Integration of QoS Parameters from IEEE 802.11s WLAN Mesh Networks into Logical P2P Overlays

Michael Rethfeldt, Peter Danielis, Björn Konieczek, Felix Uster, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics and Computer Engineering
18051 Rostock, Germany, Tel.: +49 381 498-7269
Email: michael.rethfeldt@uni-rostock.de

***Abstract*—Adopted in late 2011, IEEE 802.11s comes as the first industry standard to enable vendor-independent and interoperable WLAN mesh networks. Featuring automatic device interconnection and routing, they provide a higher scalability, flexibility, and robustness compared to common centralized WLAN infrastructures. The 802.11s standard defines mandatory support of the Hybrid Wireless Mesh Protocol (HWMP) and Airtime Link Metric (ALM) for MAC-layer routing. While 802.11s covers the physical network underlay, Peer-to-Peer (P2P) protocols are an equivalent on the application level. In contrast to centralized client/server communication, they establish a fail-safe and scalable logical network overlay for, e.g., distributed content sharing, streaming, search, or synchronization. Thus, P2P networks exhibit many of the characteristics of physical WLAN mesh networks. It is obvious to consider joint solutions, where both technologies are combined to leverage future distributed local area wireless applications. Nevertheless, common P2P protocols, such as BitTorrent, do not consider the structure of the physical underlay while performing topology management. Furthermore, they are primarily designed to be used over wired communication networks such as large parts of the Internet. When deployed over WLAN mesh networks with their quickly varying channel conditions, BitTorrent shows severe performance drawbacks. We present a cross-layer approach based on 802.11s and BitTorrent, that optimizes application layer peer selection by considering the mesh standard's routing metric ALM. Our solution was implemented and evaluated in a real-world test bed. Results show that average download time can be reduced by up to 20 % already in small network setups.**

I. INTRODUCTION

Endorsed by the increasing variety and affordability of wireless consumer devices, complex networks can be established to provide distributed, content-centric services in places of high node density and mobility. The widespread IEEE 802.11 WLAN (Wireless Local Area Network) standard family [1] is already omnipresent in today's home networks, office environments, and public facilities. In contrast to currently prevalent "infrastructure" deployments based on central Access Points (AP), decentralized WLAN mesh networks are characterized by their flexible, scalable, and fail-safe network topology [2]. The standard amendment IEEE 802.11s was adopted in late 2011 and initially adds mesh functionality to the WLAN MAC layer. In an 802.11s network, every mesh node provides data forwarding and routing capabilities [3]. Nodes within radio range automatically interconnect and establish paths to selected targets. Economic network extension is simply possible by bringing in additional mesh nodes. Thereby, path maintenance follows a radio-aware link metric that takes the properties of the wireless medium and MAC protocol into account. Thus, the network becomes more robust to changes in node availability, density, and varying link qualities. For interoperability, 802.11s defines the *Hybrid Wireless Mesh Protocol* (HWMP) and the *Airtime Link Metric* (ALM) as default combination for path selection [4].

P2P technology is used to develop distributed, scalable, and failure-resilient network applications. It overcomes the drawbacks of centralized client/server communication, that inherently includes a Single Point of Failure (SPoF) [5]. Possible applications include multimedia streaming, Voice over IP, content search, synchronization, or file sharing. P2P networks are implemented as logical overlay networks on top of a given physical underlay. Regarding robustness and scalability, they share many similarities with wireless mesh networks [6]. Therefore, logical P2P networks are particularly suitable as candidate application above a mesh underlay. A combined solution features scalability and robustness both on logical application and physical network level [7]. In a WLAN mesh network, a P2P overlay could be used to distribute firmware and configuration updates or network statistics for management, or to implement live streaming as application-layer multicast.

