only resend among adjacent WNs, which will either respond immediately with a Response message if they are already in *ESTABLISHED* or *FIN* state, or queue the Request for a later response while they are in the transitional *SETUP* or *TEARDOWN* states. Figure 4.11 depicts a corresponding message sequence chart of a loss-impacted *Pipe* install procedure.

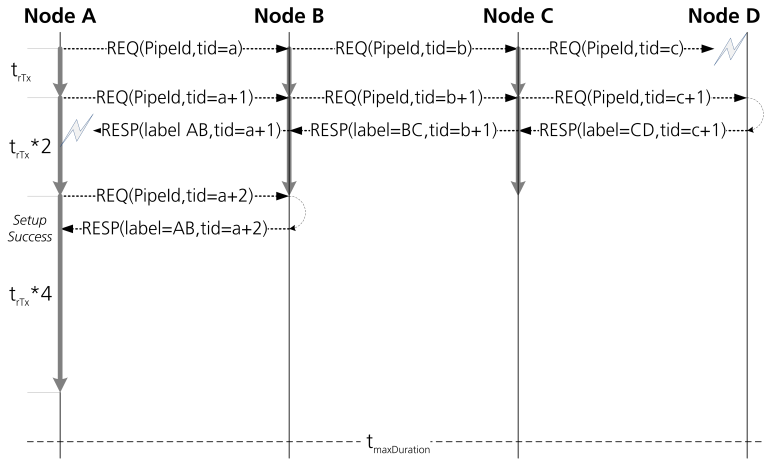

Figure 4.11: MSC of scenario shown in Figure 4.1 depicting the inter-node PMP communication focusing on retransmission timing and multiple open transactions, thus omitting the node-local resource allocation messages exchanged with the MAC Adaptor

In Figure 4.11, node A starts the setup procedure with a REQ message towards node B which in turn sends a REQ to node C eventually sending a REQ to node D. This last REQ message is lost and node C sends a new REQ with a new *TransactionId* after the retransmission timer t_{rTx} has expired. This REQ is immediately confirmed with a RESP message by node D informing node C about the downstream-assigned label. Node C, in turn, sends a RESP message to node B which sends a RESP to node A. This RESP message is lost. Hence node A will send a new REQ message ($tid = a + 2$) towards node B. Since node B is already in *ESTABLISHED* state, it immediately responds to this new REQ message with a corresponding RESP message ($tid = a + 2$). At this point the setup procedure has succeeded. The REQs with tid=a+1 and tid=b+1 were triggered by local retransmission timer expiration due to delayed RESP messages further down the path and are an inherent affect of the proposed protocol. A possibly impact of such additional messages while be discussed in Section 4.5.4.

Depending on the scenario, this mechanism can be parameterized to tolerate higher latencies and loss figures or rather to yield very fast signalling under optimal link conditions. PMP sessions can be parameterized with the following three parameters, t_{rTx}, $t_{rTxCutOff}$ and $t_{maxDuration}$ which control the initial retransmission timeout, the maximum retransmission timeout effectively limiting the exponential back-off and the maximum setup duration after which PMP considers a signalling attempt as failed. Default parameters for typical WiBACK scenarios are determined and evaluated in Section 5.

4.5.3 Multicast Signalling

Multicast 1-to-n *Tree* state is signaled for each WN joining or leaving the *Tree*. Instead of signaling end-to-end as it is done in the unicast case, only the branch from the closest branch point to the affected WN needs to be signaled. This branch point is determined by the CMF. For signaling purposes, this new branch can mostly be regarded as a unicast *Pipe*. Hence the same internal PMP primitives can be used, but are issued from the branch point and not from the root of the tree. If receiving WNs are added to an already existing branch, it depends on the underlying technology if link layer multicast can be used to reach multiple receivers with a single multicast link layer frame. In this case, no additional resource allocations are required, but the LSP state needs to be configured at the receiving WN. If link layer multicast is not supported, additional branches, with their own resource allocations, need to be configured for each receiving WN.

4.5.4 Protocol Analysis

Figures 2.15 and 4.11 show that, in a loss-free scenario, standard RSVP-TE and PMP should perform equally, requiring the same total number of messages to be exchanged and yielding a similar *Pipe* setup time. Under loss conditions, PMP should perform similar to RSVP with the *MessageId* extension, while the actual setup times depend in both cases on the parameterization. Both protocols handle lost message on a hop-by-hop basis, hence retransmitted messages are not propagated down the signalling chain.

In loss-free cases, the signalling overhead of PMP is minimal, since a path signalling procedure consists only of the equivalent of one end-to-end downstream *Request* and upstream *Response* message pair. As a hard-state protocol, PMP does not require periodic state refresh messages.

In cases of packet loss, multiple hop-by-hop transactions may be triggered on all segments upstream of the flaky link. Assuming a typical PMP parameterization as evaluated in Chapter 5, the equivalent of less than 10 transactions will be generated along the path. Compared to typical *Pipe* payload packet rates of 100+ pkts/s or 1000+ pkts/s this can be considered as not significant.

A critical aspect for the WiBACK control plane may be the rather limited number of open transactions. Under larger packet loss conditions and assuming a maximum of 10 open transactions per *Pipe*, about 200 *Management Pipes* and *Data Pipes* can be signaled in parallel per *Master* node. In larger scenarios, TMF or CMF should address this issue by tracking the *Pipe* signalling rate of their associated PMF instance and temporally distribute *Pipe* signalling requests, if necessary.

Due to the hard-state nature of PMP, signalling of multicast *Trees* is subject to scalability limitations. However, only WNs may subscribe to *Trees* and WAP nodes would act as proxies for their possibly numerous UTs. Hence, the signalling overhead should be manageable as long as *Tree* memberships are rather static. In more volatile scenarios this aspect should be reconsidered.

4.6 Topology Forming and Maintenance

In this section we present our Topology Management Function (TMF) design, which relies upon our extended Abstract Interface (AI) and follows the overall ring-based master/slave concept of the CARMEN Self-configuration Function (ScF). We considered two levels of support for unidirectional connectivity within the TMF where UDTs would either be utilized when establishing the initial management connectivity or would be avoided during the association phase. In both cases UDTs would be configured and exposed to CMF for data *Pipe* allocations, their main application in the WiBACK architecture.

4.6.1 TMF Design

The TMF is designed as an IMF User Module using a master/slave model and introduces a set of primitives to facilitate the communication among the TMF entities. TMF *Masters* are located at WiBACK Coordinator (WC) nodes while TMF *Slaves* are instantiated at each WN (Figure 4.12).

A TMF *Master* manages all possible physical links among nodes within its administrative domain, providing a framework to utilize a variety of optimization strategies depending on the use case. For this initial TMF validation, we have implemented a local optimization mechanism to form an interference-free meshed topology of point-to-point links out of all possible physical links among multi-radio WNs, see section 4.6.7. This subset, the so-called *logical links* may include 1-to-N broadcast cells and is exposed to the CMF which only operates on this subset.

In order to allow the CMF to perform constraint based path computation, the TMF *Master* describes the logical links with attributes, such as nominal bandwidth and link latency, as determined by the AI_Link_Calibrate primitive, see section 4.2.1. When topology changes occur, the TMF *Master* checks if the current logical configuration or any of its management connections are affected. In such a case it will inform the CMF that the affected links will be taken off-line and then attempt to reconfigure the available radio interfaces of the affected WNs, by either issuing an AI_Link_Calibrate request or by assigning a different channel or a new peer.

To quickly react to link quality changes which may impact the available bandwidth as well as the latency, CMF performs its own monitoring on a faster timescale and should have detected issues with *Pipes* already before TMF reports a *Link* as down. In most cases CMF should already have shifted the affected traffic onto backup *Pipes* or should have established new *Pipes* altogether. When the TMF removes a logical link, it is up to the CMF to permanently repair or rearrange those data *Pipes* affected by that link. Affected management *Pipes*, however, are reconfigured by the TMF itself in order to insure management connectivity independently from CMF. For *Pipe* management tasks, the TMF and the CMF rely on the PMF to set up, modify or remove *Pipe* state.

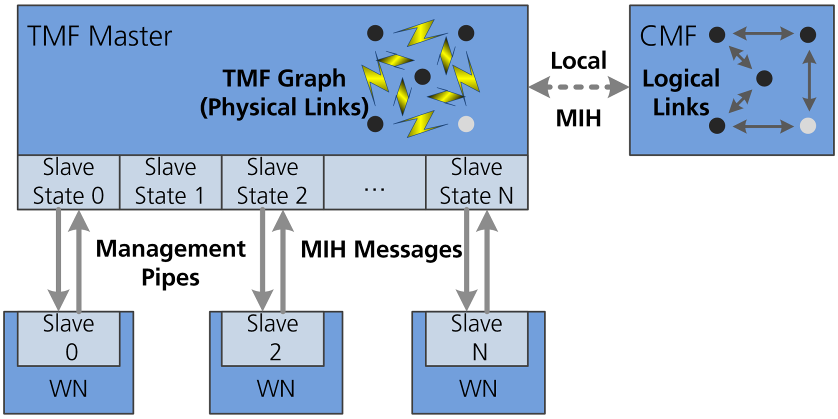

Figure 4.12: The TMF uses a master/slave model and communicates via MIH messages through so-called *Management Pipes*. The optimal subset of assigned *logical links* is made available to the CMF.

In the following subsections we first introduce the key functional components of the TMF, followed by the description of the ring-based master/slave approach, our initial optimization scheme and a protocol scalability analysis.

4.6.2 Identifiers and Resource Descriptions

The WiBACK architecture uses different identifiers for its resources, such as a NodeId, an InterfaceId, a LinkId and a PipeId. The first three identifiers have been taken from the IEEE 802.21 standard, with the NodeId being the equivalent of the *MIHFId*. InterfaceIds are generated from the hardware address of each interface and a LinkId uniquely identifies a link between two interfaces. NodeIds and InterfaceIds are considered to be unique across a WiBACK network and TMF *Masters* will verify this upon an association of a *Slave* and reject *Slaves* with non-unique identifiers.

During the WN bootstrap period, the NodeId is derived from the InterfaceIds of the local Interfaces, by e.g hashing them or using the lowest address. The InterfaceId contains a *LinkType* field which indicates the underlying technology as well as an *AddressFamily* field which indicates the hardware address type. Hence, potentially any kind of hardware

Parameter	**Type**	**Description**
InterfaceId	INTERFACE_ID	MIH InterfaceId
State	ENUMERATION	UP, DOWN, ...
Channels	LIST(CHANNEL)	Frequency, Bandwidth
TxPower	LIST(TXPOWER)	TxPower Levels
Directionality	ENUMERATION	Rx, Tx, Duplex
RegulatoryInfo	REG_INFO	Channel scanning and reaction times, etc.

Table 4.11: TMF describes Interfaces using the *TMFInterface* object

Parameter	**Type**	**Description**
NodeId	MIHF_ID	Node ID
State	ENUMERATION	Discovered, Operational
NetworkId	UNSIGNED	ID of WiBACK network
Distance	UNSIGNED	Hop Distance to Master
TimeStamp	TIMESTAMP	Node creation time
Coordinates	COORDINATES	GPS Coordinates
Interfaces	LIST(TMFInterface)	Local Interfaces

Table 4.12: TMF describes Nodes using the *TMFNode* object

address can be supported while the *LinkType* can be used to indicate hardware-specific features. In the WiBACK context, the *LinkType* is used to determine if a physical link is bidirectional or unidirectional. For example, *Ethernet*, *IEEE_80211* or *IEEE_80216* are considered to provide bidirectional connectivity while *DVB* or *ATSC* are considered to provide only unidirectional connectivity.

An MIH LinkId consists of an InterfaceId describing the source and a *LinkAddress* describing the destination which must be of the same *LinkType*. Hence a LinkId also indicates if its underlying technology provides bidirectional or unidirectional connectivity. Since WiBACK considers links as unidirectional resources, physical connectivity provided by a bidirectional technology is represented by a pair of LinkIds. During operation, TMF will verify that such a bidirectional connectivity for bidirectional technologies exists, otherwise the affected link pair will be marked as *FLAKY* and not considered for traffic forwarding until it has been reconfigured and bidirectional connectivity could be verified.

The *TMFGraph* maintained at *Master* nodes stores *TMFNodes* as vertices which may hold multiple *TMFInterfaces* while *TMFLinks* are represented as edges, see Tables 4.11, 4.12 and 4.13.

4.6.3 BeaconScan Procedure

The *BeaconScan* procedure aims at detecting all possible physical connectivity among the scanning WN and its neighboring WNs in order to allow the TMF to choose the optimal links among all possible options. Typically, the *BeaconScan* procedure is executed by associating TMF *Slaves* on all or a selection of their Rx-capable interfaces. Alternatively,

Parameter	**Type**	**Description**
LinkId	MIHLINK_ID	MIH LinkId
State	ENUMERATION	Discovered, Operational
Properties	LINK_PROPERTIES	Bandwidth, Latency

Table 4.13: TMF describes Links using the *TMFLink* object

it may be triggered by the *Master* to obtain an up-to-date view of the physical connectivity among neighboring WNs and the scanning WN.

This set of neighboring WNs serves two purposes. It may be sorted by signal quality and hop distance of the sending node in order to determine the most suitable proxy WN to associate with a *Master*. During the association procedure this set is also shared with the *Master* which examines it in order to determine all possible physical connectivity of the associating WNs. Based on this information, the *Master* may determine an optimal alternative link or WN for the on-going association.

Especially in black-out scenarios, multiple unassociated neighboring WNs may be executing the *BeaconScan* procedure simultaneously, however, without coordination. Since unassociated WNs do not send regular WiBACK beacons, they would not be able to discover each other. Therefore the *BeaconScan* procedure is split into two stages. The first stage performs a so-called *wellknown-channel-scan* where the channel is determined independently for each technology and is chosen as the lowest supported channel in the current regulatory domain. Alternatively, it may be pre-configured. In this mode, the beacon scanner periodically broadcasts WiBACK beacons with the *MasterId* set to *None* while at the same time interpreting beacons received from neighboring WN. The second stage performs a passive *all-channel-scan* for WiBACK beacons on all channels supported by the respective interfaces. Those two scanning procedures are alternated. To ensure that the independently running WNs overlap during the *well-known-channel-scan* period, this period must be longer than the *all-channel-scan* period (Figure 4.13).

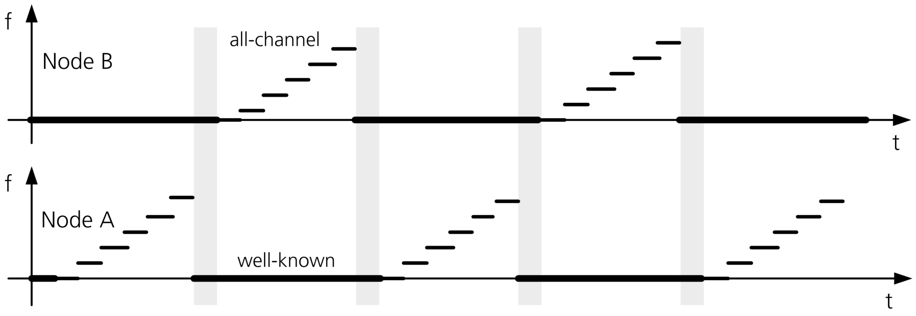

Figure 4.13: The TMF slaves execute the *BeaconScan* procedure to detect all possible physical connectivity to neighboring WNs by alternating between the *well-known-channel* and the *all-channel* scan.

4.6.4 Channel Scanning and Assignment

The *ChannelScan* procedure is executed by WNs in order to assess the current channel utilization on a specific or all rx-capable interfaces. This information is typically used as input for a local or network-wide radio planning or optimization function. To comply with possible regulatory constraints, such DFS in the 5 GHz U-NII band, the duration of the channel scan procedure on certain interfaces might have to be extended in order to detect possibly present priority users, such as weather radars.

In local optimization mode, a WN may attempt to assign the least utilized channels while maintaining a configurable channel separation among node-local interfaces to minimize inter-radio cross-talk.

In network-wide optimization mode, the *Master* might poll some or all its *Slaves* and perform channel (re-)assignments following administered optimization goals. To push such optimization results back into the network, the order of the reconfigurations should be carefully considered since channel reassignments may require link calibrations which may cause considerable connectivity interruptions.

Hence, the involved TMF instances must be aware that channel reconfigurations may fail due to, for example, communication issues on the new channel. After a timeout they should revert back to the old configuration. As a last resort, if the *Slave's* management *Pipes* are affected, that *Slave* might be required to begin a new association procedure, which might affect other WNs connected via this failed *Slave*.

4.6.5 Ring-based Master/Slave Approach

TMF is executed as a continuous process aiming at forming and maintaining a meshed network of multi-radio nodes according to its configured optimization goal. Management nodes executing a TMF *Master* instance are either assumed to be designated manually as a result of off-line network planning or may be elected at run-time. The criterion to qualify as a TMF *master* node may be the availability of sufficient resources to execute the TMF functionality or the presence of a back-bone connection, since in many practical, and especially rural use cases, a node with back-bone connectivity can be assumed to be operated more reliably. In the cases of network partitioning, other leader election mechanisms may be used to provide connectivity within the network partitions.

