Path Signalling in a Wireless Back-haul Network Integrating Unidirectional Broadcast Technologies

Mathias Kretschmer*†, Jens Moedeker*, Gheorghita Ghinea†
*Fraunhofer FOKUS, Sankt Augustin, Germany
†Brunel University, London, United Kingdom
{mathias.kretschmer|jens.moedeker}@fokus.fraunhofer.de, george.ghinea@brunel.ac.uk

***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, in our previous work, we have developed a cost-effective heterogeneous Layer 2.5 Wireless Back-Haul (WiBACK) architecture, which leverages the native multicast capabilities of broadcast technologies such as DVB to off-load high-bandwidth broadcast content delivery. Furthermore, our architecture provides support for unidirectional technologies on the data and the control plane. It adopts a centralized coordinator approach, in which coordinator nodes install so-called management and data pipes. No routing state is kept at plain WiBACK nodes, which merely store QoS-aware pipe forwarding state. Consequently, the architecture requires a reliable protocol to push resource allocation and pipe forwarding state into the network, considering possibly unidirectional connectivity. Such protocol, whose task is related to MPLS label distribution, is essential during the initial forming of WiBACK topologies and during regular network operations to reliably manage the data pipes. In this paper, we present a novel approach to extend our IEEE 802.21-inspired WiBACK *TransportService* and, based upon this, the design of an RSVP-TE-style pipe signalling protocol using nested hop-by-hop Request/Response MIH transactions which supports signalling over unidirectional technologies. A thorough evaluation and successful testbed deployments show that this protocol reliably signals pipe state even under high loss conditions.**

***Index Terms*—Heterogeneous Wireless Mesh, QoS, MPLS, RSVP-TE, Unidirectional Links, IEEE 802.21**

I. Motivation

Compared to traditional wired or fixed micro-wave-based operator back-haul networks, meshed Wireless Back-Haul (WiBACK) [1] networks offer simplified deployment and maintenance processes due to their flexible self-management characteristics [1]. These 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 Digital Video Broadcast (DVB), point-to-point micro-wave, optical or even wired solutions.

For the WiBACK architecture to be considered as an alternative for a rather over-provisioned operator back-haul network, it must meet strict requirements such as guaranteed Quality of Service (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 programming can introduce a high load on capacity-constrained interference-sensitive wireless links and approaches for IP-based networks to distribute such load either temporally, topologically or quantitatively have been proposed [2], [3], [4]. To architecturally support the efficient distribution of multicast or broadcast content, the WiBACK design aims at seamlessly integrating broadcast technologies such as DVB, see Figure 1. This would allow the spectrum as well as capacity management algorithms to dynamically shift such multicast traffic from the regular point-to-point to more efficient broadcast technologies, possibly depending on content, customer demand, density and distribution. Hence, the WiBACK architecture would support the re-use of existing broadcast infrastructure exploiting the benefits of the usually longer range of broadcast cells as well as their higher spectral efficiency [5]. In the case of DVB, the technology can easily be integrated using either the Multi Protocol Encapsulation (MPE) or the more recent Generic Stream Encapsulation (GSE).

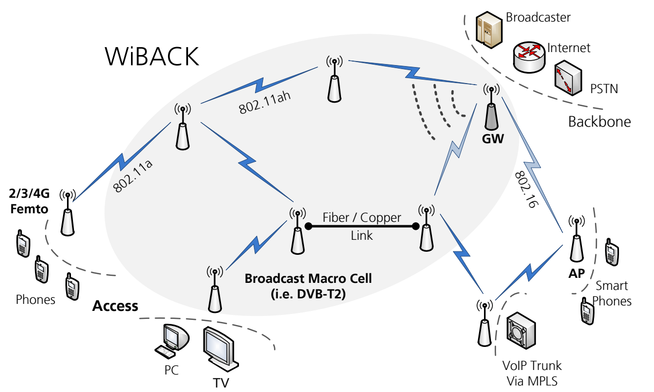

Fig. 1. The QoS-aware WiBACK architecture integrates heterogeneous technologies where a topology of nodes is centrally managed by so-called Gateway (GW) nodes. Network access via AP nodes is provided for individual Mobile Terminals (MTs) or trunked payload, such as VoIP back-hauling.