However, common P2P file sharing protocols, such as the well-known and widespread BitTorrent (BT) [8], are designed to be used over the Internet. Thus, logical topology management is optimized for wired networks, i.e. reliable links and stable channel conditions. On the contrary, link quality may change quickly in wireless mesh networks. Neighboring devices within radio range interfere on the same channel and suffer from contention-based medium access overhead. While nodes on P2P application level always appear as single-hop neighbors, they may be multiple hops away in the physical mesh underlay. Application-layer traffic over multi-hop paths puts stress on intermediary nodes that only forward data to an overlay endpoint. These nodes may also be interested in the forwarded data, but are not optimally chosen as destination in the overlay. This situation induces redundant data transmission and channel access, compared to a strategy where overlay traffic is kept physically local. Thus, when deployed over

WLAN mesh networks, the default BT protocol reveals serious performance drawbacks, when not considering the physical network topology during peer selection [9].

Consequently, we have developed a cross-layer solution to overcome the mismatch between a logical P2P overlay based on BitTorrent and a physical WLAN mesh underlay based on IEEE 802.11s. We modified the default BitTorrent choking algorithm to use the MAC-layer Airtime Link Metric (ALM) of 802.11s as criterion for peer selection. Comparing different underlay technologies, we show the need for BitTorrent optimization. Measurement results for the default and the modified choking algorithm show, that average download time can be reduced by up to 20 %, already by directly applying ALM for peer selection and combining it with a limited neighborhood scope.

The remainder of this paper is organized as follows: Section II first outlines the basic principles of the IEEE 802.11s WLAN mesh standard and its Linux implementation *open80211s*. Then, we introduce the BitTorrent P2P file sharing protocol and its peer selection algorithm. Finally, we point out the mismatch between logical P2P overlay and physical mesh underlay. In Section III, we discuss related work in combining WLAN Mesh Networks and BitTorrent. Section IV illustrates the design of our cross-layer solution and the modifications we applied to the default BitTorrent peer selection. In Section V, we discuss the measurement results that were obtained in a real-world test bed. Finally, we give a conclusion in Section VI and briefly state possible improvements and approaches for future research.

II. TECHNOLOGICAL BASIS

A. IEEE 802.11s

As first common industry WLAN mesh standard, IEEE 802.11s was ratified in September 2011 [1], [3], [10]. It enables vendor-independent infrastructure-less multi-hop communication based on the widespread WLAN technology. The central Access Point (AP) role is delegated to all distributed nodes, as each Mesh Point (MP) in an 802.11s network supports frame forwarding and path selection. Mesh functionality is directly integrated into the 802.11 MAC layer specification. Any changes to the underlying physical layer are avoided and mesh support can be easily added on the driver level to be used with existing WLAN hardware. The common 802.11 data and management frames have been extended to enable automatic peer discovery, peer link establishment, frame forwarding and routing. Figure 1 shows 802.11s in the ISO/OSI stack. Since path selection is handled on the MAC layer, it is supposed to generate less overhead than existing network-layer mesh routing protocols [10], [11]. Moreover, mesh operation becomes transparent to all higher layers.