The initial discovery of new WNs is performed by forming logical rings around a *Master*, see Figure 4.14. The first ring consists of WNs with direct radio associations. WNs belonging to the next outer rings aim at associating with nodes on the inner rings. WNs discover associated neighbors by passively scanning all available channels on their Rx-capable interfaces for WiBACK beacons. If the BeaconScan procedure reports a *Master* or already associated WNs the associating WN will attempt to associate through them with the *Master*. Especially after larger network outages, careful consideration is required to avoid storms of association requests, since depending on node distribution and connectivity, the number of potentially newly discovered WNs may increase exponentially with

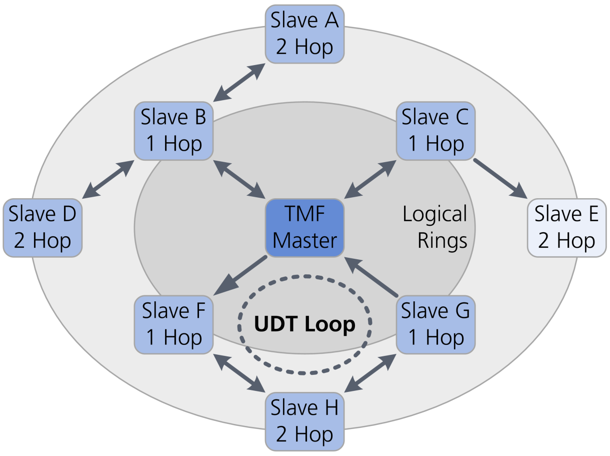

Figure 4.14: The *Master* node discovers new WNs by forming logical rings based on the hop distance.

the hop distance from the *Master*. Larger bursts of association requests may overload the *Master* which might perform rather complex topology optimizations taking into account the new connectivity options provided by the associating WN. Links in the vicinity of the *Master* may experience additional load bursts since an association requires increased signalling to establish *Pipes* and to update neighbor information.

To temporally distribute the association requests of a potentially exponentially growing number of WNs, we have applied an exponentially increasing randomized back-off timer depending on the hop distance d of the associating WN. To limit the growth of the back-off time for the maximum 10-hop WiBACK use case, we are using a *tamed* exponential back-off function:

$$MaxBackoff(d) = \frac{2^d}{(d+1)^2} * C \tag{4.1}$$

The constant C determines the range of the random back-off timer with smaller values of C yielding a tighter timing and thus shorter discovery times but possibly introducing a high load at *Master* nodes.

4.6.6 Association Procedure

The following subsections describe the association procedure from the *Master's* (section 4.6.6) and the *Slave's* (section 4.6.6) point of view.

Master State Machine

During its initialization phase, a *Master* first detects its own capabilities in terms of number, type and properties of available radio interfaces by calling the AI_RadioGetProperties primitive for each interface reported by the IMF. Then, the *Master* performs a channel scan on all available Rx-capable interfaces in order to determine the least utilized channels. After the successful completion of the channel scan procedure, the *Master* determines and

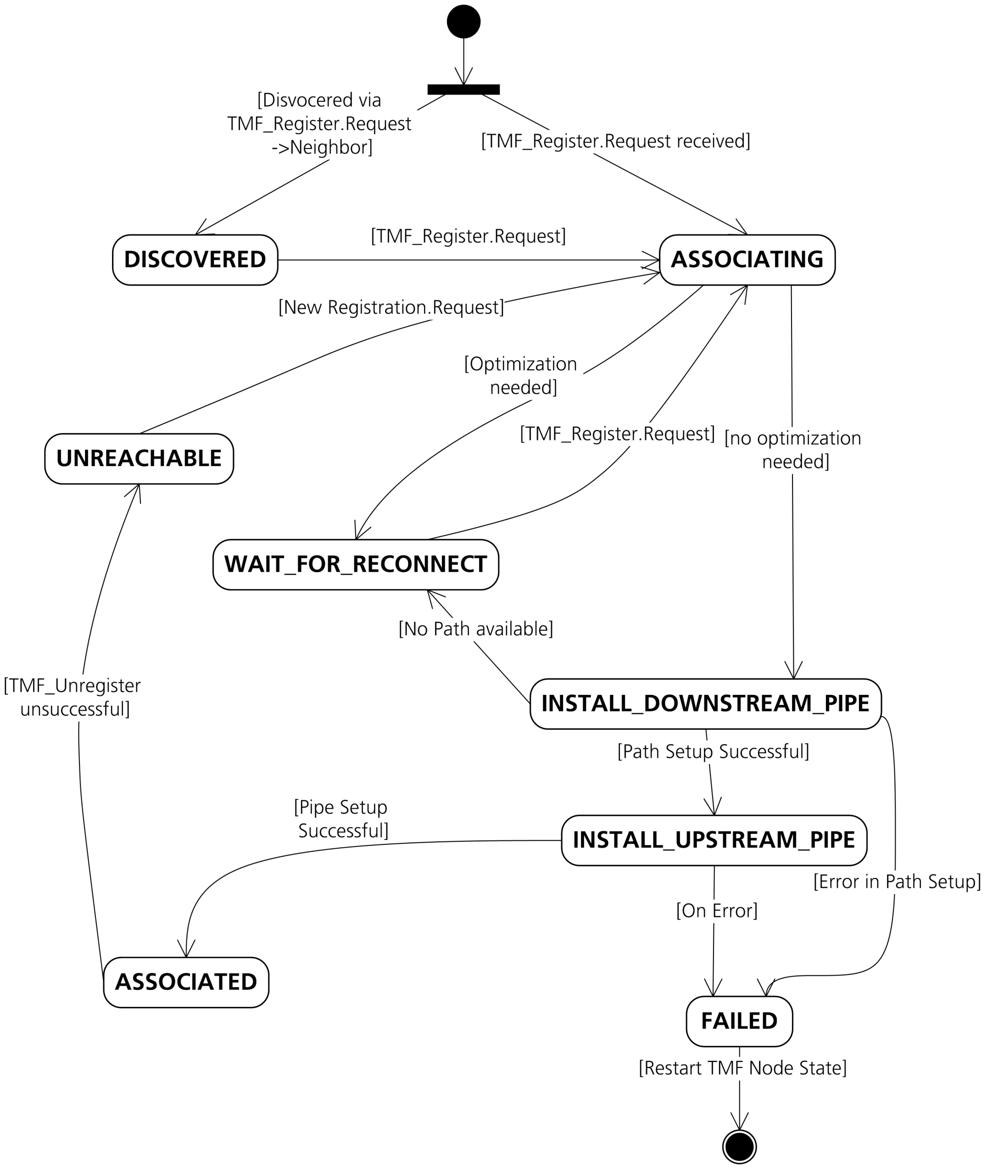

Figure 4.15: Each *Slave* node is managed via its own *SlaveState* object and an associated state machine at the *Master*

configures the optimal channels for its radio interfaces. Finally, it sets up the WiBACK beacon which in turn starts the topology discovery process.

To manage its *Slaves*, the *Master* maintains a *SlaveState* object for each discovered WN which keeps track of the WN's state. New *SlaveStates* are either directly created upon the reception of a *TMF_LinkRegister* association request from the new WN or indirectly by parsing the neighbor information contained in a *TMF_LinkRegister* request. The state machine of the *SlaveState* object is depicted in Figure 4.15. A *TMF_LinkRegister* request is either sent directly to a *Master*, or, if no direct radio association exists, to an already associated WN. This WN will then act as a proxy and forward the request message to the *Master* through its own management *Pipe*. On the first hop, this initial request is sent via IMF *LinkVector* routing in order to specify the specific link to be used for the association, while regular IMF messaging only allows the destination node to be specified.

Upon reception of a *TMF_LinkRegister* request, the *Master* will set the links used for the association to *ASSIGNED* state so it can be considered for management *Pipe* computation. Since this link was chosen by the *Slave* solely based on its local knowledge, the master may invoke an optimization algorithm to check for possibly more optimal connectivity options for the associating WN, see section 4.6.7. If such options are available, the *Slave*'s state is set to *WAIT_FOR_RECONNECT* and the association request is rejected with a set of either black-listed or white-listed links to indicate to the *Slave* which links to avoid or to choose for future association attempts. Even in white-listed mode, the *Master* should return multiple options, where available, since the *Slave* may not be able to establish bidirectional connectivity on the specified links due to physical layer issues.

The paths of the management *Pipes* may be computed by applying a Dijkstra search on the topology graph where the metric may consider, for example, hop distance, signal quality, low interference or more complex metrics. The TMF first ensures that it can compute a downstream and an upstream path and then triggers the PMF to push the downstream *Pipe* state into the network. Upon successful completion, the TMF triggers the PMF to push the upstream *Pipe* state from the associating WN back to the *Master*. Once the *Pipes* have been successfully set up, the new WN is marked as *ASSOCIATED* in the TMF master's topology representation and a *TMF_Link_Register* response is returned indicating a successful association. If an error occurs during the path setup the association process is aborted and the *Slave*'s state is set to *UNREACHABLE*.

Slave State Machine

During their initialization phase, *Slaves* first detect their own capabilities in terms of number, type and properties of available radio interfaces by calling the AI_RadioGetProperties primitive for each interface reported by the IMF, see Figure 4.16. Then they continuously execute the BeaconScan procedure on all receive-capable interfaces in order to discover potential *Masters* or associated WNs. Upon each completed BeaconScan the *Slave* evaluates the collected information by first filtering out Neighbors with mismatching NetworkIds. If at least one beacon from an associated WN has been detected, the *Slave* will start the *Association Manager* passing it the filtered set of beacons of associated WNs.

The *Association Manager* may sort this set according to its local optimization goal, such as bidirectional links first, best signal quality or shortest hop distance and will then begin with the association procedure starting with the highest rated neighbor by sending a *TMF_Link_Register* request towards the *Master* or an associated WN which then acts as proxy for this association procedure. This request includes a set of neighbors gathered during the BeaconScan period to allow the *Master* to determine the optimal link for the association. To avoid a possible fragmentation of this initial association transaction, the number of neighbors may be limited and a randomly chosen subset of associated and unassociated neighbors is sent, possibly excluding UDLs where they are avoided for management path computation. The complete set, including UDLs, is then pushed towards

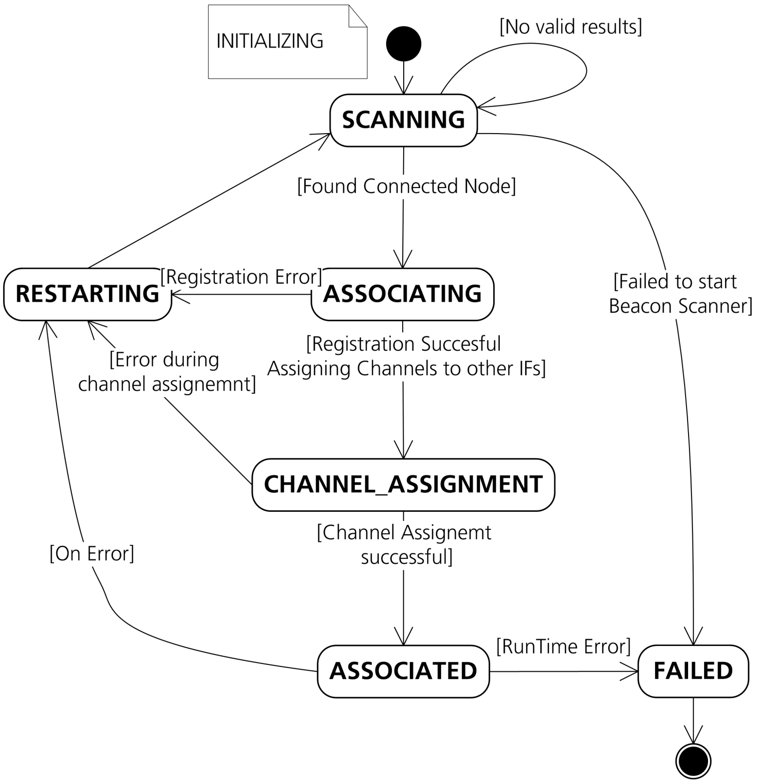

Figure 4.16: TMF *Slave* state machine

the *Master* via a TMF_UpdateNeighbor message once the *Slave* has been successfully associated. This complete set is examined by the *Master* in order to keep the topology information up to date. If UDLs are detected during the association process, the *Master* may make them available to CMF. As described below in Section 4.6.10, they can optionally be used to establish management connectivity, but for most typical use cases this adds extra complexity and possible stability issues especially under non-optimal link conditions.

Next, the *Slave's* additional radios will be configured. The configuration may either be determined and administered locally by e.g. choosing the least utilized channels or remotely by *Master* enforcing a configuration following its network-wide optimization goals. Eventually, the WN enables the transmission of WiBACK beacons and all of its transmit-capable interfaces to allowing other WNs to associate with the WiBACK network.

Associations may fail if either the request or response messages are lost e.g. due to link stability issues or because the *Master* has rejected the request for e.g. administrative reasons, due to *Pipe* setup failures or due to the availability of alternative and more suitable connectivity options. In cases of packet loss or *Pipe* setup errors, the *Slave* should proceed with the next possible link. If the *Master* has rejected the request and provided a set of either black-listed or white-listed links, the *Slave* will start a new association attempt either avoiding the black-listed links or preferring the white-listed links.

In the unlikely event that all attempts avoiding black-listed or using white-listed links fail due to communication errors, the client may force an association to the WiBACK network by setting the *ForceAssociation* flag in the request message. This allows the WN to still join the network, and further optimizations can take place once the WN is associated.

4.6.7 Local Optimization

To validate our TMF framework we have implemented an optimization algorithm which aims at forming point-to-point links among adjacent multi-radio WNs whenever possible. This optimization is performed during the association procedure of a WN by determining the optimal link for the association. Hence, our algorithm performs a *local* optimization, only affecting the associating node, but taking into account the centralized network-wide topology knowledge regarding already associated nodes and their connectivity options. Since the *local* optimization can leverage network-wide knowledge, it can compute a set of optimal candidate links and therefore uses white-listing to indicate its decisions to the *Slave*. This set of candidate links are ordered according to the *Master's* preference. In high connectivity scenarios multiple WNs might compete for a white-listed link, therefore an association might be rejected and an updated list of white-listed links is returned. The parameter R controls how often such a rejection is accepted until the WN forces an association, possible by-passing the optimization and tolerating a suboptimal connection.

This network-wide optimization mechanism affects only the currently associating WN and enables the network to form point-to-point links where available without requiring reconfigurations of established links, thus, for this initial evaluation, avoiding issues such as interruptions of established links or even oscillations of the optimization algorithm.

4.6.8 Failure Recovery

Once a *Slave* has associated with a *Master* both sides start monitoring the management *Pipes* for possible connectivity issues by subscribing to AI_PipeDown indications on their respective egress *Pipe*. Upon an AI_PipeDown event, the *Master* will mark the affected *Slave* as *UNREACHABLE* in its graph. Then a TMF_LinkUnRegister message is sent notifying the *Slave* of the state change in cases where only the *Master* has detected an issue on its EGress *Pipe* due to, for example, asymmetric paths. The affected *Slave*, in turn, will attempt to re-associate with the WiBACK network possibly via alternative links or alternative neighboring WNs.

Any WN connected via the failed WN may experience a loss of connectivity either due to the original cause, for example, management *Pipes* on the same broken link, or due the the failed WN switching back in association mode thus dropping all active *Pipes*. Possible backup paths might keep the other WNs connected.

4.6.9 Analysis

In order to analyze the performance of the TMF's topology forming mechanism we consider an error-free case of a black-out scenario, where all WNs are restarted at roughly the same time. Due to the ring-based approach the duration should mainly depend on the number of logical rings formed during the discovery phase. Assuming a fixed duration for a neighborhood scan t_{Scan}, the lower bound of the discovery time for a d-hop topology can be expressed as follows, with $t_{backoffMin}$ denoting an optional minimum back-off time adhered to by *Slaves* before attempting an association.

$$t_{DiscMin}(d) = d \cdot (t_{Scan} + t_{backoffMin}) \tag{4.2}$$

To determine the upper bound, the exponential back-off function (4.1) as well as the possible impact of the optimization algorithm must be considered. Hence, the upper bound of the discovery time can be expressed as follows, where the factor R covers the possibility of a maximum of R rejections during the association procedure due to optimization which would result in repeated association attempts subject to a new back-off timer period.

$$t_{DiscMax}(d) = d \cdot t_{Scan} + R \cdot \sum_{i=1}^{d} \frac{2^i}{(i+1)^2} \cdot C * 1s \tag{4.3}$$

Since the purpose is to estimate the performance of the basic TMF functionality, the above formulas do not consider computational overheads introduced by possibly complex topology optimization or link calibration algorithms. Accordingly, the *AI_RadioCalibrateLink* primitive was implemented to simply return the logical link parameters reflection the current interface configuration. The signaling overhead of the TMF_LinkReqister primitive and the required PMF signalling of the management *Pipes* has been omitted since it is typically in the order of tens of milliseconds while TMF timing is in the order of seconds.