The scope of the 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. Our heterogeneous WiBACK architecture is inspired by the consolidated outcomes of the EU FP7 CARrier grade wireless MEsh Network (CARMEN)[6], [7] project and is based on a cross-layer concept that operates *below the Network Layer* [8] and does not rely on

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

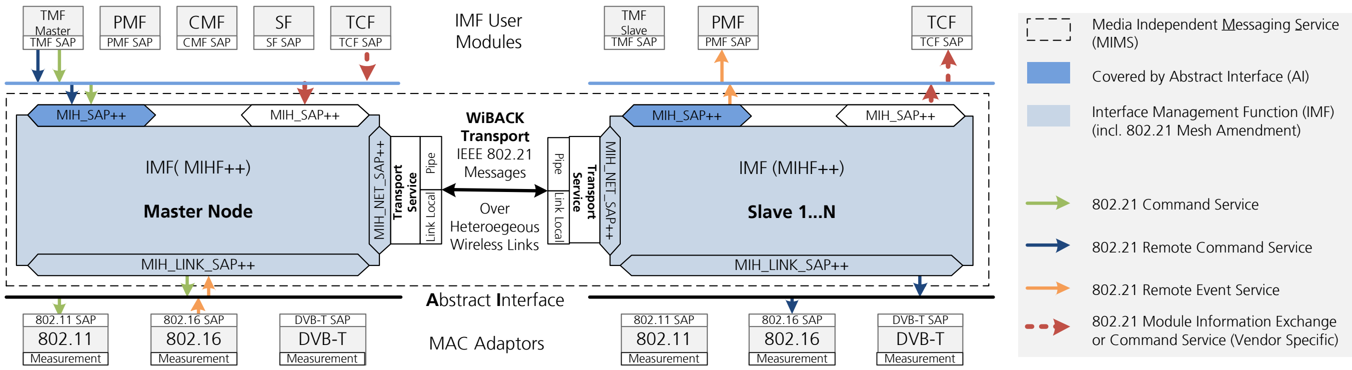

Fig. 2. The WiBACK control plane: IMF extends the IEEE 802.21 MIHF with an Abstract Interface and Module-to-Module Communication

an operational Internet Protocol (IP)-based network. It follows a centralized self-management approach and builds on an extended version of IEEE 802.21 command, event and information services, hardware abstraction, technology independent Multi Protocol Label Switching (MPLS)-based Traffic Engineering[9] and a model to address potentially shared wireless channel resources. In the WiBACK architecture, MPLS Label-Switched Paths (LSPs) are associated with per-hop resource allocation and referred to as *Pipes*. These are used as aggregates providing resource isolation among traffic classes as well as individual *data pipes* of the same traffic class. The *Pipe* concept also provides support for MT mobility through interaction with, for example, Proxy Mobile IP (PMIP)[10].

WiBACK *Master* nodes communicate with their *Slave* nodes via dedicated so-called *Management Pipes*, while actual payload is carried via *Data Pipes*. Hence, in order to signal *Pipe* state, especially during the topology forming phase where no other signalling options are available, a reliable protocol meeting the following requirements is needed:

- Signaling over Heterogeneous Technologies including Unidirectional Technologies
- Fast and confirmed *Pipe* setup (hard-state)
- Support for per-hop resource allocations
- Support for fast-failover signalling
- Signalling of a *Pipe's* payload type
- Support for downstream- and upstream-assigned (multicast) MPLS labels

The WiBACK control plane already provides an extended version of the media independent IEEE 802.21 messaging architecture. To avoid redundant functionality, the proposed *Pipe* signalling mechanism should consider leveraging the WiBACK messaging functionality, probably extending it where required.

This article is structured as follows. In the next section we provide a summary of the relevant components of the WiBACK architecture followed by a discussion of related work and a comparison against the WiBACK *Pipe* signalling mechanism requirements. We then describe our contribution, the integration of explicit source routing into WiBACK's IEEE 802.21-based messaging system as well as the design of the Pipe Management Protocol (PMP). In section four, we validate and evaluate our approach. Concluding, we summarize our contribution and give an outlook on further work.

II. Related Work

In this section we first introduce the relevant aspects of the IEEE 802.21 architecture followed by the amendments introduced by the CARMEN project. Some of those amendments have been accepted for inclusion in the upcoming IEEE 802.21b standard, currently under sponsor ballot. Built upon the amended IEEE 802.21 architecture, we summarize the relevant aspects of the WiBACK architecture focusing on *Pipe* signalling and its use cases. Following that, we discuss the related work regarding *Pipe* or LSP signalling and support for Unidirectional Technologies (UDTs).