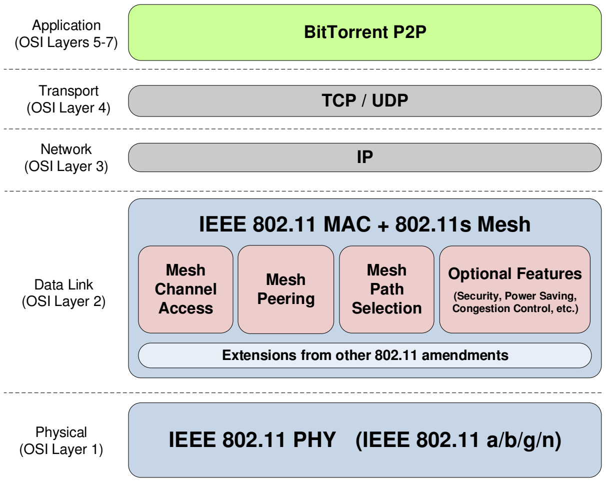

Fig. 1. IEEE 802.11s in the ISO/OSI stack

To ensure interoperability, every MP must support the *Hybrid Wireless Mesh Protocol* (HWMP) and the *Airtime Link Metric* (ALM) [4] as mandatory default combination for path selection. HWMP is based on the reactive Ad-Hoc On-Demand Distance Vector (AODV) routing protocol [12]. Analogous to AODV's route request (RREQ) and reply (RREP) messages, HWMP defines path request and reply frames (PREQ/PREP). Optionally, a tree-based proactive routing mode is available. This is designated for paths to static mesh nodes such as gateways and may be used along with the reactive on-demand mode [10], [13]. If a source MP needs to establish a path to a destination MP on-demand, it broadcasts path requests (PREQ) to all nodes within radio range. Each intermediary MP along a possible path adds the ALM value of the receiving link to the metric field in the PREQ frame and forwards it. Only PREQs with better (smaller) ALM value and higher sequence number are forwarded. Once the destination is found, a PREP is send as unicast along the best path, found in the PREQ flooding phase. When the PREP finally reaches its PREQ source, the bidirectional path is established.

Mesh paths are chosen according to the link metric, which is ALM by default. It represents the costs for transmitting a frame over a specific link in the mesh network by considering technology parameters of the WLAN physical layer and the wireless medium. The so-called *airtime cost* (c_a) is calculated as follows:

$$c_a = \left[O_{ca} + O_p + \frac{B_t}{r} \right] \cdot \frac{1}{1 - e_{fr}} \tag{1}$$

O_{ca} and O_p are constants for the channel access and MAC protocol overhead. B_t is the test frame size. By default, a frame size of 8192 bit is used. r denotes the test frame data rate, given in Mbit/s, whereas e_{fr} denotes the expected frame error rate. The estimation of e_{fr} as well as the values of the overhead constants are not predefined by the 802.11s standard but left open to vendor implementations [14].

As usual for distance vector protocols, an MP using HWMP only knows its direct neighbors and nodes in multi-hop distance, to which communication has been explicitly initiated, e.g., on higher layers. Path tables on every MP contain forwarding rules to target nodes via neighbors ("next hops"). These rules are periodically updated in small intervals within their expiry time. To every target, only the best next hop with smallest ALM is kept. Since a path commonly consists of multiple consecutive links, ALM works as cumulative metric. Although shorter paths often result in a smaller metric, longer paths may sometimes be preferred. This occurs if intermediate nodes on a longer path have better peer links in total (higher data rate and/or less frame errors) and thus the cumulative path cost is smaller than that of the alternative shorter path. Thus, ALM represents both overall link quality and path length. The studies in [15] and [16] show that the 802.11s mandatory default combination of HWMP and ALM outperforms other routing protocol combinations with the metrics ETX and ETT in different traffic and network scenarios.

B. *open80211s*

The Linux project *open80211s* [17] is currently the most advanced open-source reference implementation of 802.11s. It already satisfies all mandatory and various optional parts of the standard. The code base is integral part of the *mac80211* kernel module, i.e. the software MAC layer of the Linux WLAN stack [18]. Since some parameters in ALM calculation are left open to vendor implementations, open80211s provides own variants for frame error rate estimation and overhead constants (see equation 1). While O_{ca} and O_p are summarized to 1, the data rate r (of the last unicast frame transmission) depends on the rate control algorithm (RCA). In current Linux kernels, *minstrel* is used as default RCA [19]. The estimation of e_f follows equation 2:

$$e_f\left[t_k\right] = \frac{80 \cdot e_f\left[t_{k-1}\right] + 5}{100} + 20 \cdot \delta\left[t_k\right] \tag{2}$$

The Boolean parameter $\delta\left[t_k\right]$ indicates a successful (0) or failed (1) last frame transmission. The error rate results in values between 0 and 100 (the right operand is truncated). It is then normalized to range from 0 to 1, before being used in the final ALM calculation (equation 1). As e_f is updated on every frame transmission, always the most recent value is available on a triggered path refresh. Real-world evaluations have already demonstrated the HWMP performance of open80211s [14], [20].