4.6.10 UDT Support

During the design phase of the TMF we have evaluated two different approaches of UDT involvement, bearing in mind that the main application of UDTs in the WiBACK architecture is to efficiently distribute broadcast content. Hence, in both cases, UDTs should be detected, configured and exposed to CMF.

The involvement of UDTs within the TMF varies depending on the requirement towards physical WN connectivity. If the minimum requirement for a WN is to be equipped with one tx-capable and one rx-capable interface, TMF would have to actively utilize UDTs to establish management connectivity between the *Master* and such *Slaves*. If the minimum requirement is that each WN has to be equipped with, at least, one bidirectional interface, UDTs can be avoided by the TMF for its management connectivity.

In order to support UDTs during the topology forming phase, a detection mechanism is required since transmitting interfaces can not readily detect their possible receivers. Moreover, if the underlying technology is capable of operating in more than one channel, the detection mechanisms would also have to support channel selection. Given that the UDTs considered primarily within the WiBACK context operate in licensed spectrum, this may not readily be possibly or permitted. Especially during the topology forming phase, such uncoordinated transmissions might interfere with regularly operation of transmitters in some channels. Since the WNs in question are not associated at this point, no coordination can be provided by the *Master*.

Hence, to validate our UDT support including nodes meeting only the minimal connectivity requirements, we have made the assumption that all UDTs are operated on the same fixed channel configuration. Under this assumption, a loop detection mechanism similar to the approach proposed by [66] can be used, aiming at detecting *inclusive cycles* to establish bidirectional connectivity between the *Master* and the affected *Slave*.

Loop Detection

WNs without a bidirectional interface or link partner on such interfaces attempt to establish management connectivity following the *inclusive cycle* mechanism [66], which assumes that the unidirectional links are already established. This is a crucial requirement for this mechanism, which will broadcast TMF_LoopFormation messages on tx-only interface assuming that rx-capable interface within its transmission range receive such messages. This mechanisms works as follows:

The *TMF_LoopFormation* extension attempts to form *inclusive cycles*, so-called *loops*, using the TMF_LoopFormation primitive and is triggered when a TMF slave receives WiBACK beacons from associated WNs but does not have a transmit-capable interface to register directly via any of the associated WNs. After a configurable timeout, such WNs start broadcasting TMF_LoopFormation messages to their direct neighbors via all tx-capable interfaces. This broadcast message includes the node information as well as information obtained about its physical connectivity with neighboring WNs determined via the received WiBACK beacons. If such messages are received by other unassociated WNs, they add themselves to the list of unassociated WNs and start broadcasting TMF_LoopFormation messages themselves. This process continues until an associated WN is reached or a configurable maximum of hops is exceeded. An associated WN receiving TMF_LoopFormation messages forwards the contained information to its TMF *Master* which adds all listed WNs and their possible physical connectivity to its topology graph. The *Master* then checks for each of those WNs if paths for management *Pipes* can be computed. Where possible it triggers the setup of management *Pipes*, and, after a successful *Pipe* setup, marks this WN as associated.

The bottom part of Figure 4.14 depicts a UDT scenario involving the WNs F, G and H where a *loop* can be formed allowing all affected nodes to establish bidirectional management connectivity with the *Master*. WN E, on the other hand, can receive WiBACK beacons from the associated WN C on the inner ring, but can not establish a *loop* since no additional WNs are within its range.

Integration Considerations

In [6] we have presented validation results showing that the *TMF_LoopFormation* extension successfully detects loop and utilizes UDTs to establish management connectivity.

Depending on the scenario, a *loop* might consist of a larger number of WN. Packet loss on any segment would impact the detection procedure which, due to the limited connectivity, can only rely on retransmissions to improve the resilience of the detection mechanism. Also, the neighbor information contained in the TMF_LoopFormation messages might grow significantly, i.e. exponentially, with an increasing number of hops, quickly exceeding typical MTU limits. This would cause those messages to be fragmented, thus increasing the susceptibility to corruption or loss.

Taking into account the above mentioned issues regarding channel selection, the default setting for TMF is to avoid UDTs for management traffic and solely dedicate them for data *Pipes* at the CMF's discretion.

4.7 Summary

In this chapter we have presented our design to integrate UDTs into the WiBACK architecture where the signalling operates *below the network layer*. The literature review in Chapter 2 has highlighted the shortcomings of traditional protocols and mechanisms to support UDTs in wireless back-haul networks. The main aspects to be addressed were interface property description, monitoring, *Pipe* signalling as well as topology forming and maintenance.

Addressing those shortcomings which are reflected by our research questions, we have proposed Abstract Interface (AI) extensions to properly detect, describe and configure UDTs providing the foundation for UDT support within the WiBACK architecture. Complementing the lower level support for UDTs, we have proposed a passive receiver-side UDT-aware monitoring framework which also performs LSP monitoring similar to the mechanisms proposed in the recently published RFC 6374 [89].

The WiBACK *TransportService* was enhanced with the *LinkVector* extension to support source routed message forwarding among IMF entities. Building on this mechanism, we have proposed the Pipe Management Protocol (PMP) which provides RSVP-TE-like *Pipe* signalling among IMF entities and support for signalling *around* UDTs while still traversing all nodes in the path. Since the centralized WiBACK architecture does not maintain any routing state among WNs, the PMP is a crucial component to signal management connectivity and to manage regular data *Pipes*.

The *LinkVector* extension is also relied upon by the TMF during the WN association phase. Our proposed TMF can support UDTs with different levels of integration. It may either detect, configure and report them to the CMF for capacity allocation, or it may actively utilize them during the topology forming phase in order to establish management connectivity.

Initial evaluations during the various design phases have indicated that our extended WiBACK architecture properly detects and integrates UDTs. In the following chapter, we validate our work, perform a thorough evaluation and discuss the results.

Chapter 5

Validation and Evaluation

In this chapter we first validate the mechanisms and protocols developed in the previous chapter, verify the proper handling of UDTs and then quantitatively evaluate the main aspects focusing on scalability and robustness.

In Chapter 3 we have described and justified our methodology which relies on quantitative evaluations of designed artifacts. Our artifacts are built upon our SENF-based low-latency NetEMU framework [9] which also provides a real-time network emulator component. This allows us to evaluate the same binary code on emulated or real embedded nodes. For emulated interfaces random packet loss and a fixed link latency can be introduced and the transmission range of an interface is emulated considering the frequency-dependent free space loss as well as a varying transitional zone.

5.1 WiBACK Testbeds and Scenarios

To evaluate the artifacts developed within the framework of this thesis, we mainly rely on emulated testbeds and specifically constructed scenarios in order to evaluate the relevant aspects of our artifacts in controlled environments, while varying the per-link packet loss and latency characteristics. Our main scenario is referred to as the *core-test*, see Figure 5.1, and consists of six nodes. Depending on the specific test case, different interface types, emulated or real, unidirectional or bi-directional, may be configured. Such interface types include IEEE 802.11a, 802.11ah, a DVB-T transmitter and DVB-T receivers. Most functional validations were performed using variants of this testbed.

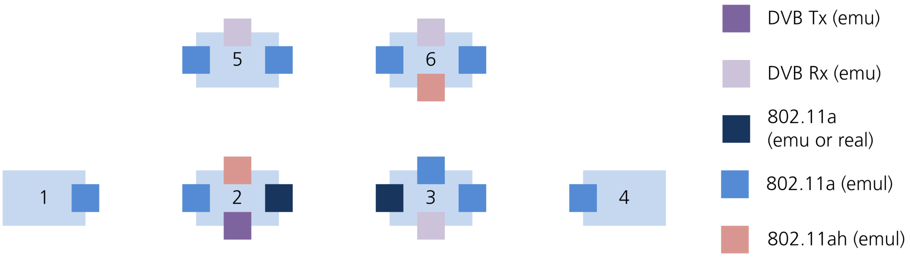

Figure 5.1: The *core-test* scenario was used to validate our TIM, PMF and TMF artifacts

To perform controlled scalability evaluations, we have introduced the so-called benchmark *chain-test*, see Figure 5.2, which consists of eleven nodes, geographically arranged as a linear chain. This scenario therefore allows us the evaluate our artifacts for he maximum WiBACK-considered hop distance of ten. The first and the last node were equipped with one radio each while the intermediate nodes were equipped with two radios each. The transmission range was chosen so that each node could communicate with its one and two hop neighbors. This testbed was used to thoroughly evaluate the PMF as well as the TMF under varying hop count, per-link loss and latency configurations. The optional two-hop connectivity was used to evaluate the *point-to-point link* optimization algorithm of the TMF which should always form a *chain* scenario avoiding the longer range two-hop links.

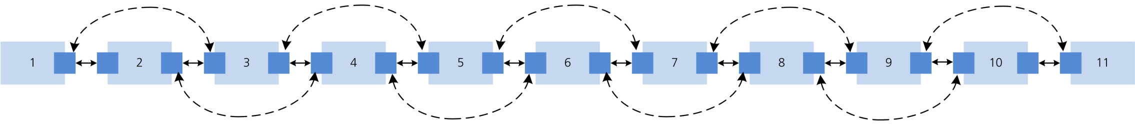

Figure 5.2: The benchmark *chain-test* scenario was used to evaluate the PMF and the TMF

As of the time of writing we also operate three outdoor testbeds at or around the Fraunhofer Campus in Sankt Augustin, Germany. Here individual aspects, but mainly complete WiBACK deployment scenarios are evaluated which always includes the core functionality developed within the framework of this thesis. These testbeds have been setup to test our software under real world conditions or to address specific aspects such as channel assignment and separation in dense deployment scenarios, the stability of 10km long-distance links or to validate a possible pilot installation in Maseru, Lesotho [12] (Figure 5.3). The testbeds have been used in the scope of this study to verify our emulated testbed results on real hardware.

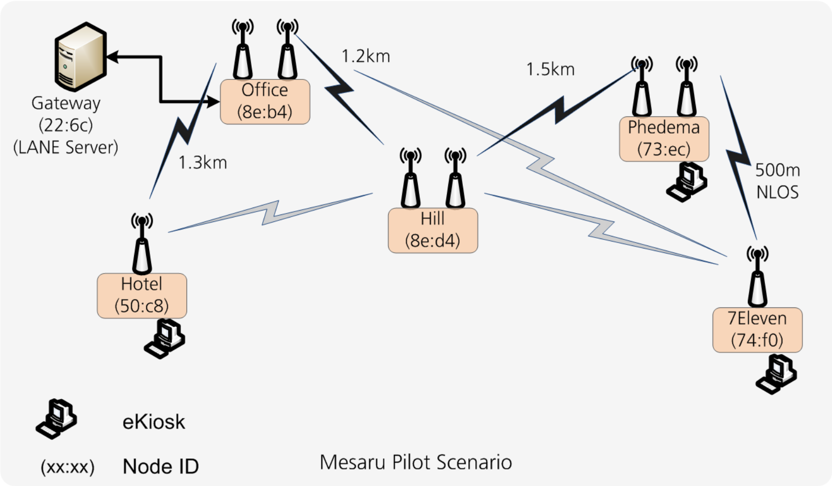

Figure 5.3: The initial pilot in Maseru, Lesotho consists of five outdoor WiBACK nodes and one indoor node acting as the WiBACK and eKiosk management node.

5.2 Goals and Assessment Metrics

The goal of our validation and evaluation efforts was to validate proper UDT support and to establish a quantitative assessment of the performance of our proposed protocol and algorithm designs.

The overall UDT support of our contributions was validated with the PMF and TMF validation scenarios which cover all our contributions and require proper UDT handling to detect, configure and signal via UDTs.

The performance of the PMP *Adjustable Reliability Mechanism* will thoroughly be evaluated under varying loss and latency patterns in order to establish the typical success rate under such conditions and to investigate the impact of higher loss ratios at different segments in the path. From the results we also derive a suggested setupTimeLimit for the hard-state PMP to achieve reasonable success rates, i.e. $\geq 95\%$, under typical loss and latency conditions.

The performance of the ring-based TMF and our exemplary *point-to-point* link optimization algorithm will be evaluated using different scenarios. The results will compared against the theoretical bounds expressed in Chapter 4.

5.3 Abstract Interface Extension

The AI and, in particular, our proposed extensions have been validated via unit tests to ensure their proper operation and to support automated testing if underlying components are extended or modified. All relevant AI primitives listed in Section 2.1.4 will be implicitly validated during the PMF and especially TMF validation, since both modules rely on the AI to manage the affected interfaces.

5.4 LinkVector Extension

The *LinkVector* extension to support source-routed messaging has been integrated into the WiBACK *Transport Service*. Unit tests have been written to validate its functionality and to support automated testing if related components are extended or modified. The *LinkVector* extension will be implicitly validated during the PMF and TMF validation, since both modules rely on this communication mechanism.

5.5 Technology Independent Monitoring

We have implemented the major components of the WiBACK monitoring modules and present initial results confirming TIM delay and loss measurements as well as a validation of the event creation via *Rating Agents* indicating an under-performing link.

The scenario depicted in Figure 5.4 is based on the *core-test* and consists of four emulated nodes as well as two Linux PCs running the *mgen*[1] traffic generator. The links between nodes 1 and 2 and nodes 3 and 4 are emulated, while the link between nodes 2 and 3 was made up of two real IEEE 802.11a interfaces in ad-hoc mode. To compare the TIM measurement results against the *mgen* results, we have introduced distinct fixed latencies of 5ms and 2ms respectively as well as an average loss of 2% on the emulated links. Three different flows are sent via separate LSPs, a 64kbps VoIP flow, a 2Mbps video flow and an ICMP flow created by a *flood-ping*.

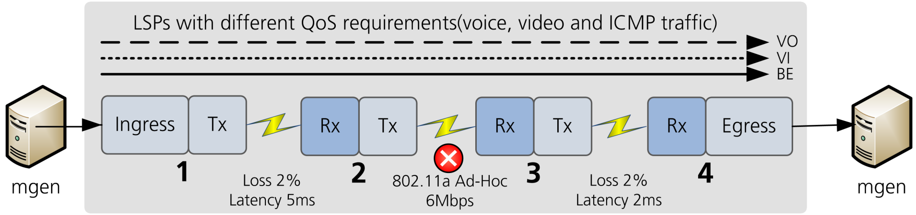

Figure 5.4: The TIM evaluation scenario consists of 4 nodes with two emulated and one real 802.11a link

5.5.1 LSP Statistics

The upper part of Figure 5.5 depicts the TIM delay measurements of the video LSP obtained at the intermediate MPLS nodes as well as end-to-end, while the lower part depicts the corresponding loss figures. The result for the VoIP and ICMP LSP are similar and have been omitted to avoid cluttering the graph.

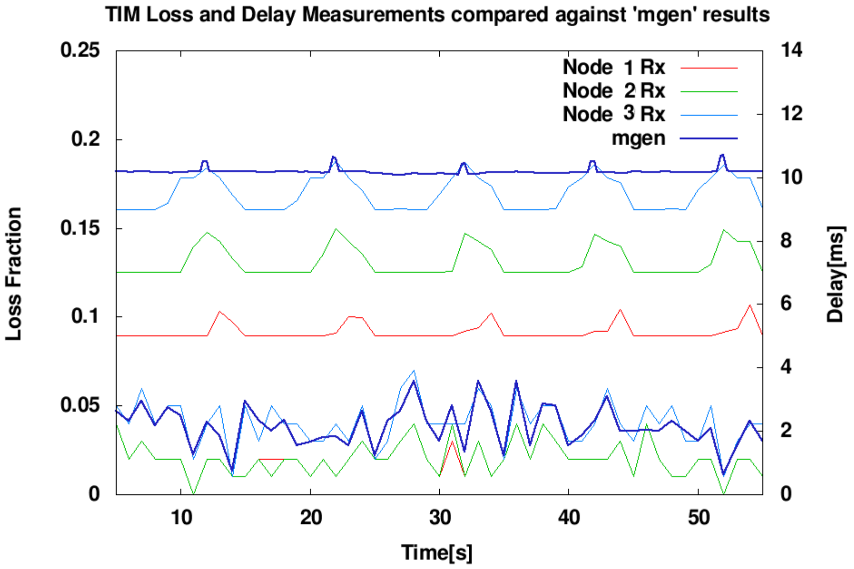

Figure 5.5: TIM receiver side measurements at nodes 2, 3 and 4 compared to end-to-end *mgen* results

---

[1] http://cs.itd.nrl.navy.mil/work/mgen

The graph shows that TIM in clock-synchronized mode accurately captures loss and delay introduced by the emulated links AB and CD, as well as a 2ms delay introduced by the almost loss-free 802.11a link. The TIM end-to-end measurements represented by the thin blue lines are confirmed by the *mgen* results represented by the thick blue lines. Note that *mgen* was run on external PCs connected via Ethernet. Hence the average latency measured is slightly higher.

5.5.2 LSP Inactivity Event

Figure 5.6 depicts the inactivity per LSP on the 802.11a link. The two thick lines represent the period between the last activity seen and the time when the *Rating Agents* have created inactivity events, which may result in AI_PipeDown.Indications to be sent. Different per-LSP inactivity thresholds have been configured: 100ms for the VoIP flow, 1000 ms for the video flow and 5000ms for the *best effort* IGMP flow.