A. IEEE 802.21

The WiBACK control plane is based on an extended version of the IEEE 802.21 [11], [12] architecture, which aims at facilitating a handover between heterogeneous access networks including wired and wireless technologies by providing link layer intelligence for the upper layers. Those technologies include IEEE 802 and non-802 networks such as those specified by 3GPP or DVB. Thus, the use of IEEE 802.21 can improve the user experience of mobile devices by enabling seamless hand-overs wherever this is supported by the underlying network environment. For this purpose, IEEE 802.21 defines a media-independent abstraction layer in the form of service primitives which provide a uniform interface to the higher layers. This cross-layer design allows for a reduced complexity as well as modularity in the design and implementation of upper layer modules or protocols in a media-independent manner, while leveraging the knowledge about the particularities of the lower layers.

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 *Management Pipes* or link-local multicast transmissions. The IEEE 802.21 messaging service, as well as the majority of the defined Media Independent Handover (MIH) primitives can also be utilized for non-handover related purposes. This includes managing local and remote radio technologies in a media independent manner. Consequently, the CARMEN project has chosen to base the WiBACK architecture design upon the general IEEE 802.21 architecture introducing new primitives or messaging service extensions where needed [13].

As depicted in Figure 2, the WiBACK Interface Management Function (IMF) extends the IEEE 802.21 Media Independent Handover Function (MIHF) with primitives specific to wireless network management, therefore the name IMF has been chosen reflecting its responsibilities beyond *Media Independent Hand-overs*. This extension to IEEE 802.21 provides a single interface for realizing MT hand-overs as well as building and managing heterogeneous wireless networks. The separation between the messaging mechanism and the protocols implemented on top of it can be compared to the Next Steps in Signalling (NSIS)[14], [15] architecture developed by the IETF. Similar to NSIS, the IEEE 802.21-based messaging mainly addresses unicast signaling while, instead, the Resource ReSerVation Protocol (RSVP)[16] was specifically designed to address the scalability issues of multicast session signaling. In the WiBACK context, unicast messaging is considered for *Pipe* signalling and also for 1-to-N multicast LSP *Trees* among WiBACK Nodes (WNs) since they are created or maintained by successively adding or removing branches.

B. WiBACK Architecture

A WiBACK [17], [18] network is managed on two time scales. On a slower time scale, centralized Topology Management Functions (TMFs) located at *Master* nodes manage *Slave* nodes, their radio interfaces and the overall spectrum resources, while at a faster time scale the stateful Capacity Management Function (CMF) assigns the available capacity to resource requests for user payload between WNs. The CMF operates on a set of logical links which is the active subset of all possible physical links managed by the TMF. Both link types are identified by their *LinkId* which consists of the source and destination link layer addresses.

The WiBACK control plane communication take place almost exclusively between the WNs hosting the TMF and CMF *Master* entities and the *Slave* WNs. Hence, it is essential for *Slaves* to maintain a reliable and resilient management connectivity to the TMF and CMF *Master* nodes. However, *Slaves* do not need to execute a routing protocol since network-wide routing state is not required. To facilitate the management connectivity, dedicated *Pipes* are configured between *Master* nodes and each of their associated *Slaves*. To increase the resilience to intermediate node or link failures, *Pipes* can be protected with backups using the MPLS Fast Reroute (FRR) feature, see [19]. Fail-over events may be signaled via 802.21 publish/subscribe-style indication messages and should be sent directly to the respective Point of Local Repair (PLR) of the affected LSP, possibly via pre-determined paths.

In order to establish, modify, or remove *Pipe* state, our proposed Pipe Management Function (PMF) works in close cooperation with the TMF since it is responsible for the setup of the initial *Management Pipes* once a new node is to be joined. It is therefore important that PMF relies on a fast and reliable protocol to setup and remove *Pipes*, which also reports the node and reason if it encounters an error during the setup procedure. Management messages are sent using the highest queuing priority, therefore repeated packet loss would indicate serious issues regarding the wireless link in question. PMF merely executes TMF or CMF decisions and can not judge the importance of a link failure or weakness. It must therefore report such incidents for TMF or CMF to take appropriate actions. To address different links and network configurations, PMF should be configurable for each individual *Pipe* setup in terms of amount and frequency of retransmission attempts. Similar functionality is, for example, provided by the RSVP[16] protocol suite, which, in IP-based operator networks, is used to facilitate flow resource reservation and management.

C. LSP Signalling

In the MPLS context, two main protocols suites have been developed to perform label distribution, namely Label Distribution Protocol (LDP) respectively Constraint-based Routing Label Distribution Protocol (CR-LDP) [20][21] and Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)[22]. As of February 2003, the IETF MPLS working group deprecated CR-LDP and decided to focus solely on RSVP-TE, which we consider here for our purposes. Conceptionally, both protocols support *explicitly routed* LSP signalling, and both assume bi-directional connectivity between neighboring nodes. Hence, most of the considerations presented in this paper regarding Unidirectional Technology (UDT) support would also apply to CR-LDP.

RSVP signals end-to-end while intermediate RSVP-capable routers may intercept and process such messages. 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[23] and RFC 5332[24] which introduced upstream-assigned labels, which 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.

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 [25], it was shown that standard RSVP performs poorly over links with higher loss probabilities, which must be considered in the WiBACK context. Reliable messaging via Message IDs was introduced with RFC 2961[26] 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 3.

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

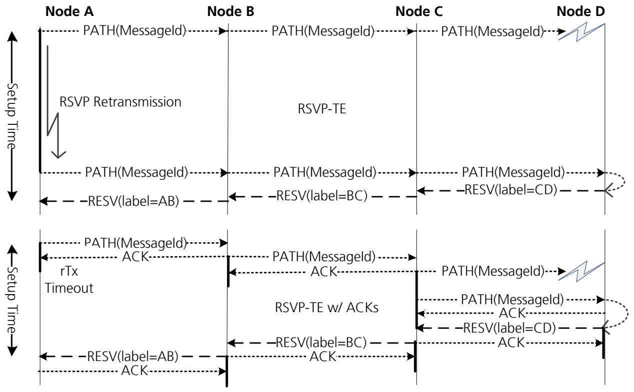

Fig. 3. In case of a packet loss, standard RSVP relies to periodic end-to-end refresh messages, while the RFC 2961 extensions allow for hop-by-hop ACKs and explicit retransmissions

would require the use of nested hop-by-hop Request/Response transactions in order to realize a conceptually similar signalling mechanism.

RSVP messages are typically forwarded via regular IP routing. In order to support Traffic Engineering (TE), forwarding along pre-computed paths can be enforced using the Explicit Route Object (ERO) [27] 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[28] 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.

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, max. latency and max. 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*.

D. Unidirectional Technologies

In RFC3077 [29], a standardized Link Layer Tunneling Mechanism (LLTM) for Unidirectional Links (UDLs) in IP networks has been develop by the IETF's Unidirectional Link Routing (UDLR) working group. LLTM specifies a mechanism to provide bidirectional connectivity between all nodes that are directly connected via a UDT, where the receive-only nodes use a tunneling mechanism to forward link layer datagrams back to send-only nodes via a separate IP connectivity. A typical tunneling protocol used in combination with LLTM is Generic Routing Encapsulation (GRE)[30]. This tunnel may encapsulate data link layer frames, hence this approach can be considered transparent to higher layer protocols. The main use case for LLTM is to provide *best-effort* virtual return links across foreign network clouds, such as the Internet, while possible QoS support for LLTM return links was studied in [31].

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.

RSVP signalling in the presence of UDTs could also be achieved by using the ERO, and either statically pre-configured host routes and static Address Resolution Protocol (ARP) or IPv6 neighbor table entries in order to *hard-code* a return path into affected nodes, or by relaying on the unnumbered links extension of RFC 3477. However, any RSVP signalling would, at least, require a minimal IP subsystem to be operational, which the TMF would need to configure during the boot strap phase. This would also require additional mechanisms to maintain such states in the case of topology changes.

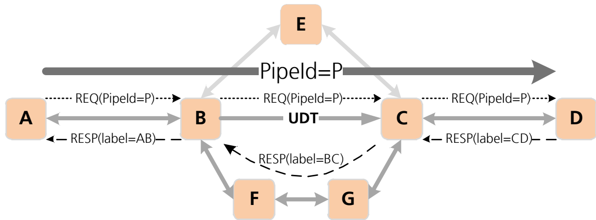

Fig. 4. PMP uses the *LinkVector* extension to signal around a UDT via source routing to signal an *Pipe* from node A to D

In [8] we have shown that for the centralized WiBACK architecture, typical distributed IP-based Network layer signalling is suboptimal. We therefore proposed to address WiBACK control plane signalling as well as seamless UDT integration below the network layer by relying on MPLS to provide node connectivity across a topology of radio links dynamically optimized and configured by the TMF. This approach requires a mechanism similar to RSVP-TE to configure *Pipe* state which may be implemented via explicit source routing among the IMFs along the path 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. Figure 4 depicts such a scenario, where the return path traverses the additional nodes F and G in order to signal *around* the UDT.