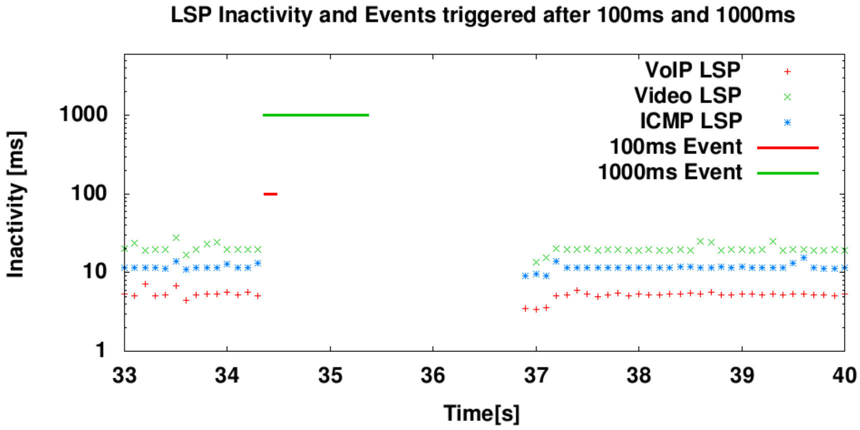

Figure 5.6: Inactivity and triggered events on a per-LSP basis

After we disabled the 802.11a link at time t=34s, the two inactivity events for the VoIP and the video flow have been triggered after 100ms and 1000ms respectively. We re-enabled the link right after these events, and regular activity has resumed after about 3000ms, hence the third event with a timeout of 5000ms did not get triggered.

5.6 Pipe Signalling

The PMF has been implemented as an IMF User Module according to the specifications provided in this thesis. At each hop, PMP negotiates the downstream or upstream MPLS labels, installs the corresponding LSP state, reserves the requested *Pipe* resources with the MAC Adaptor of the respective outgoing interface and determines the end-to-end MTU. If an error occurs during this procedure, the PMP reports the NodeId of the first node detecting the error back to the ingress node.

As stated in Chapter 4, especially the PMP artifact has been refined multiple times to incorporate new findings learned during the implementation phase. During this process the adjustable reliability mechanism has been significantly improved and the IMF-related signalling aspects, such as the limited number of transactions, have been addressed.

5.6.1 Functional Validation

As a functional validation, we verified the basic PMP functionality, the a) setup of a *Pipe* around a UDT, and the b) setup of an upstream-assigned multicast *Pipe*. Both tests were run using the *core-test* scenario, where the UDT connectivity was provided by an emulated DVB transmitter and an emulated DVB receiver interface respectively. A shell script was used to query the *LinkIds* from the emulated nodes. Using those *LinkIds* the paths of the *Pipes* have been determined and PMF was triggered to set up and tear down the respective *Pipe*. To verify the proper *Pipe* setup, ICMP echo requests were sent through the *Pipe* using the *ping* command. At the egress node, the number of received packets was counted to verify proper forwarding. The output of this script is shown below:

```
WiBACK (build date 20111021-1906CET)

        PMP Test topology

             5<--->6
            /       \
     1<--->2---UDT-->3<--->4

* Unicast UDT Pipe Setup (1 -> 2 -> 3 -> 4): OK
* Multicast Pipe Setup (1 -> 2 -> 5 -> 6 -> 3 -> 4): OK
* Performing unicast ping test with 200 packets
  at a rate of 10 pkt/s on unicast pipe: PASSED
* Performing multicast ping test with 200 packets
  at a rate of 10 pkt/s on multicast pipe: PASSED
* Unicast Pipe teardown: OK
* Multicast Pipe teardown: OK
```

5.6.2 Robustness Evaluation

Following the initial successful validation, we evaluated the PMP performance under typical packet loss and link latency conditions. For those tests, the *chain-scenario* with direct point-to-point connectivity was used. For each hop, an independent channel was assigned to avoid interferences.

First, an initial measurement was run to determine a reasonable default setupTimeLimit that fits a typical heterogeneous, i.e 802.11, 802.16, DVB-T, WiBACK scenario. Our criterion was that 95% of all setup attempts should succeed up to a conservative As stated in Chapter 4, especially the PMP artifact has been refined multiple times to incorporate new findings learned during the implementation phase. During this process the adjustable reliability mechanism has been improved and the IMF-related signalling has aspects, such as the limited number of transactions, have been addressed.maximum per-hop latency of 50ms which covers typical IEEE 802.16 latencies of about 30ms as well as DVB-T latencies of about 20ms. The maximum per-link error rate was assumed to be 10%, about an order of magnitude higher than the loss expected on stable WiBACK links. Higher loss figure may occur during the topology forming phase, where links might not be calibrated.

For a typical WiBACK 802.11-based scenario in our outdoor testbeds, the maximum link latency in the *Management* traffic class, even under heavily loaded link conditions, has been determined to be roughly 2 ms. Hence, given our eleven-hop *chain-scenario* scenario, this would result in a round trip signalling time of $10 \cdot 2 \cdot 2ms = 40ms$. To leave some headroom for a fast successful completion without retransmissions, the initial retransmission timeout was set to 50ms and the setupTimeLimit was determined to be about 2000ms for this parameterization assuming the 95% success rate target.

For the following measurements, 1000 *Pipes* had been established for each combination of loss rate and link latency. The measured setup times have been reported in the form of a {min,avg,max,std-dev,success rate} tuple.

Figure 5.7 depicts the *Pipe* setup time over an increasing link latency. The measurements were run for different per-link loss figures ranging from 0%, 1%, 3%, 5%, 10% up to 20%.

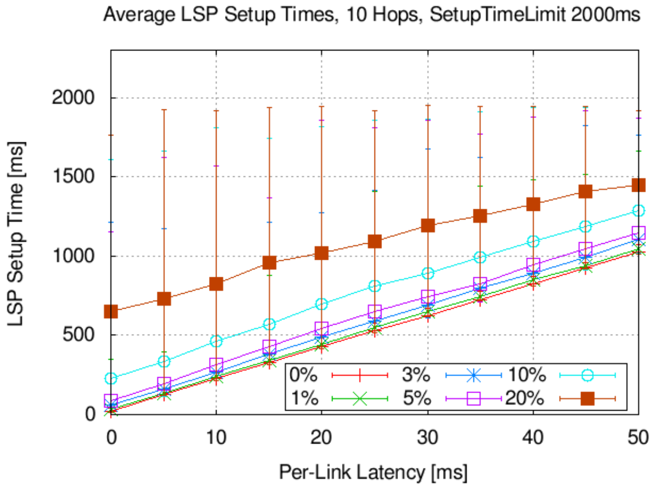

Figure 5.7: *Pipe* setup time over per-link latency under varying per-link loss fractions

As expected, the link setup times increase linearly with an increasing link latency. Packet loss on average only causes a minimal increase of the setup times, while the upper bound is capped at 2000ms due to the predetermined limit. Figure 5.8, which depicts the success rate of the same set of measurements, shows that up to a per-link loss of up to 5% all *Pipes* could be established within the 2000ms limit. Even with 10% per-link loss rate, the target success rate of 95% was achieved, except for relatively high per-link latency figures. For 20% per-link loss, the success rate drops significantly. The success rate could be improved by TMF adapting the PMP parameters, but since such a scenario is not typical, the results here have only been reported for completeness reasons.

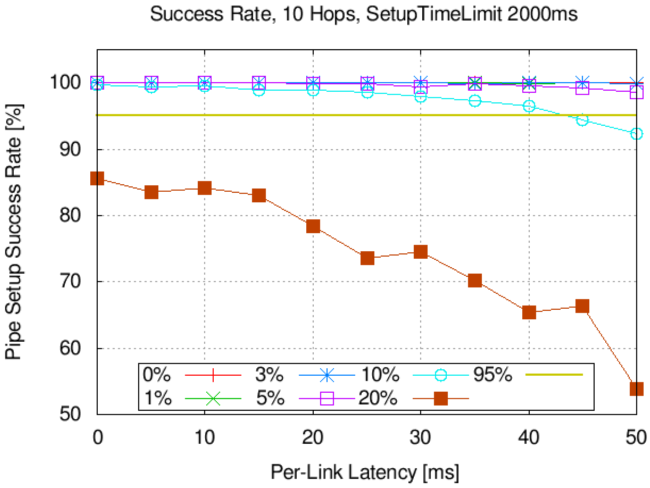

Figure 5.8: *Pipe* setup success rate over per-link latency under varying per-link loss fractions

Figure 5.9 depicts the results of a typical WiBACK scenario where most links are considered relative stable and almost loss free while one link might be experiencing high packet loss. Since PMF uses nested *Request/Response* messages, we analyzed the dependency of the setup time and success rate on the distance of the faulty link from the ingress node. The link latency for all hops has been set to 2ms and the packet loss rate for the stable links has been set to 0%, while the loss rate for the hop to be examined is varied from 0% up to 50%.

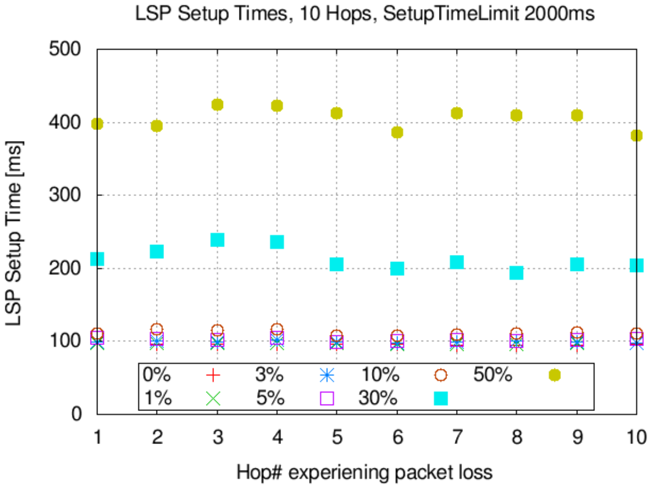

Figure 5.9: *Pipe* setup time over hop distance of errored link

The results show that the position of the faulty link in the chain has no significant impact on neither the average setup time nor the success rate. The results also show that with the chosen default parameterization, a single hop loss probability of 30% can be tolerated with a success rate of almost 100% resulting in a fast average setup time of about 200ms. Even assuming 50% loss, the success rates are still above 80% which might

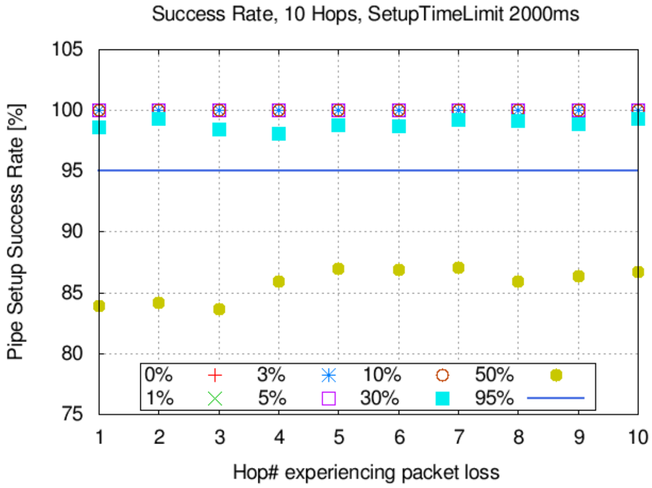

Figure 5.10: *Pipe* setup success rate over hop distance of errored link

require multiple setup/teardown attempts but still allows TMF to reach the affected node to, for example, trigger corrective actions.

5.7 Topology Forming

TMF has been implemented as a IMF User Module. The topology representation is based on the C++ boost graph library [2] as a *directed multi graph*. The boost graph library allows for graphs to be easily serialized into *dot* or *GraphML* format in order to visualize the discovered topologies as well as their properties.

The TMF artifact has been refined multiple times to incorporate new findings learned during the implementation phase. During this process the support for UDTs has been improved and the *point-to-point* link optimization algorithm was refined which included multiple refinements of the beacon scan procedure to most reliably detect the possible physical connectivity among WNs.

5.7.1 Functional Validation

We validated the basic TMF functionality, the discovery and optimized forming of a topology based on the *core-test* scenario, where node 2 was equipped with a DVB-T transmitter and the nodes 3, 5 and 6 were equipped with DVB-T receivers. Bidirectional connectivity was provided by IEEE 802.11a interfaces. For all interfaces, omni-directional antennas were assumed. The TxPower settings where fixed and have been chosen to exhibit *FLAKY* links. As depicted in Figure 5.11, the *Master* located at node 1 has successfully discovered and associated all *Slaves* and formed a broadcast cell among the detected DVB-T interfaces. The discovery process took 62 seconds to complete and 89 possible links have been discovered of which 22 where marked as *FLAKY*[3], among them the potential link

---

[2] http://www.boost.org

[3] see Section 4.6.2

d1:37 ↔ ff:c1. The affected radios itself have been reconfigured to establish connectivity with nodes 5 and 3 respectively. The center frequency for the DVB-T cell was fixed at 714MHz, while the frequencies for the IEEE 802.11a links have been assigned by the TMF, maintaining, at least, a 60Mhz separation between the center frequencies.

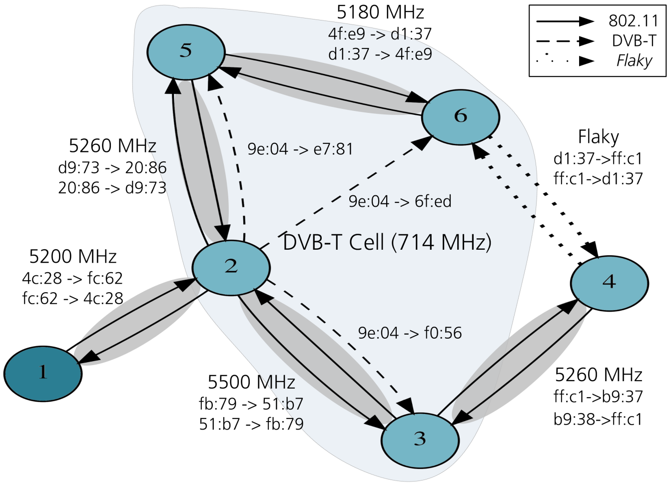

Figure 5.11: The *dot* output of the graph maintained by the TMF master at node 1 shows the discovered and optimized topology including the DVB-T cell and *FLAKY* links.

The above results confirm earlier evaluations performed on real Linux-based multi-radio nodes validating an IEEE 802.11-based pilot deployment scenario to be located in Maseru, Lesotho [12]. Figures 5.3 depicts the expected physical connectivity for this scenario which was set up at the Fraunhofer campus in Sankt Augustin, Germany. To create similar physical connectivity, directional antennas were used and hand-adjusted. More details regarding this scenario and its evaluations can be found in [12]. In the context of this study, it is important to note that the TMF *Master* at the *Gateway* node has configured point-to-point links while maintaining a channel separation of 60MHz among local interfaces, see Figure 5.12.

5.7.2 Scalability Evaluation

In order to evaluate the scalability of our ring-based TMF design we considered black-out situations as the worst case and performed the evaluation for three different scenarios. The *chain-test* scenario was used to represent a minimum-connectivity scenario. Additionally, we randomly generated two 100-node scenarios with dense and sparse inter-node connectivity, with the *sparse* scenario resembling a typical WiBACK scenario most closely.

For the above described scenarios we first evaluated the TMF topology forming duration per logical-ring, which should be in between the bounds expressed by the terms (4.2) and (4.3) defined in the previous section. Since the TMF relies on the IMF and its *AckService* for reliable messaging and as well as PMF for path setup signalling, the obtained results reflect the performance of the combined modules. For the below measurements the following parameterization has been used: $C = 6s$, $R = 2$, $t_{backoffMin} = 500ms$, $t_{Scan} = 5s$

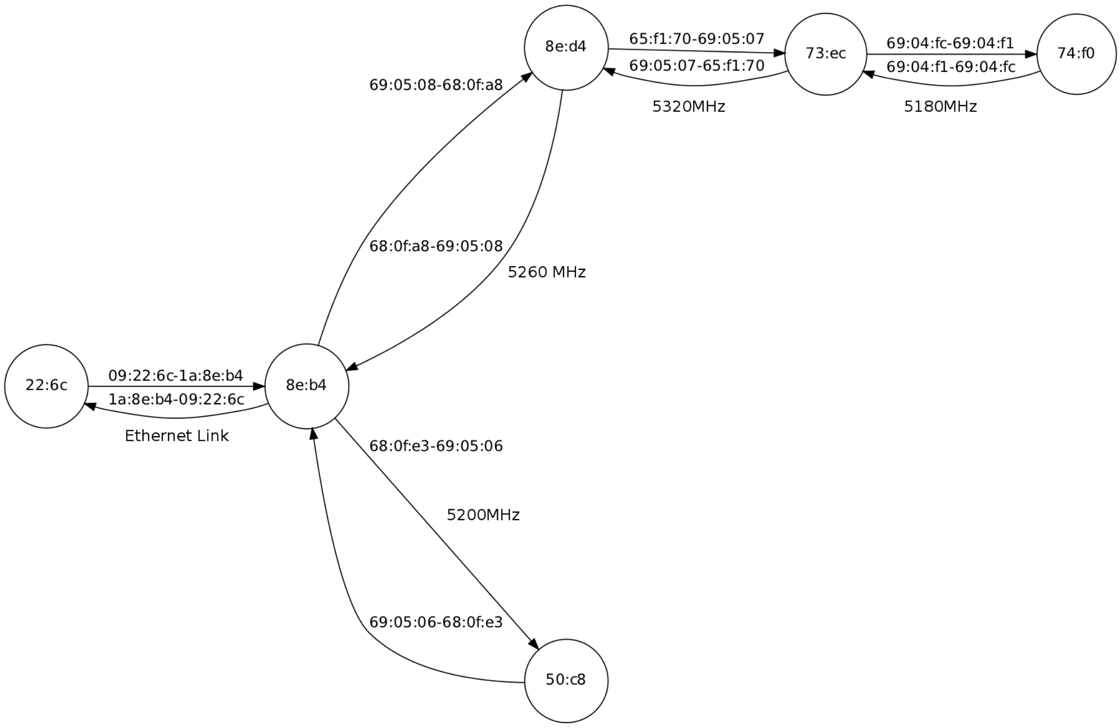

Figure 5.12: The TMF *Master* at node 22:6c (*Gateway*) has successfully discovered and configured the network by forming point-to-point links and ensuring at least 60Mhz channel separation among local interfaces.

based on a WiBACK beacon interval of 250ms and eight available IEEE 802.11a channels. LINK_DOWN detection threshold was set to 5s. All measurements were run 50 times and the averages are shown.

It can be observed in Figure 5.13 that the topology forming time is within the theoretical bounds for all scenarios and we expect that the number of WNs to be joined per ring has no significant impact on the per-ring discovery times as long as the *Master* node or the links in its vicinity are not overloaded. For the *chain* scenario the *local* optimization formed point-to-point links for all hops, while for the *sparse* and *dense* scenarios about 5% of all *ASSIGNED* links could not be optimized either due to timing issues among competing WNs within the allowed rejection count R or due to limited connectivity options.

To evaluate the topology forming phase under varying link error conditions, we have chosen the *sparse* scenario as an example for a WiBACK scenario. Figure 5.14 depicts the results for one, five and ten percent of per-link packet loss, which may result in up to 10%, 50% or 100% of end-to-end packet over 10 hops. It can be observed that the WiBACK control plane is relatively robust against these rather high loss figures with the total discovery times only increasing moderately, which shows that the *AckService* as well as the PMF can cope well with the rather even loss distribution introduced by the emulation. We suspect that larger burst losses, i.e. short link outages, would have a stronger impact. Investigating the resilience of the WiBACK control plane is ongoing work.

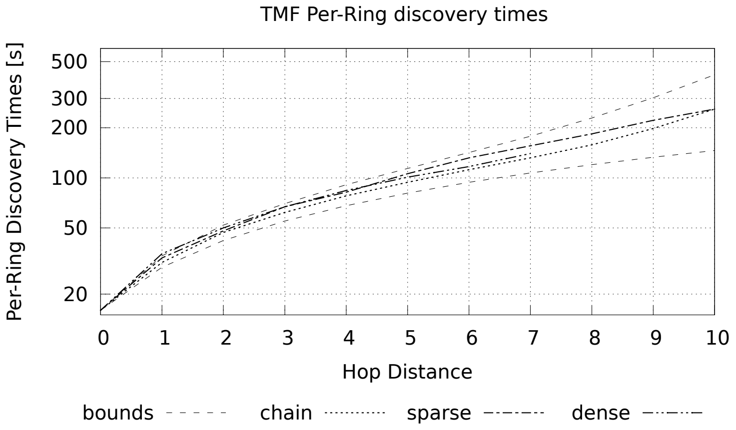

Figure 5.13: Per-ring topology forming times for the *chain*, *dense* and *sparse* scenarios with up to ten hops

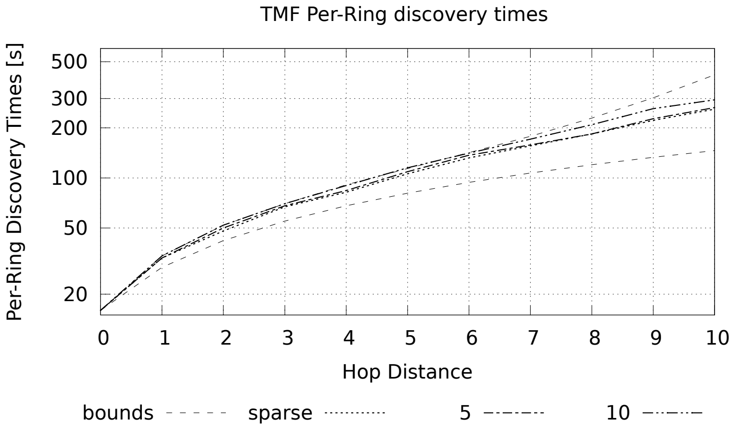

Figure 5.14: Per-ring topology forming times for the *sparse* scenario under 0%, 5% and 10% per-link error conditions

Next, we have evaluated the recovery times for the three scenarios in cases of node failures, where the complete topology has been discovered and then the hop distance d of the failed node from the *Master* was varied from one to ten. Figure 5.15 depicts the results for the *chain* scenario which yielded constant results. It can be observed that for failures of the first hop node, the failure detection and recovery time is roughly identical to a complete topology discovery time determined above. For nodes farther away from the *Master*, the discovery time decreases. Hence, the recovery procedure also performs within deterministic bounds. The results for the *dense* and *sparse* scenarios varied significantly due to the randomness of the formed topologies and the number of nodes affected by a node failure and have therefore been omitted. It should be noted that the maximum discovery time was still bound.

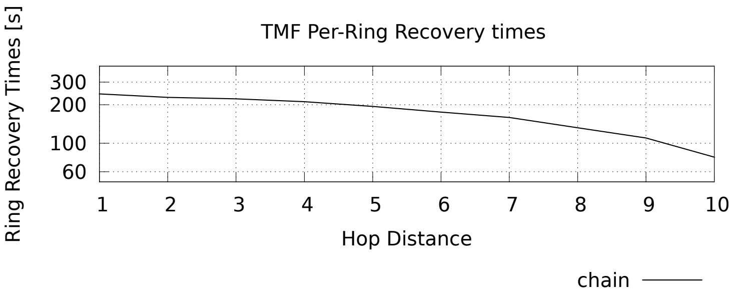

Figure 5.15: Recovery times after failure of a d-hop node for the benchmark *chain* scenario

5.7.3 Outdoor Testbed Results

In order to evaluate our TMF approach on real hardware we have deployed the same code on a dense seven-node outdoor testbed. This testbed is located on the roof of an office building with an area of roughly 20m x 50m in size. Six nodes are actual low-power outdoor mesh nodes with two IEEE 802.11a radios and an omni-directional antenna each. The *Master* is hosted on a dedicated server and is connected to one of the mesh nodes via a 100Mbps Ethernet link. The roof is cluttered with air conditioning units, satellite dishes and other equipment. Hence, not all nodes have line-of-sight connectivity and reflections, etc. are to be expected.

Figure 5.16 depicts snapshots of the *TMFGraph* visualization tool. On the left, all possible physical links are shown, while on the right only the assigned links are shown together with the chosen center frequency. The two links with a center frequency of 0 MHz refer to the Ethernet cable between the *Master* the WN. It can be observed, that, due to the current link conditions, the TMF has mostly configured independent point-to-point links while maintaining a node-local channel separation of, at least, 60 MHz. Our future work item, a network-wide optimization mechanism, would over time attempt to reconfigure the non-independent links, in this case the link between nodes 61:bc and 61:38, to form an interference-free topology.

5.8 Summary

We have successfully validated our approach for UDT integration into the WiBACK architecture. Results obtained in emulated and real testbeds show that our proposed mechanisms support bidirectional and well as unidirectional technologies.

Considering the research questions defined in Chapter 1, we have implicitly validated our proposal regarding the identification and description of UDTs within the Abstract Interface (AI) through the evaluation of the PMF and TMF modules. Hence, the WiBACK architecture can clearly distinguish between natively unidirectional technologies and malfunctioning bidirectional, mostly *ad-hoc*, technologies.

Our proposed Technology Independent Monitoring (TIM) extension for the monitoring component has been shown to accurately capture the loss and latency of *Pipes* using passive receiver-side monitoring, which has been verified with the external *mgen* tool. Likewise, we have shown that the monitoring component can trigger AI_PipeDown.Indications dependent on the individual *Pipe* QoS requirements.

Building on the above functionality, we have throughly evaluated the PMF module and shown that the proposed Pipe Management Protocol (PMP) supports the setup of *Pipes* with either upstream or downstream assigned labels while signalling around UDTs. The PMP is designed following proven RSVP-TE concepts, but relies on the IEEE 802.21-inspired IMF for its signalling purposes. While this assures technologies independence, it required the introduction of the so-called *LinkVector* extension, to support explicit hop-by-hop message passing. RSVP-TE and PMP can not be compared directly, since they address different requirements. RSVP or RSVP-TE are typically used to signal flows or LSPs in already operational networks, while PMP must support the signalling of initial multi-hop management connectivity among WNs in a loss-impacted environment. Hence, the hard-state PMP implements a rather aggressive acknowledgement mechanism to quickly and reliably determine if a *Pipe* setup was successful. Conceptually, depending on the parameterization, PMP should perform equally to RSVP-TE using its *MessageId* extension.

Utilizing the PMP to setup communication paths, so-called *Management Pipes*, among WNs we have evaluated our proposed Topology Management Function (TMF) framework and the related Abstract Interface (AI) primitives. It could be shown, that the TMF and the exemplary optimization algorithm were able to reliably form optimized topologies out of a given set of nodes and their wireless interfaces. UDTs where either considered during the initial connectivity setup phase, or detected after a successful association of a WN and then made available to the Capacity Management Function (CMF) for capacity allocations. As stated in Chapter 2, to the best of our knowledge no comparable architectural approach to support heterogeneous MR-WMNs including UDTs has been proposed and evaluated.

A comparison of our results against proposed IEEE 802.11-specific MR-WMNs channel assignment approaches under quantitative aspects is not possible, since such approaches rather focus on the optimization algorithms while for our work, the exemplary point-to-point link optimization was mainly a tool to evaluate our architectural support for such spectrum allocation algorithms.

Our exemplary ring-based approach to form point-to-point links could be considered related to the IEEE 802.11 *Infrastructure Mode*-based work described in [80]. In both cases, the resulting topology may be a tree, rooted at a *gateway* node. However, the authors in [80] to do not consider link calibrations or capacity management, and their results focus on mobility support or ARP optimizations, while we have evaluated the initial topology forming times, as well as time it took to discover and recover from node failures.

It should be noted that the TMF recovery evaluation has implicitly covered a combination of all our contributions of this study: interface descriptions, monitoring and event creation, pipe signalling under loss conditions and topology forming and maintenance.

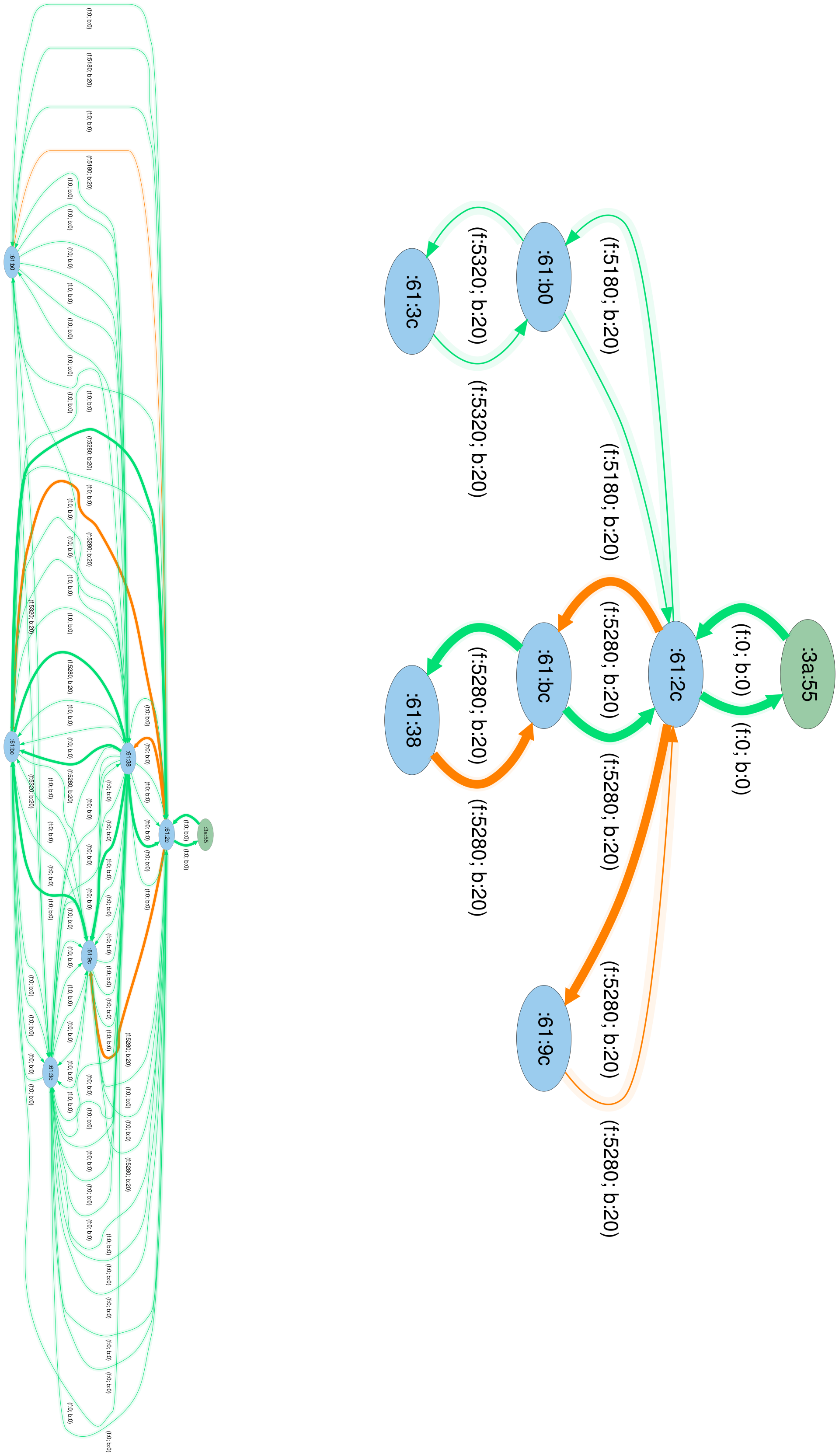

Figure 5.16: Live snapshots of our 7-node two-radio outdoor testbed showing all possible links (left) and the subset of assigned links (right). The orange color of some links indicates that minor link quality issues have been reported.

Chapter 6

Conclusion

The aim of the work reported in this thesis was *'To describe and integrate UDTs into the WiBACK architecture so that spectrum or capacity optimization algorithms can consider and utilize them'*. We have shown that this aim could be broken down into four distinctive research questions:

- How to identify and describe UDTs?
- How to perform monitoring on UDTs?
- How to signal paths in the presence of UDTs?
- How to include UDTs into Topology Management?

During the course of this study, we have studied prior work on related issues, refined and eventually presented our design and evaluated our artifacts.

6.1 Research Contributions

Summarizing our study, we can state that we have successfully integrated UDTs into the WiBACK architecture, allowing the TMF and the CMF to describe, configure and utilize UDTs when considered advantageous according to their network-wide optimization goals. In the following, we highlight how our contributions answer the research questions defined in Chapter 1.

6.1.1 How to identify and describe UDTs?

To describe UDTs within the Abstract Interface (AI), we have introduced the notion of natively unidirectional technologies and malfunctioning bi-directional technologies. Hence, contrary to existing WMN or MANET protocols, our extended AI and TMF components can describe and differentiate broken or misconfigured links from links provided by a natively unidirectional technology.

6.1.2 How to perform monitoring on UDTs?

We have investigated how the CARMEN monitoring subsystem could support UDTs. Since WiBACK follows proven Traffic Engineering (TE) concepts and considers its resources unidirectionally, we designed a passive receiver-side monitoring module, where link and *Pipe* statistics are gathered and analyzed on the respective receiving end. We have shown that, based on those statistics, the relevant events, such as AI_LinkDown, AI_PipeDown, AI_RegulatoryEvent can be created. Moreover, the architecture would also support predictive events, such as a AI_LinkGoingDown. Hence, our proposed monitoring subsystem can readily consider UDTs.

6.1.3 How to signal paths in the presence of UDTs?

We have introduced the *LinkVector* extension for the *TransportService* to support message forwarding along source routed path without requiring any further address lookups or routing state on intermediate nodes. While this extension was mainly introduced to support an RSVP-TE style LSP signalling, in particular across UDTs, it also proved essential during the topology forming phase, where a specific link out of a set of possible links towards the same destination was to be used for the initial association. A crucial requirement not previously supported by the CARMEN-defined IMF. Another aspect relying on this extension may be the Fast Reroute (FRR) signalling of the PMF, which may choose to either set up and maintain stateful *Management Pipes* towards a PLR, or to store source routes for the FRR indication messages. Thus, this extension is crucial to signal within a network without routing state or when specific links or path must be followed.