E. Summary

The soft-state RSVP suite was designed to enhance the QoS signalling of an already configured and properly working IP network built on top of bi-directional link layer technologies. The WiBACK architecture, however, requires a fundamental network forwarding state setup protocol supporting and complementing the TMF as well as the CMF in forming and managing a WiBACK network which may include UDTs and multiple possible links between any two WNs. In the centralized WiBACK architecture, distributed IP-routing state is not required, and without this, RSVP-TE can not be used without modifications. However, the concepts of RSVP-TE should be considered in the design of the Pipe Management Protocol (PMP), the WiBACK alternative for RSVP-TE.

III. Approach

Our approach to an RSVP-TE-inspired Pipe Management Protocol (PMP) for the WiBACK architecture is split into two aspects. First, we propose the *LinkVector* extension for the WiBACK *TransportService* to support source routed MIH encapsulation messages in order to allow IMF user modules to specify the exact path of MIH messages towards the destination node. Second, we describe the PMP which relies on this extension for *Pipe* signalling.

A. Link Vector Extension for WiBACK Transport Service

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* Type-Length-Value (TLV) object holding the source routed path, similar to the ERO in RSVP-TE. The actual payload of those Encapsulation message is the encapsulated original MIH message contained in a special Encapsulation TLV, see Figure 5.

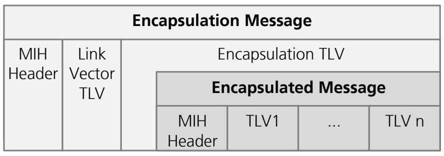

Fig. 5. The MIH_Encapsulation Primitives consist of an outer MIH header, a *LinkVector* TLV and an encapsulation TLV holding the encapsulated message

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.

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.

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 I
The PMF_InstallPipe.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
Statistics	PMP_Statistics	*Pipe* signalling statistics

TABLE II
The PMF_InstallPipe.Response primitive contains the above TLV-encoded parameters

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, in case where a Response message is to be sent as a reply to a Request message, it is assumed that the path is determined by other application specific means.

This *LinkVector* extension, though, has proven to be a crucial mechanism not only for the PMF, but also during the topology forming phase where TMF [17] instances of unassociated nodes may be required to exchange initial configuration messages.

B. Pipe Management Protocol

The Pipe Management Protocol (PMP) builds upon proven RSVP-TE concepts where possible and heavily utilizes the *LinkVector* extension. PMP is executed by the PMF which has been designed as an IMF user modules and is present at each WN.

PMF with its four use cases, *Pipe* setup, resource allocation modification and *Pipe* removal as well as fail-over signalling, is built upon five new IEEE 802.21-compatible primitives which contain the relevant *Pipe* state and QoS resource allocation information, similar to RSVP-TE. The *PATH* and *RESV* messages have been implemented via PMF_PipeSetup.Request and PMF_PipeSetup.Response primitives. 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 I and II for an exemplary overview of the information contained the PMF_InstallPipe.Request and PMF_InstallPipe.Response primitives, respectively.

On the forward path as specified by the *DownstreamLinkVec* TLV, PMP uses single hop forwarding, 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. Here multi-hop *LinkVector* forwarding is be used, since the Response message only needs to be actively processed by the WNs in the signalled path. In Figure 4, 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. The return path may either be explicitly specified using the textitUpstreamLinkVec TLV, or if that TLV is not present, it is derived by reversing the *DownstreamLinkVec*.

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 can be 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 on the respective MAC Adaptor.

For each *Pipe*, the PMF maintains a state object at each traversed WN. *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 or PMP_PipeRemove.Request message. A *Pipe* identifies an LSP and its associated QoS resources given in the form of a *TrafficSpecifications* record.

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 IV.

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 an 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 the *Pipe* state from the affected nodes directly.

C. Failover Signalling

If a failure of an underlying link is detected by the monitoring component [19], 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. In cases where the PLR is not the TMF *Master* node, no *Management Pipe* might exists 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 is 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.

D. 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 resend 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 [32], but, for a ten-hop scenario, was found to yield rather long setup times of up to 50 seconds under loss conditions. Therefore, in this article, we present a slightly more aggressive mechanism which may use multiple parallel MIH transaction in order to more quickly recover from message loss, see Figure 6. 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 6 depicts a corresponding message sequence chart of a loss-impacted *Pipe* install procedure.