Building on the *LinkVector* extension, we have designed the Pipe Management Protocol (PMP) which is executed by PMF instances at each WN to provide RSVP-TE-style *Pipe* signalling. PMP provides a *PathVector* object which can be compared to the Explicit Route Object (ERO) in RSVP-TE. Contrary to RSVP-TE a second *PathVector* object may be present to describe an alternative return path to, for example, allow signalling around UDTs while traversing all nodes on the downstream path. To support reliable *Pipe* signalling even under higher frame loss conditions, PMP provides a configurable hop-by-hop retransmission and acknowledgement mechanism which was inspired by the RSVP acknowledgement extension, while taking into account the peculiarities of the IMF transaction paradigm. PMP has been shown to quickly signal *Pipes*, even under high loss conditions and is a crucial protocol within the WiBACK architecture which almost exclusively relies on *Management Pipes* between *Master* and *Slave* nodes for its signalling purposes. Moreover, PMP building upon the *LinkVector* extension provides UDT support without requiring any further routing protocol or address lookup mechanisms.

6.1.4 How to include UDTs into Topology Management?

The Topology Management Function (TMF) is tasked to discover and form topologies among participating WNs, while the CMF assigns resources of activated logical links to *Pipes* or multicast *Trees*. The CMF was found not to require any modifications in order to be able to consider UDTs or their resources for path computations, since it considers its logical resources unidirectionally. The TMF, however, must address UDTs by, at least, detecting and configuring them for the CMF to utilize their capacity. We have shown that, optionally, the TMF may also utilize UDTs to establish management *Pipes*. Moreover, the TMF must distinguish between links provided by natively unidirectional technologies and links provided by broken or misconfigured bidirectional *ad-hoc* technologies. Such links are marked as *FLAKY* by the TMF and may be reconfigured or disabled.

6.2 Further Considerations

The explicit consideration of unidirectionality in all relevant aspects of the WiBACK architecture has improved the flexibility of the overall architecture which can now readily support extensions such as the IEEE 802.11e *NOACK* policy, which may significantly improve the throughput over long distance links. From the point of view of the standard or even advanced IEEE 802.11 rate adaptation algorithms, such a link has become unidirectional or even *broken*, since such algorithms expect an ACK frame for their feedback-based adaptation. The WiBACK architecture can readily support such a setup by adjusting the link parameters via the AI_RadioCalibrateLink primitive and by relying on passive receiver-side monitoring to provide an ACK-free feedback loop in order to create events in cases of link deterioration.

The integration of UDTs below the Network layer complements the centralized WiBACK network management approach for back-hauling traffic patterns. Higher layer protocols transparently benefit from the extra capacity provided by UDTs, which are hidden underneath the MPLS LSPs.

6.3 Limitations

The CARMEN-inspired WiBACK architecture addresses specific deployment scenarios where traffic is mainly forwarded from Radio Access Networks (RANs) towards the backbone or the Internet and vice versa. The centralized approach was chosen under the assumption of a rather limited coverage area and the QoS and service availability requirements of an operator back-haul network which had to be met with rather limited wireless resources. As discussed in Section 4.1 we believe that our approach to integrate UDTs below the Network layer provides the most efficient integration into the WiBACK architecture and its specific deployment scenarios and their requirements.

In the following, we discuss possible limitations of the WiBACK architecture focusing on limitations that also impact our proposed UDT integration.

6.3.1 Traffic between User Terminals

The WiBACK architecture and because of that also our UDT integration was designed to support typical back-haul traffic patterns where traffic is mainly forwarded between UTs and the WGW nodes. UT to UT traffic would have to be routed via WGWs or via dedicated *Pipes* among the WNs the UTs are attached to.

Assuming a significant fraction of inter-UT traffic, *Pipes* might have to be established directly among any two WNs forming a WiBACK network. This would avoid routing via WGWs but dramatically increase the number of *Pipes* and therefore the resource computation and allocation complexity in the network.

While this is not a limitation introduced by UDTs as such, an integration at the Network Layer would more readily support the deployment of alternative routing protocols depending on the deployment scenario. As discussed in Section 4.1, such protocols would then require modifications to interface with the TMF and the AI in order to properly support UDTs.

6.3.2 MPLS on the Data Plane

The technologies considered in this study support the of use MPLS on top of their MAC layer implementation, either natively or via MPE or GSE in the DVB case. Supporting virtual links tunneled across an IP-based network would be possible, but would increase the per-packet protocol overhead. Moreover, the monitoring or resource allocation capabilities might be limited.

Supporting the distribution of broadcast TV content via the WiBACK architecture would require traditional set-top boxes to handle MPE and MPLS instead of plain MPEG transport streams. Plain off-the-shelf set-top boxes or TV sets with built-in DVB receivers would not be able to decode such content. As discussed in Chapter A, this issue could be addressed by an advanced content distribution system.

6.3.3 Network Size

The WiBACK architecture and because of that also our UDT integration was designed to cover a limited network size in order to provide connectivity between the backbone and access point nodes, where the maximum distance was assumed to be ten hops [38]. The CARMEN project has considered and eventually chosen the centralized approach since it was determined to best support the requirements of such a back-haul network.

With an increasing number of hops, the protocol timing or back-off considerations might have to be adjusted. Moreover, an increasing number of WNs per *Master* might require the signalling protocols to be optimized to limit the amount of signalling traffic on the links in the vicinity of the master nodes as well as the CPU load on the *Master* nodes itself.

6.3.4 White Space Coordination

While in Chapter A we have presented an approach to provide orchestrated *white space* sharing, the standard WiBACK utilization of broadcast cells in the *white space* spectrum might pose issues for other *white space* technologies. Such devices might expect broadcast cells to be permanently active on a certain channel, while WiBACK might dynamically turn off unused cells or adjust their transmission range. This might lead non-WiBACK controlled devices to assume that certain channels may be free to use.

6.3.5 Asymmetric Behavior of Bidirectional Sessions

Certain applications or protocols expect symmetric bidirectional communication. Since WiBACK computes *Pipes* unidirectionally, it might determine asymmetric path for the downstream/upstream *Pipe* pair. This issue may become even more likely when UDTs are available which might, in one direction, provide even *shorter*, more direct connectivity.

6.4 Future Work

Future work may look into different aspects. We plan to further evaluate and complete our support for *Dynamic Broadcasting* and coordinated white space sharing. Furthermore, we plan to integrate energy-awareness on node and link levels into the WiBACK architecture to allow an energy-optimized operation of, for example, solar-powered nodes in rural deployment scenarios.

6.4.1 Link Local Connectivity via Unidirectional Technologies

A recent deployment scenario developed within our NET4DC[1] initiative suggest that we reconsider the support for UDTs for management connectivity. Instead of a rather generic solution supporting an arbitrary *loop* size, a rather limited mechanism might be sufficient to support *loops* consisting of exactly two links, one from node A to B and one from node B back to node A. This requirement stems from a request to support modified IEEE 802.11 radios which only operate in either in rx-only or tx-only mode. Hence, two pairs are required to provide bidirectional connectivity. Such a setup could either be supported via a modified MAC protocol or natively by the aforementioned simplified *loop* detection mechanism.

6.4.2 Dynamic Broadcast

The WiBACK architecture may be used to orchestrate coordinated white space usage, or coexist with other white space sharing systems, see [14]. Moreover, as a result of the UDT integration, the WiBACK architecture can support novel spectrum sharing approaches such as *Dynamic Broadcast*, which may temporarily free up VHF or UHF spectrum, but

---

[1] http://www.net4dc.org

require architectural support from the network to enforce their decisions and to optimally utilize such precious resources. See Appendix A for a more detailed discussion of a WiBACK-based *Dynamic Broadcast* scenario at Layer 2.5.

Especially for deployments in rural developing areas, an integration with a *Dynamic Broadcast* architecture may provide a variety of TV or radio programming while requiring a limited amount of infrastructure resources.

6.4.3 Optimized Macro Cell Sizing

To further improve the efficiency of *white space* coexistence in a *Dynamic Broadcast* scenario, an arbitration mechanism could be developed in order to determine the optimal size of a macro cell by choosing the transmit power and MCS settings in such a way that the optimal set of receiving nodes can be reached. Such a mechanism might extend the AI_RadioCalibrateLink primitive to address multiple receivers at once. This primitive might then be executed by a multicast tree computation algorithm in order to determine the optimal link and cell configurations.

6.4.4 Energy-Awareness

Deployment scenarios, especially in rural developing regions, such as the ones envisaged by the SolarMesh project[2], often include nodes or areas without access to a permanently available power grid. Here we plan to investigate how to extend the AI to support the notion of an *Energy Profile* of a node, its interfaces and power sources or batteries. Based on such a profile and the knowledge about current or predicted future traffic patterns, the centralized network management modules may optimize the active topology to bypass or even suspend nodes with low battery levels or to limit the traffic on certain links to emergency communication.

Energy-aware networking is a broad domain which currently receives tremendous attention from the research community but also by equipment vendors which must adhere to stricter energy consumption regulations. While in the developed world, a reduced energy footprint mainly reduces the electricity bill of the network operators, in rural developing areas, low-energy and awareness of temporal energy availability may enable the setup of communication networks in areas previously unconnected.

6.4.5 Next Steps

The first permanent outdoor WiBACK testbed was setup in June 2011 connecting a rural farm, which is *connected* in parallel with a DVB-T link using the Link Layer Tunneling Mechanism (LLTM) and a *DSL light* connection. Following the integration of UDTs into the extended WiBACK architecture, we are currently migrating the scenario to an integrated WiBACK testbed which would include the DVB-T cell for unicast and multicast capacity considerations.

---

[2] http://www.solarmesh.de

As a next major aspect we see the consideration of energy-awareness within the WiBACK architecture, which, so far, assumes that links are permanently available and allocated *Pipe* resources indicate a link as *used*, even though no traffic is temporarily forwarded through some *Pipes*.

At the same time, more sophisticated spectrum and capacity allocation algorithms are required to allow a more dynamic reallocation of *Pipe* or *Tree* resources to free up unused links which could then be suspended until increasing demand would trigger them to be brought back on-line.

While, in the developed world, such considerations may mainly help to reduce the energy footprint, in the emerging world, this may enable the coverage of previously unconnected regions.

References

[1] M. Kretschmer, J. Moedeker, and G. Ghinea, "Path signalling in a wireless back-haul network integrating unidirectional broadcast technologies," *IEEE Transactions on Broadcasting*, Under Review.

[2] M. Kretschmer, P. Batroff, and G. Ghinea, "Topology forming and optimization framework for heterogeneous wireless back-haul networks supporting unidirectional technologies," *Journal of Computer Communications (COMCOM, Elsevier)*, Under Review.

[3] M. Kretschmer, C. Niephaus, and G. Ghinea, *Wireless Multi-Access Environments and Quality of Service Provisioning: Solutions and Application*, ch. Heterogeneous Meshed Wireless Back-haul Network Integrating Unidirectional Technologies, pp. 139–160. Information Science Reference, USA/UK: IGI Global, 2012.

[4] M. Kretschmer and G. Ghinea, "Seamless integration of unidirectional broadcast links into qos-constrained broadband wireless mesh access networks," in *Proc. The 4th International Conference for Internet Technology and Secured Transactions*, 2009.

[5] A. Banchs, N. Bayer, D. Chieng, A. de la Oliva, B. Gloss, M. Kretschmer, S. Murphy, M. Natkaniec, and F. Zdarsky, "Carmen: Delivering carrier grade services over wireless mesh networks," in *Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2008*, pp. 1–6, Sept. 15–18, 2008.

[6] M. Kretschmer, P. Batroff, C. Niephaus, and G. Ghinea, "Topology discovery and maintenance for heterogeneous wireless Back-Haul networks supporting unidirectional technologies," in *17th Asia-Pacific Conference on Communications (APCC 2011)*, (Kota Kinabalu, Sabah, Malaysia), Oct. 2011.

[7] M. Kretschmer and G. Ghinea, "An IEEE 802.21-based approach for seamless wireless mobile integration using QoS-aware paths supporting unidirectional links," in *IEEE Globecom 2010 Workshop on Seamless Wireless Mobility (SWM 2010)*, (Miami, Florida, USA), 12 2010.

[8] D. Henkel, S. Englaender, M. Kretschmer, and C. Niephaus, "Connecting the unconnected - economical constraints and technical requirements towards a Back-Haul net-

work for rural areas," in *IEEE Globecom 2011 Workshop on Rural Communications-Technologies, Applications, Strategies and Policies (RuralComm 2011) (GC'11 Workshop - RuralComm)*, (Houston, Texas, USA), Dec. 2011.

[9] T. Horstmann, M. Kretschmer, C. Niephaus, J. Mödeker, and S. Sauer, "Development framework for prototyping heterogeneous Multi-Radio wireless networks," in *ICCCN 2011 Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN 2011)*, (Maui, Hawaii, USA), July 2011.

[10] M. Kretschmer, C. Niephaus, and G. Ghinea, *Towards QoS Provisioning in a Heterogeneous Carrier-Grade Wireless Mesh Access Networks Using Unidirectional Overlay Cells*, vol. 22 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.* Springer Berlin Heidelberg, 2009.

[11] M. Kretschmer, C. Niephaus, and G. Ghinea, "QoS-aware flow monitoring and event creation in heterogeneous MPLS-based wireless mesh networks supporting unidirectional links," in *9th IEEE Malaysia International Conference on Communications 2009*, (Kuala Lumpur, Malaysia), 2009.

[12] M. Kretschmer, C. Niephaus, D. Henkel, and G. Ghinea, "QoS-aware wireless back-haul network for rural areas with support for broadcast services in practice," in *Fifth IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking (IEEE MeshTech 2011)*, (Valencia, Spain), Oct. 2011.

[13] M. Kretschmer, S. Robitzsch, C. Niephaus, K. Jonas, and G. Ghinea, "Wireless mesh network coverage with QoS differentiation for rural areas," in *First International Workshop on Wireless Broadband Access for Communities and Rural Developing Regions*, (Karlstad, Sweden), 12 2008.

[14] M. Kretschmer, C. Niephaus, and G. Ghinea, "A wireless back-haul architecture supporting dynamic broadcast and white space coexistence," in *ICCCN 2012 Workshops: 6th International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN) (WiMAN 2012)*, (Munich, Germany), July 2012.

[15] M. Kretschmer, T. Horstmann, P. Batroff, M. Rademacher, and G. Ghinea, "Link calibration and property estimation in Self-Managed wireless Back-Haul networks," in *18th Asia-Pacific Conference on Communications (APCC 2012)*, (Ramada Plaza Jeju Hotel, Jeju Island, Korea), Oct. 2012.

[16] M. Kretschmer, C. Niephaus, and G. Ghinea, "Long-range wireless mesh networks for unconnected regions," in *E-Infrastructures and E-Services on Developing Countries: First International ICST Conference, AFRICOMM December 2009, Maputo, Mozambique*, vol. 38, LNICST, Springer, June 2010.

[17] M. Kretschmer, P. Hasse, C. Niephaus, T. Horstmann, and K. Jonas, "Connecting mobile phones via carrier-grade meshed wireless back-haul networks," in *E-Infrastructures and E-Services on Developing Countries: Second International ICST Conference, AFRICOMM 2010, Cape Town, South Africa* (Springer, ed.), October 2011.

[18] J. Lessmann, A. De La Oliva, C. Sengul, A. Garcia, M. Kretschmer, S. Murphy, and P. Patras, "On the scalability of carrier-grade mesh network architectures," in *Future Network Mobile Summit (FutureNetw), 2011*, pp. 1 –8, june 2011.

[19] S. Robitzsch, C. Niephaus, J. Fitzpatrick, and M. Kretschmer, "Measurements and evaluations for an ieee 802.11a based carrier-grade multi-radio wireless mesh network deployment," in *Wireless and Mobile Communications, 2009. ICWMC '09. Fifth International Conference on*, pp. 272–278, Aug. 2009.

[20] M. Kretschmer, C. Niephaus, T. Horstmann, and K. Jonas, "Providing mobile phone access in rural areas via heterogeneous meshed wireless back-haul networks," in *Communications Workshops (ICC), 2011 IEEE International Conference on*, pp. 1 –6, june 2011.

[21] C. Niephaus and M. Kretschmer, "Towards an energy management framework for carrier-grade wireless Back-Haul networks," in *ICCCN 2012 Workshops: 6th International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN) (WiMAN 2012)*, (Munich, Germany), July 2012.

[22] V. K. Vaishnavi and W. Kuechler, *Design Science Research Methods and Patterns: Innovating Information and Communication Technology.* Auerbach Publications, 1 ed., Oct. 2007.

[23] J. Qi, P. Neumann, and U. Reimers, "Dynamic broadcast," in *Electronic Media Technology (CEMT), 2011 14th ITG Conference on*, pp. 1 –6, march 2011.

[24] U. Reimers, "Technology trends and the future of broadband multimedia," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*, pp. 1 –7, march 2010.