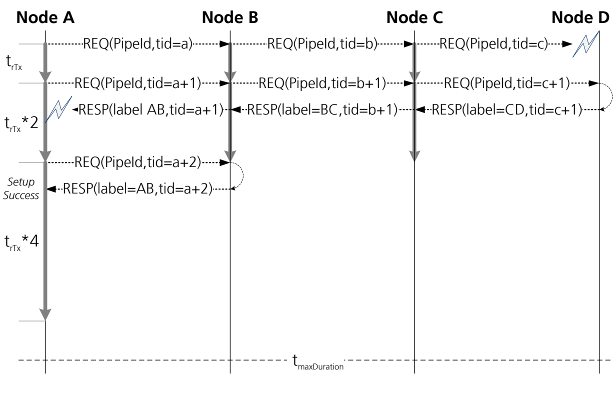

Fig. 6. MSC depicting the inter-node PMP communication focusing on retransmission timing and multiple open transactions, thus omitting the node-local resource allocation messages exchanged with the MAC Adaptor

In Figure 6, 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 unnecessarily triggered by local retransmission timer expiration due to late RESP messages further down the path. The *PipeId* in the REQ messages is used by PMF to identify the PMP session across the multiple MIH transactions. Resource allocation via the AI_LinkAllocateResource primitive is performed at each node upon reception of a successful RESP from the downstream node. If the allocation is successful the node enters the *ESTABLISHED* state end sends a successful RESP to its upstream node. If the resource allocation fails, an error is send upstream and the tear down sequence is triggered towards the already established downstream segments of the *Pipe*.

Depending on the scenario, this mechanism can be parameterized to tolerate higher latencies and loss figures or to rather 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 IV.

E. Protocol Analysis

Figures 3 and 6 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 error link. Assuming a typical PMP parameterization as evaluated in the next section, 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, since, in the current design, each *branch* of the multicast *Tree* must be signalled with a REQ and explicitly confirmed with a RESP message. Larger macro cells, such as a Digital Video Broadcast - Second Generation Terrestrial (DVB-T2) cell, may potentially cover an unlimited number of receivers. In the WiBACK architecture, however, only WNs, preferably those also acting as APs for MTs, would actively join an LSP *Tree*. This would reduce the number of *branches* to the order of hundred nodes, which may join such an LSP *Tree* on behalf of their possibly numerous MTs. Hence, the signalling overhead should be manageable as long as *Tree* memberships are rather static. In more volatile scenarios, where macro cells with hundreds of WNs are frequently created and destroyed, this aspect should be reconsidered.

IV. Evaluation

The *LinkVector* extension to support source-routed messaging has been integrated into the *TransportService* and the PMF has been implemented as an IMF user module within our WiBACK testbed according to the specifications provided in this article. At each hop, PMP negotiates the downstream or upstream MPLS labels, installs the corresponding LSP state and also reserves the requested *Pipe* resources with the MAC Adaptor of the respective outgoing interface. If an error occurs, PMF reports the first node in downstream direction back to the ingress node.

The software is built upon our C++ Simple and Extensible Network Framework (SENF)[2] framework which also provides a real-time network emulator component supporting the mixed use of emulated and hardware interfaces. This allows us to evaluate the same binary code on emulated nodes, real Linux-based nodes or a combination of both while using proven external measurement tools for validation. For emulated interfaces, random packet loss and a fixed link latency and can be introduced [33]. The random loss module is based on a *Mersenne Twister* implementation and provides a rather uniform packet loss distribution including shorter burst losses.

The following results have been obtained in real-time emulation mode. Since the real-time emulation is running on a multi-core Linux host, the operating system may introduce slight random scheduling latencies as well as variances thereof which actually help to expose the real-world behavior of the evaluated implementation. Such variances are exhibited, for example, in Figure 7 with loss and latency set to 0 ms. In a typical simulation environment, no variances would be expected here. This accumulated latency introduced by the internal protocol and emulation processing amounts to about 5 ms accumulated over eleven nodes and 20 hops, ten in the downstream and ten in the upstream path.

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

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

            F<--->G
           /       \
     A<--->B---UDT-->C<--->D

* Unicast UDT Pipe Setup (A -> B -> C -> D): OK
* Multicast Pipe Setup (A -> B -> F -> G -> C -> D): OK
* Performing unicast ping test with 200 packets: PASSED
* Performing multicast ping test with 200 packets: PASSED
* Unicast Pipe teardown: OK
* Multicast Pipe teardown: OK
```

As an initial validation, we verified the basic PMP functionality, the a) setup of an *Pipe* around a UDT, and the b) setup of an upstream-assigned multicast *Pipe*. Both tests were run in an emulated 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 above.

Following the initial successful validation, we evaluated the PMP performance in a controlled environment under varying typical packet loss and link latency conditions. Since the WiBACK architecture supports hop distances of up to ten hops, our PMP signalling evaluation scenario consisted of eleven emulated nodes, one GW and one AP node with one radio interface each and nine regular WNs with two radio interfaces. The interfaces were assigned orthogonal channels, thus forming a concatenated 10-hop chain of nodes.

To start off, an initial measurement was run to determine a reasonable default $t_{maxDuration}$ that fits a typical heterogeneous, i.e 802.11, 802.16, Digital Video Broadcast - Terrestrial (DVB-T), WiBACK scenario. Our criterion was that 95% of all setup attempts should succeed up to a conservative maximum per-hop latency of 50 ms and a per link error rate of 10%. For a WiBACK 802.11-based scenario in our outdoor testbed, the maximum link latency in the Management traffic class, even under heavily loaded link conditions, has been determined to be roughly 2 ms. For our WiBACK 10-hop benchmark 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 t_{rTx} was set to 50 ms and the $t_{maxDuration}$ was determined to be about 2000 ms 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.

Figure 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%. As expected, the link setup times increase linearly

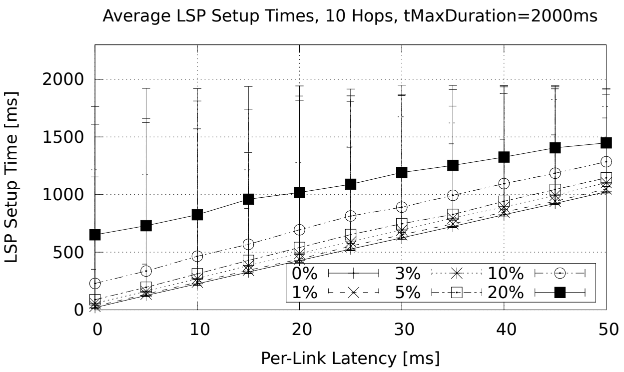

Fig. 7. *Pipe* setup time over per-link latency under varying per-link loss fractions

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 2000 ms due to the predetermined limit. Figure 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 2000 ms 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.

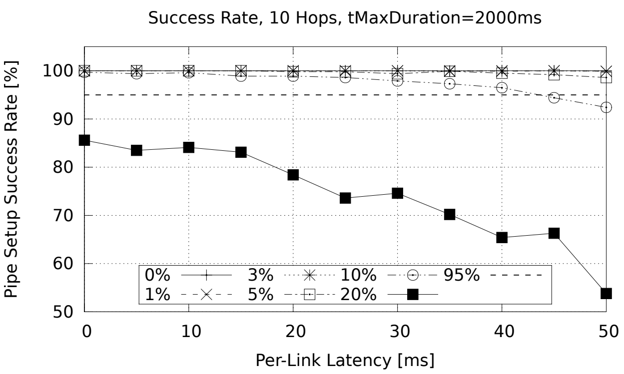

Fig. 8. *Pipe* setup success rate over per-link latency under varying per-link loss fractions

Figures 9 and 10 depict 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 2 ms 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%.

The results show that the position of the faulty link in the chain has no significant impact on either the average setup time nor the success rate. The results also show that with the chosen default parameterization, a single hop loss probability

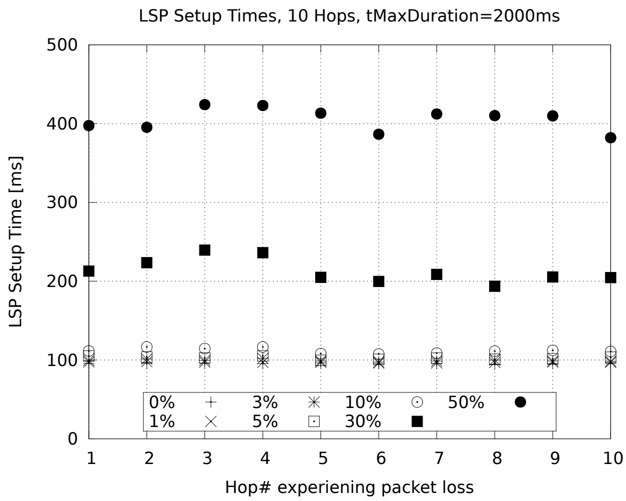

Fig. 9. *Pipe* setup time over hop distance of errored link

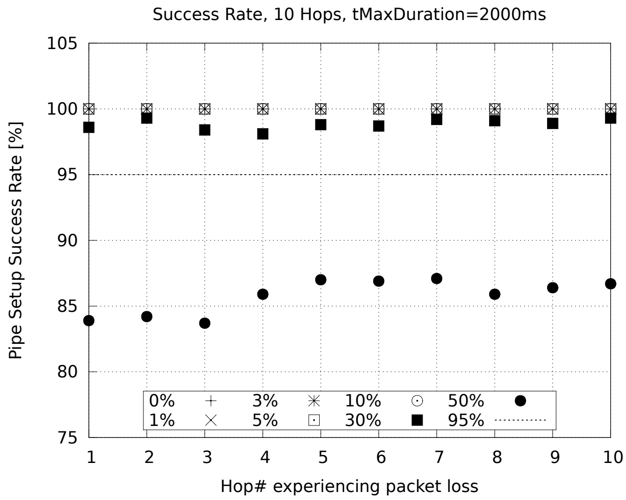

Fig. 10. *Pipe* setup success rate over hop distance of errored link

of 30% can be tolerated with a success rate of almost 100%, resulting in a fast average setup time of about 200 ms. Even assuming 50% loss, the success rates are still above 80% which might require multiple setup or tear down attempts but still allows TMF to reach the affected node to, for example, trigger corrective actions.