[25] H. Kim and H. Yeom, "Dynamic scheme transition adaptable to variable video popularity in a digital broadcast network," *Multimedia, IEEE Transactions on*, vol. 11, pp. 486 –493, april 2009.

[26] D. Dechene and A. Shami, "Experimental triple-play service delivery using commodity wireless lan hardware," in *Communications, 2009. ICC '09. IEEE International Conference on*, pp. 1 –5, june 2009.

[27] F. Guo and T. cker Chiueh, "Software tdma for voip applications over ieee802.11 wireless lan," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2366 –2370, may 2007.

[28] N. Corporation, "Paso link solutions," December 2011.

[29] D. Chieng, D. Hugo, and A. Banchs, "A cost sensitivity analysis for carrier grade wireless mesh networks with tabu optimization," in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, pp. 1 –6, 2010.

[30] A. Mihailovic, I. Chochliouros, A. Kousaridas, G. Nguengang, C. Polychronopoulos, J. Borgel, M. Israel, V. Conan, M. Belesioti, E. Sfakianakis, G. Agapiou, H. Aghvami, and N. Alonistioti, "Architectural principles for synergy of self-management and future internet evolution," in *Proceedings of Future Network & MobileSummit 2009 Conference*, p. 8, Jun 2009.

[31] "Ieee standard for information technology–telecommunications and information exchange between systems wireless regional area networks (wran)–specific requirements part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures for operation in the tv bands," *IEEE Std 802.22-2011*, pp. 1 –680, 1 2011.

[32] J. Moy, "OSPF Version 2." RFC 2328 (Standard), Apr. 1998. Updated by RFC 5709.

[33] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)." RFC 1771 (Draft Standard), Mar. 1995. Obsoleted by RFC 4271.

[34] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." RFC 2460 (Draft Standard), Dec. 1998. Updated by RFCs 5095, 5722, 5871, 6437.

[35] "Ieee 802.3-2005/cor 1-2006," 2006.

[36] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, 12 2007.

[37] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *In First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON*, pp. 517–526, 2004.

[38] CARMEN-Consortium, "Ratified architecture deliverable."

[39] V. Ramasubramanian and D. Mosse, "Bra: A bidirectional routing abstraction for asymmetric mobile ad hoc networks," vol. IEEE/ACM TON 16, no. 1, pp. 116–129, 2008.

[40] T. C. Consortium, "Carrier-grade wireless mesh networks," January 2012.

[41] A. Azcorra, T. Banniza, D. Chieng, J. Fitzpatrick, D. Von-Hugo, M. Natkaniec, S. Robitzsch, and F. Zdarsky, "Supporting Carrier Grade Services over Wireless Mesh Networks - The Approach of the European FP-7 STREP CARMEN," *IEEE Communications Magazine*, vol. 47, pp. 14–16, Apr. 2009.

[42] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6." RFC 5213 (Proposed Standard), Aug. 2008.

[43] L. Zhou, B. Geller, B. Zheng, A. Wei, and J. Cui, "System scheduling for multi-description video streaming over wireless multi-hop networks," *Broadcasting, IEEE Transactions on*, vol. 55, no. 4, pp. 731 –741, 2009.

[44] "Ieee standard for local and metropolitan area networks- part 21: Media independent handover," *IEEE Std 802.21-2008*, pp. c1 –301, jan. 2009.

[45] A. De La Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal, "An overview of ieee 802.21: media-independent handover services," *Wireless Communications, IEEE*, vol. 15, pp. 96 –103, aug. 2008.

[46] A. d. la Oliva, I. Soto, A. Banchs, J. Lessmann, C. Niephaus, and T. Melia, "Ieee 802.21: Media independence beyond handover," *Comput. Stand. Interfaces*, vol. 33, pp. 556–564, November 2011.

[47] W. Team, "A solution for seamless handover among multi-provider heterogeneous networks," 2006.

[48] A. Arjona and H. Verkasalo, "Unlicensed mobile access (uma) handover and packet data performance analysis," in *Digital Telecommunications, 2007. ICDT '07. Second International Conference on*, p. 9, 2007.

[49] P. Serrano, P. Patras, X. Perez-Costa, B. Gloss, and D. Chieng, "A MAC Layer Abstraction for Heterogeneous Carrier Grade Mesh Networks," in *ICT-MobileSummit '09*, (Santander, Spain), June 10-12 2009.

[50] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing." RFC 3561 (Experimental), July 2003.

[51] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4." RFC 4728 (Experimental), Feb. 2007.

[52] G. Fairhurst and B. Collini-Nocker, "Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS)." RFC 4326 (Proposed Standard), Dec. 2005.

[53] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, "A Link-Layer Tunneling Mechanism for Unidirectional Links." RFC 3077 (Proposed Standard), Mar. 2001.

[54] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic Routing Encapsulation (GRE)." RFC 2784 (Proposed Standard), Mar. 2000. Updated by RFC 2890.

[55] T. Buburuzan, G. May, T. Melia, J. Modeker, and M. Wetterwald, "Integration of broadcast technologies with heterogeneous networks - an ieee 802.21 centric approach," in *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, pp. 1–2, Jan. 2007.

[56] L. Andersson and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field." RFC 5462 (Proposed Standard), Feb. 2009.

[57] L. Guoqing and Q. Zhaowei, "Unidirectional link problem in aodv routing protocol," in *Broadband Network Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on*, pp. 1 –4, oct. 2009.

[58] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271 (Draft Standard), Jan. 2006. Updated by RFC 6286.

[59] G. Malkin, "RIP Version 2." RFC 2453 (Standard), Nov. 1998. Updated by RFC 4822.

[60] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan." RFC 4632 (Best Current Practice), Aug. 2006.

[61] G. Malkin and R. Minnear, "RIPng for IPv6." RFC 2080 (Proposed Standard), Jan. 1997.

[62] D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2." RFC 3630 (Proposed Standard), Sept. 2003. Updated by RFCs 4203, 5786.

[63] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering." RFC 3272 (Informational), May 2002. Updated by RFC 5462.

[64] C. J. Bernardos, J. Fitzpatrick, F.-C. Kuo, M. Kretschmer, J. Lessmann, C. Niephaus, A. de la Oliva, S. Robitzsch, and F. Zdarsky, "Carrier-grade wireless mesh networks: D3.4-unicast and multicast routing specification and analysis." http://www.ict-carmen.eu/wp-uploads/2009/D3.4.pdf, Dec 2009.

[65] R. Ogier and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding." RFC 5614 (Experimental), Aug. 2009.

[66] L. Bao and J. J. Garcia-Luna-Aceves, "Link-state routing in networks with unidirectional links," in *Proc. Eight International Conference on Computer Communications and Networks*, pp. 358–363, 1999.

[67] K. Sridhar, C. Casetti, and C.-F. Chiasserini, "A localized and distributed channel assignment scheme for wireless mesh networks," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pp. 45 –52, oct. 2009.

[68] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)." RFC 3626 (Experimental), Oct. 2003.

[69] I. F. Akyildiz and X. Wang, *Wireless Mesh Networks.* John Wiley & Sons Ltd, 1 ed., 2009.

[70] "Ieee draft standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications-amendment 10: Mesh networking," *IEEE P802.11s/D8.0, December 2010*, pp. 1 –350, 23 2010.

[71] A. O. Lim, X. Wang, Y. Kado, and B. Zhang, "A hybrid centralized routing protocol for 802.11s wmns," *Mob. Netw. Appl.*, vol. 13, pp. 117–131, April 2008.

[72] A. Ksentini and O. Abassi, "A comparison of voip performance over three routing protocols for ieee 802.11s-based wireless mesh networks (wlan mesh)," in *Proceedings of the 6th ACM international symposium on Mobility management and wireless access*, MobiWac '08, (New York, NY, USA), pp. 147–150, ACM, 2008.

[73] K. Ghaboosi, M. Latva-Aho, and R. Kohno, "On a distributed cognitive mac protocol for ieee 802.11s wireless mesh networks," *Wirel. Pers. Commun.*, vol. 58, pp. 565–580, June 2011.

[74] J. Ben-Othman, L. Mokdad, and M. Cheikh, "Q-hwmp: Improving end-to-end qos for 802.11s based mesh networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1 –6, dec. 2010.

[75] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 1459–1473, Dec. 2008.

[76] G. Athanasiou, I. Broustis, T. Korakis, and L. Tassiulas, "Routing-aware channel selection in multi-radio mesh networks," in *Communications, 2009. ICC '09. IEEE International Conference on*, pp. 1 –6, june 2009.

[77] W. Kim, A. Kassler, M. Di Felice, and M. Gerla, "Urban-x: Towards distributed channel assignment in cognitive multi-radio mesh networks," in *Wireless Days (WD), 2010 IFIP*, pp. 1 –5, oct. 2010.

[78] V. Mirchandani, A. Prodan, and O. Marce, “A non-tpc based enhanced topology control process for multi-radio wireless mesh networks,” in *Sensor Technologies and Applications, 2008. SENSORCOMM ’08. Second International Conference on*, pp. 758 –763, aug. 2008.

[79] Y. Zhou, M. Yun, T. Kim, A. Arora, and H.-A. Choi, “Rl-based queue management for qos support in multi-channel multi-radio mesh networks,” in *Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on*, pp. 306 –309, july 2009.

[80] S. Maurina, J. Fitzpatrick, L. Trifan, and L. Murphy, “An Enhanced Bridged-Based Multi-Hop Wireless Network Implementation,” in *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, pp. 1–9, 2010.

[81] J. Y. Yu, P. H. J. Chong, and M. Yang, “Performance of microcell/macrocell cellular systems with reuse partitioning,” in *Mobility ’06: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, (New York, NY, USA), p. 51, ACM, 2006.

[82] Q. Huang, K.-T. Ko, S. Chan, and V. B. Iversen, “Loss performance evaluation in heterogeneous hierarchical networks,” in *Mobility ’08: Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, (New York, NY, USA), pp. 1–7, ACM, 2008.

[83] J. Deissner and G. P. Fettweis, “Increased capacity through hierarchical cellular structures with inter-layer reuse in an enhanced gsm radio network,” *Mob. Netw. Appl.*, vol. 6, no. 5, pp. 471–480, 2001.

[84] A. Reaz, V. Ramamurthi, D. Ghosal, J. Benko, W. Li, S. Dixit, and B. Mukherjee, “Enhancing multi-hop wireless mesh networks with a ring overlay,” in *Proc. 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops SECON Workshops ’08*, pp. 1–6, 2008.

[85] P. Zhou, B. S. Manoj, and R. Rao, “A gateway placement algorithm in wireless mesh networks,” in *WICON ’07: Proceedings of the 3rd international conference on Wireless internet*, (ICST, Brussels, Belgium, Belgium), pp. 1–9, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[86] B. Liu, P. Thiran, and D. Towsley, “Capacity of a wireless ad hoc network with infrastructure,” in *MobiHoc ’07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, (New York, NY, USA), pp. 239–246, ACM, 2007.

[87] E. Haddad and J.-C. Gregoire, “Implementation issues for the deployment of a wmn with a hybrid fixed/cellular backhaul network in emergency situations,” in *Proc. 1st*

International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology Wireless VITAE 2009, pp. 525–529, 2009.

[88] J. del Prado Pavon and S. Shankar, "Impact of frame size, number of stations and mobility on the throughput performance of ieee 802.11e," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 2, pp. 789 – 795 Vol.2, march 2004.

[89] D. Frost and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks." RFC 6374 (Proposed Standard), Sept. 2011.

[90] A. H. Asgari, P. Trimintzios, M. Irons, R. Egan, and G. Pavlou, "Building quality-of-service monitoring systems for traffic engineering and service management," *J. Netw. Syst. Manage.*, vol. 11, no. 4, pp. 399–426, 2003.

[91] P. Pan, G. Swallow, and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels." RFC 4090 (Proposed Standard), May 2005.

[92] A. Raj and O. C. Ibe, "A survey of ip and multiprotocol label switching fast reroute schemes," *Comput. Netw.*, vol. 51, no. 8, pp. 1882–1907, 2007.

[93] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, (New York, NY, USA), pp. 246–259, ACM, 2006.

[94] M. Rondinone, J. Ansari, J. Riihijärvi, and P. Mähönen, "Designing a reliable and stable link quality metric for wireless sensor networks," in *REALWSN '08: Proceedings of the workshop on Real-world wireless sensor networks*, (New York, NY, USA), pp. 6–10, ACM, 2008.

[95] S. Jiang, D. He, and J. Rao, "A prediction-based link availability estimation for routing metrics in manets," *IEEE/ACM Trans. Netw.*, vol. 13, no. 6, pp. 1302–1312, 2005.

[96] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information." RFC 5101 (Proposed Standard), Jan. 2008.

[97] B. Claise, G. Dhandapani, P. Aitken, and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)." RFC 6313 (Proposed Standard), July 2011.

[98] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture." RFC 3031 (Proposed Standard), Jan. 2001. Updated by RFC 6178.

[99] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta, "MPLS Label Stack Encoding." RFC 3032 (Proposed Standard), Jan. 2001. Updated by RFCs 3443, 4182, 5332, 3270, 5129, 5462, 5586.

[100] L. Martini, E. Rosen, N. El-Aawar, and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks." RFC 4448 (Proposed Standard), Apr. 2006. Updated by RFC 5462.

[101] J. Lessmann, M. Schoeller, and F. Zdarsky, "Rope ladder routing: Position-based multipath routing for wireless mesh networks," in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pp. 1 –6, 2010.

[102] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis, "Constraint-Based LSP Setup using LDP." RFC 3212 (Proposed Standard), Jan. 2002. Updated by RFC 3468.

[103] J. Ash, Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki, and L. Li, "LSP Modification Using CR-LDP." RFC 3214 (Proposed Standard), Jan. 2002.

[104] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification." RFC 3036 (Proposed Standard), Jan. 2001. Obsoleted by RFC 5036.

[105] L. Andersson and G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols." RFC 3468 (Informational), Feb. 2003.

[106] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification." RFC 2205 (Proposed Standard), Sept. 1997. Updated by RFCs 2750, 3936, 4495, 5946, 6437.

[107] K. Kompella and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)." RFC 3477 (Proposed Standard), Jan. 2003. Updated by RFC 6107.

[108] 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), Dec. 2001. Updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711.

[109] R. Aggarwal, Y. Rekhter, and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space." RFC 5331 (Proposed Standard), Aug. 2008.

[110] T. Eckert, E. Rosen, R. Aggarwal, and Y. Rekhter, "MPLS Multicast Encapsulations." RFC 5332 (Proposed Standard), Aug. 2008.

[111] O. Komolafe and J. Sventek, "Analysis of rsvp-te graceful restart," pp. 2324 –2329, jun. 2007.

[112] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini, “RSVP Refresh Overhead Reduction Extensions.” RFC 2961 (Proposed Standard), Apr. 2001. Updated by RFC 5063.

[113] A. Farrel, J.-P. Vasseur, and J. Ash, “A Path Computation Element (PCE)-Based Architecture.” RFC 4655 (Informational), Aug. 2006.

[114] J. Vasseur and J. L. Roux, “Path Computation Element (PCE) Communication Protocol (PCEP).” RFC 5440 (Proposed Standard), Mar. 2009.

[115] L. Given, *The Sage Encyclopedia of Qualitative Research Methods.* Thousand Oaks: Sage Publications, Inc, 2008.

[116] A. Navarro Sada and A. Maldonado, “Research methods in education. sixth edition - by louis cohen, lawrence manion and keith morrison,” *British Journal of Educational Studies*, vol. 55, no. 4, pp. 469–470, 2007.

[117] I. Hacking, *Scientific Revolutions.* Oxford Oxfordshire: Oxford University Press, 1981.

[118] I. Niiniluoto, “Scientific progress,” in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), summer 2011 ed., 2011.

[119] A. Hevner, S. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Quarterly*, no. 28, pp. 75–105, 2004.

[120] S. Ivanov, A. Herms, and G. Lukas, “Experimental validation of the ns-2 wireless model using simulation, emulation, and real network,” *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference*, pp. 1 –12, 26 2007-march 2 2007.

[121] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala, “Improving simulation for network research,” Tech. Rep. 99-702b, University of Southern California, March 1999.

[122] K. Pawlikowski, H.-D. Jeong, and J.-S. Lee, “On credibility of simulation studies of telecommunication networks,” *Communications Magazine, IEEE*, vol. 40, pp. 132 –139, jan 2002.

[123] R. M. Fujimoto, K. S. Perumalla, and G. F. Riley, *Network Simulation.* Morgan & Claypool, 2007.

[124] J. Banks, “Introduction to simulation,” in *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future - Volume 1*, WSC ’99, (New York, NY, USA), pp. 7–13, ACM, 1999.

[125] K. Pawlikowski, H.-D. Jeong, and J.-S. Lee, “On credibility of simulation studies of telecommunication networks,” *Communications Magazine, IEEE*, vol. 40, pp. 132 –139, Jan. 2002.

[126] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The click modular router,” *ACM Trans. Comput. Syst.*, vol. 18, pp. 263–297, August 2000.

[127] A. Zimmermann, M. Gunes, M. Wenig, U. Meis, and J. Ritzerfeld, “How to study wireless mesh networks: A hybrid testbed approach,” in *Advanced Information Networking and Applications, 2007. AINA ’07. 21st International Conference on*, pp. 853 –860, May 2007.

[128] J. Whiteaker, F. Schneider, and R. Teixeira, “Explaining packet delays under virtualization,” *Computer Communications Review (CCR)*, vol. 41, pp. 38–44, January 2011.

[129] D. Wagner, J. Moedeker, and T. Horstmann, “Dynamic protocol functionality in cognitive future internet elements,” in *Proceedings of Future Network & MobileSummit 2010 Conference*, p. 8, Jun 2010.

[130] C. Demichelis and P. Chimento, “IP Packet Delay Variation Metric for IP Performance Metrics (IPPM).” RFC 3393 (Proposed Standard), Nov. 2002.

[131] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, “Senseless: A database-driven white spaces network,” *Mobile Computing, IEEE Transactions on*, vol. 11, pp. 189 –203, feb. 2012.

[132] C. Gerami, N. Mandayam, and L. Greenstein, “Backhauling in tv white spaces,” in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pp. 1 –6, dec. 2010.

Appendix A

Use Case: Dynamic Broadcast

In this chapter we discuss how the UDT-aware WiBACK architecture can support the recently proposed *Dynamic Broadcast* paradigm by providing the framework to orchestrate coordinated white space usage. We show that as a result of the UDT integration, the WiBACK architecture can support novel spectrum sharing approaches such as *Dynamic Broadcast*, which may temporarily free up VHF or UHF spectrum, but require architectural support from the network to enforce their decisions and to optimally utilize such precious resources.

A.1 Scenario

In a recent paper [23] a *Dynamic Broadcast* architecture was proposed to optimize the UHF spectrum usage taking into account new user terminals which are capable of receiving content from different networks and also providing temporary storage for non-live content. The authors argue that modern broadcast technologies, such as DVB-T2, which provide very high spectral efficiencies are ideal to efficiently distribute live content to many users simultaneously. They further argue that by cleverly decomposing traditional TV programming into live and non-live segments and optimizing the distribution schedule of live and non-live content, parts of the precious UHF spectrum could be temporarily freed up. Those temporarily vacated *white spaces* could be used by wireless operator or wireless Internet Service Provider (ISP) networks to increase their coverage, especially in sparsely populated rural areas.

Compared to its predecessors, DVB-T2 supports Advanced Coding and Modulation (ACM) which allows for the Modulation and Coding Scheme (MCS) to be adapted on a per-receiver basis. Hence depending on the receiver distribution and their received signal quality, the MCS could be adapted, possibly freeing up additional resources compared to a traditionally rather conservative configuration. In addition, the transmit power could be reduced to, for example, decrease the coverage range allowing for a more efficient frequency reuse or to conserve energy.

While the authors focus mainly on the higher layer broadcast scheduling and playout system requirements, the aim of this study is to show that our WiBACK architecture provides the required primitives and mechanisms to support a *Dynamic Broadcast* system:

- Heterogeneous bi- and unidirectional Technologies
- Spectrum Management
- Transmission parameter configuration
- Resource allocation and QoS enforcement

Thus, we will show that the WiBACK architecture can facilitate the management of the scarce wireless spectra and support a controlled coexistence of heterogeneous technologies within the so-called *white spaces* providing the foundation for converged broadcast and data networks.

A.2 WiBACK-based Architecture

In the following sections, we will describe the WiBACK architecture components addressing the above requirements of a *Dynamic Broadcast* system.

A.2.1 Heterogeneous Technologies

To identify radio interfaces, WiBACK relies on the IEEE 802.21 *InterfaceId* which contains a *LinkType*, an *AddressFamily* and the *LinkAddress*. Thus, WiBACK can identify almost any kind of technology. DVB-T2, for example, can readily be supported via either MPE or the more recent GSE. For each supported technology a specific MAC Adaptor would be required to map the below listed Abstract Interface (AI) primitives onto technology specific mechanisms, see Table A.1. During the bootstrap phase, each WN performs a capability discovery to determine the types and capabilities of its local interfaces. Based on this information, the TMF forms or adapts the topology following configured optimization goals.

In [3], we have discussed in detail how UDTs are integrated into the WiBACK architecture so that they can be utilized when beneficial to achieve network-wide optimization goals while exploiting the full lower layer information provided by the AI. This aspect concerns the description of radio interfaces, link monitoring, topology management, UDT-aware RSVP-TE-style *Pipe* signaling and, to a lesser degree, path or multicast tree computation.

Considering the above subset of AI primitives, the TMF is aware that, for example, transmit-only interfaces may provide primitives to control the transmit power level or to set the MCS, but can not perform a channel scan, and vice versa for receive-only interfaces.

Primitive Name	Description
AI_RadioGetProperties	Returns interface properties
AI_RadioGetEnvelope	Returns parameters describing the typical spectral envelope
AI_RadioJoinCell	Instructs interface to join a cell
AI_RadioLeaveCell	Instructs interface to leave a cell
AI_RadioSetParameters	Sets TxPower, MCS, range
AI_RadioCalibrateLink	Instructs interface to calibrate a link
AI_RadioChannelScan	Triggers a channel or regulatory scan
AI_LinkDown	Indicates a stale or broken link
AI_PipeDown	Indicates a broken or underperforming *Pipe* (i.e. QoS violation)
AI_RegulatoryEvent	Indicates a regulatory issue on a link (i.e. radar, broadcaster detected)

Table A.1: Subset of AI primitives to manage heterogeneous radios

A.2.2 Spectrum Management

The WiBACK AI supports spectrum management with primitives to query interface capabilities such as the typical spectral envelope, the supported frequency range, transmit power levels or MCS options. It also provides a primitive to trigger channel and regulatory scans on receive-capable interfaces. Taking into account current capacity demands and node locations, the gathered information can be used by the TMF *Master* to chose the most suitable technologies, channels and link configurations to optimally utilize the spectrum resources. Our architectural work focuses on the primitives and mechanisms to manage the coordinated coexistence of heterogeneous technologies, while it allows for different spectrum allocation or optimization algorithms, such as database-driven [131] or rather cognitive [132] approaches to be deployed. Moreover, the WiBACK architecture supports the provision of Internet, VoIP and broadcast content using the most suitable technologies under, for example, considerations of regulatory issues, availability or cost factors. For example, for broadcast traffic the rather static DVB-T or the more dynamic DVB-T2 might be used, while for data back-hauling in the *white space* spectrum IEEE 802.22 or sub-GHz IEEE 802.11ah technologies could be used. Additional back-haul capacity in licensed bands, i.e. via IEEE 802.16, or unlicensed bands, i.e. via IEEE 802.11a, might be included. Moreover, fixed micro-ware or even optical fiber links can be integrated.

A.2.3 Regulatory Constraints

The WiBACK architecture supports coexistence, as mandated, for example, for TV *white spaces* or the U-NII band, via the AI_RadioChannelScan primitive and the AI_RegulatoryEvent event. Passive channel scans can be enforced before an interface actively switches into a certain channel. If, during operation, an interface detects a *regulatory event*, such as a present radar signal, a broadcast signal, or a wireless microphone, it may, through its MAC Adaptor, trigger an AI_RegulatoryEvent.Indication towards the TMF. The TMF in turn, must then disable the link and may attempt a reassignment onto another channel.

A.2.4 Transmission Parameter Configuration

Once the TMF *Master* has chosen the interfaces and assigned the channels, the established links can be configured by either issuing an AI_RadioSetParameter primitive or by issuing an AI_RadioCalibrateLink primitive which will perform an automated link calibration to, for example, determine optimal MCS, transmit power level or range parameters. In the unidirectional case of, for example, DVB-T2 in ACM mode, a temporary feedback *Pipe* can readily be provided by the WiBACK architecture to support link calibrations over UDTs, greatly simplifying an otherwise often manually administered task. This mechanism can be compared to DVB-S2/RCS ACM adaptations.

A.2.5 Resource Allocation and QoS Enforcement

While physical links are described with properties such as channel, Tx-Power and MCS, logical links are described with properties such as capacity, typical latency or loss probability. The logical parameters of a link are returned as the result of the AI_RadioCalibrateLink primitive. The capacity returned by the abstract AI_RadioCalibrateLink primitive is assumed to be the brutto capacity of the calibrated link assuming that this link may exclusively utilize all available wireless cell resources. It is the task of the CMF to determine all links sharing a cell and to orchestrate the resource distribution among such links and the *Pipes* established on top of them. The details of the CMF resource allocation model are outside of the scope of this study. In order to allocate resources for *Pipes* between any two WNs, the stateful CMF would attempt to compute a path satisfying the resource requirements. If such a path exists, a corresponding *Pipe* is set up by pushing LSP state into the involved WNs via the RSVP-TE-inspired PMF, which also determines the end-to-end MTU and performs resource allocation on the transmitting interfaces along the path. This information may be used to program MAC schedulers, traffic shapers or may be used as thresholds for link or LSP monitoring, as, for example, described in RFC 6374 [89], or in [11] for the WiBACK architecture.

A.3 Example Scenario

The WiBACK software is built upon our Simple and Extensible Network Framework (SENF)[1]-based NetEMU framework which provides a real-time network emulator. This allows us to evaluate the same binary code on emulated or real embedded nodes [9]. The topology representation is based on the C++ boost graph library as a *directed multi graph* and can easily be serialized into *dot* or the XML-derivate *GraphML* format for external processing or visualization. We have implemented an exemplary spectrum allocation algorithm for our architectural validation purposes which aims at choosing *free* channels as reported by the AI_RadioChannelScan primitive and attempts to configure point-to-point links to minimize the risk of interferences and collisions. Where an active DVB-T cell and

---

[1] http://senf.berlios.de

a regular back-haul link contend for the same spectrum resources, the DVB-T cell would take precedence. Once a link has been established, the AI_RadioCalibrateLink primitive is executed to determine the optimal link configuration as well as the resulting logical link properties, such as the brutto capacity or nominal latency.

To validate our proposed support for *Dynamic Broadcast*, we have created an emulated scenario consisting of four different technologies, a 100BaseT Ethernet link, IEEE 802.11a radios, a DVB-T cell, as well as proprietary sub-GHz IEEE 802.11 radios with a center frequency of 768 MHz at 20Mhz bandwidth[2]. The *dot* outputs shown in Figure A.1 depict the topologies formed by the TMF, represented by their activated logical links. Logical links are described as unidirectional resources and the annotations of each link show in the first line the (truncated) LinkId (source and destination MAC address) and in the second line the determined brutto capacity in Mbps as well as the assigned center frequency in MHz.

On the left, Figure A.1 depicts the frequency assignment with an active DVB-T cell at 762 MHz, taking precedence over the sub-GHz radios. On the right the same topology is depicted with the DVB-T cell turned off. In this case, our exemplary topology optimization algorithm was able to establish a higher bandwidth link between nodes 2 and 6 via the longer range sub-GHz radios. Freeing up a IEEE 802.11a radio, this also led to a direct point-to-point link between node 2 and node 5. As a result, node 5 is now connected via a dedicated 9Mbps link, while node 6 is connected via a dedicated 36Mbps link, significantly increasing the the total bandwidth available to both nodes. Hence, the vacated *white space* spectrum was used to tremendously increase the data network's capacity.

Our outdoor WiBACK testbed at the Fraunhofer Campus in Sankt Augustin, Germany consists of multiple-radio nodes equipped with IEEE 802.11a and the above mentioned sub-GHz WLAN radios [20]. From a previous project we operate a DVB-T transmitter to evaluate back-haul connectivity to a remote farm using the RFC3077 [53] LLTM approach where the return channel is realized via a DSL connection. We're currently working the technical, but mainly regulatory issues to include the DVB-T cell in our WiBACK testbed to validate the aforementioned emulation results in a real network.

A.4 Discussion

We have shown that our WiBACK architecture can support a *Dynamic Broadcast* system [23] in parallel to Internet and VoIP data services by providing the mechanisms to dynamically manage the temporarily freed up wireless spectrum resources. Moreover, our architecture supports the co-existence of heterogeneous broadcast and wireless data technologies in the *white space* spectrum and is complementary to actual dynamic or static spectrum sharing mechanisms as described, for example, in [131] or [132].

While the policies of spectrum sharing would have to be negotiated among the respective license holders or regulatory bodies, the WiBACK architecture provides the mechanisms

---

[2] i.e. Ubiquity XR7, http://www.ubnt.com/xr7

to enforce them. Hence, the WiBACK architecture can help to meet the ever increasing demand for wireless capacity exploiting vacant *white spaces*.

Supporting the distribution of *live* TV content via the WiBACK architecture would require traditional set-top boxes to handle MPE and MPLS instead of Moving Picture Experts Group (MPEG) transport streams. Plain off-the-shelf set-top boxes or TV sets with built-in DVB receivers would not be able to decode such content. On the other hand, a *Dynamic Broadcast* architecture, for example, would require more capable hardware in any case, which could then readily support such streams, possibly following an all-IP approach.

While the WiBACK architecture targets medium scale deployment scenarios, it could interface with country-wide *white space* management systems to dynamically orchestrate the efficient utilization of all available spectrum within its domain to best accommodate capacity demands for data, VoIP or TV programming services.

In this context, the role of such an advanced set-top box would have to be discussed, since the WiBACK architecture would consider it as a User Terminal (UT). In the current WiBACK design, UTs can only communicate with access interfaces of WiBACK access point nodes. Hence, they could not directly subscribe to content distributed via *Pipes* or multicast *Trees* inside the WiBACK network. This would mean that the access point would have to receive such content via its back-haul interfaces (i.e. DVB, but also any other technology) and forward it to the client via the access interfaces, which might quickly lead to scalability issues regarding the access capacities if a larger number of such UTs is active.

Alternatively, the access points could implement a *proxy mode*, where they would regularly subscribe to the respective *Pipes* or *Trees*, but would delegate the actual reception of the content to the UT itself, if it has been determined that the UT is capable and permitted to do so.

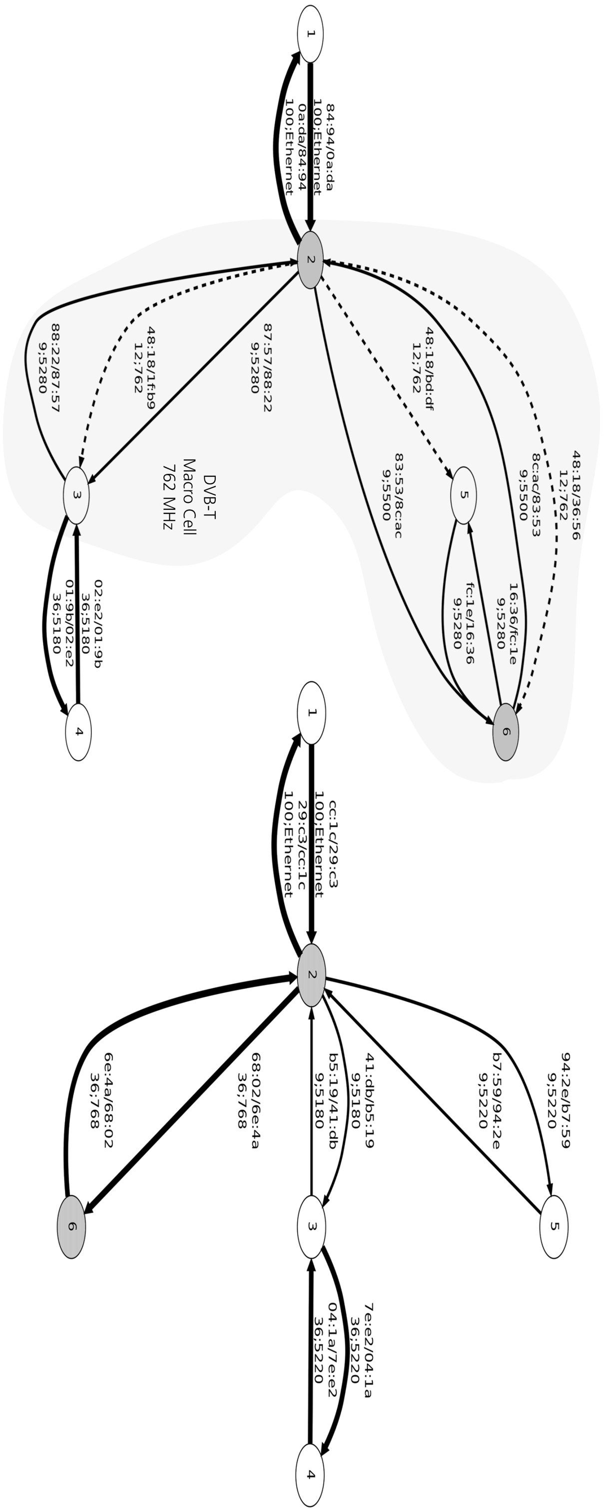

Figure A.1: *dot* visualization of left) the frequency assignment with an active DVB-T cell and right) without the DVB cell where the sub-GHz 802.11 links are operating in the vacant spectrum providing a direct link between nodes 2 and 6. The link addresses shown have been shortened to avoid cluttering the figure. The numbers in the second row of each link label show the determined brutto link capacity in Mbps and the channel center frequency in MHz.