Measurements as well as statistics collected in our multi-radio testbeds [17], [18] consisting of long-distance IEEE 802.11a, sub-Ghz 802.11ah as well as Ethernet links, confirm the emulation results regarding setup times and reliability. Due to the rather low latency of the IEEE 802.11 links, $t_{maxDuration}$ is set to 1000 ms here and average setup times range from of 5 ms to 100 ms depending on hop count, link stability and system load.

Concluding, we can state that our PMP could be shown, in emulated and real-world measurements, to be fairly resilient against random loss pattern, even under higher loss ratios. Under sub-optimal link conditions, larger burst losses or link outages in the order of $maxDuration$ may occur and should be handled properly by PMP. Therefore, we plan, as a future work item, to evaluate PMP under such link conditions in order to further assess its resilience and to derive alternative parameterizations to be applied in such situations. Given the optimization goal of the WiBACK architecture to exclude such unstable links from its set of active links, such situations should be considered as exceptions. Hence, a second pipe signalling attempt with a temporally adjusted parameterization should be preferred over a permanent adjustment which would require an increased $maxDuration$.

V. Conclusion and Future Work

Our proposed PMP as well as the *LinkVector* extension introducing source routed MIH messages have been shown to support a fast and reliable signaling of *Pipe* state within a WiBACK network while transparently signalling *around* UDTs. The results obtained here may be of interest for further studies of RSVP-TE-like signalling of LSP state in data networks involving unidirectional satellite links or, in general, for networks experiencing noticeable packet loss.

Due to the integration of the source routing mechanism via MIH encapsulation messages, the WiBACK control plane has been extended by a powerful link layer independent messaging mechanism. As a future work item, we plan to also deploy it for the delivery of re-routing *Indications* for backup *Pipe* signalling.

Multicast *Tree* signalling in situations of rather volatile membership changes requires further investigations, possibly exploiting cases where larger quantities of WNs are simultaneously joined into or removed from 1-to-N cells while the associated resources only need to be allocated once.

VI. Acknowledgment

This work has been funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01 BU 1116). The authors alone are responsible for the content of this paper.

References

[1] 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.

[2] S.-B. Lee, A. Smeaton, and G.-M. Muntean, "Quality-oriented multiple-source multimedia delivery over heterogeneous wireless networks," *Broadcasting, IEEE Transactions on*, vol. 57, pp. 216 –230, june 2011.

[3] G.-M. Muntean and N. Cranley, "Resource efficient quality-oriented wireless broadcasting of adaptive multimedia content," *Broadcasting, IEEE Transactions on*, vol. 53, pp. 362 –368, march 2007.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] J. Manner, R. Bless, J. Loughney, and E. Davies, "Using and Extending the NSIS Protocol Family." RFC 5978 (Informational), Oct. 2010.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

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

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.