C. *BitTorrent*

Currently, BitTorrent (BT) is the most prevalent P2P network for file sharing. It contributes about 6 % to the overall Internet traffic [21]. Its popularity results from its capability of achieving high download rates, which is usually the main interest of users. For each file to be shared, one specific logical network is created. To search for a file, usually a web site is contacted to get a *.torrent* metadata file. This file contains, among other things, the address of a *tracker* and information about the file to be downloaded. The tracker is contacted to get a list of BT users (peers) holding the file (or parts of it – so-called *pieces*, which are further subdivided into *sub-pieces*). Thereby, the pieces and sub-pieces can be of different size, e.g., ranging from 32 kBytes to 32 MBytes. All peers, which are interested in this file, form a so-called *swarm*. Complete downloaders serving the whole file are called *seeds*. Incomplete downloaders are called *leechers*. BT peers start to download pieces in *random* order and change to *rarest first* order after the first piece is completed [22]. Thereby, they follow the *strict priority* of solely requesting sub-pieces of a particular piece before sub-pieces from the next piece.

For selecting other peers who may download a piece, each peer applies the so-called *choking algorithm* [22]. In a nutshell, this is a variant of the tit-for-tat strategy. Only peers offering sufficient upload performance are given download time in return (they are *unchoked*). The choking algorithm to determine a peer that may download pieces is executed periodically because upload performance of peers can change quickly. As an exception, each peer has an *optimistic unchoke* available to unchoke one other peer regardless of his upload performance. This is performed to increase piece diversity and parallelism of data transmissions in the network. By that, faraway peers can also contribute in redistribution of pieces, although they would not have been chosen following the upload rate criterion. By default, the choking algorithm is executed in a 10 seconds period. Given the number of upload slots N (4 by default), N-1 peers are unchoked as a result. Every 30 seconds a new optimistic unchoke is selected for the remaining upload slot.

Once a peer has finished its download, it may decide to stay in the network for a while (lingering), operating as a seed. During this time span, it only uploads pieces preferring peers, to which it has the best upload rates.

D. *Mismatch between P2P overlay and mesh underlay*

Common P2P file sharing protocols, such as BitTorrent (BT), do not consider the structure of the physical underlay while performing topology management and peer selection. They are optimized for wired networks, which show reliable links and stable channel conditions. In wireless mesh networks, however, link conditions may vary quickly due to distance, mobility, or the occurrence of obstacles, obscuring the line of sight between nodes. Devices within radio range interfere on the same channel and suffer from contention-based medium access overhead. Moreover, in a multi-hop scenario the cost of a transmission increases, as each forwarding step requires additional channel access and is again subject to possible frame errors and collisions. Therefore, deploying BT over a wireless mesh underlay is challenging and the selection of suitable logical peers has an immense effect on network performance.

While BT peers on application level always appear as direct neighbors, they may be multiple hops away from each other in the physical mesh underlay and wireless link quality can differ severely. In Figure 2, this is shown for Peer 1 and 2,

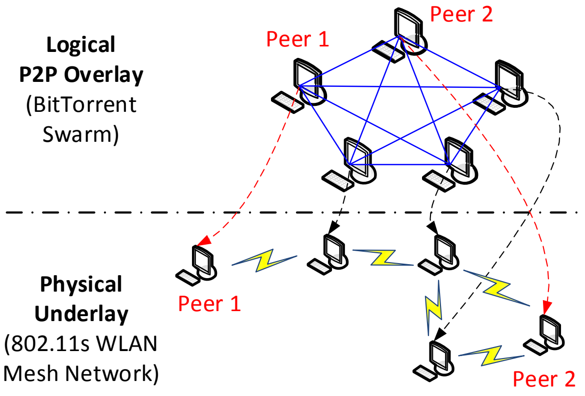

Fig. 2. Mismatch between logical overlay and physical underlay

which are connected by a three-hop mesh path in the underlay but maintained as immediate neighbors in the BT overlay. Imagining the intention of content synchronization starting from Peer 1 as a seed, selecting Peer 2 as a leecher would be the worst case, as this generates unnecessary stress on the intermediary nodes. Instead, physically close peers should be usually preferred, which allows for collaborative distribution of pieces while traffic is kept local. Moreover, peers that are endpoints of low-quality paths should be avoided and saved up to be served by more suitable peers in the distribution process. Both aspects can be faced by the 802.11s standard's routing metric ALM (see Section II-A), as it expresses both path length and overall link quality. Thus, it is a better criterion for BT peer selection in 802.11s mesh networks compared to the default BT choking algorithm.

III. RELATED WORK

There are already some approaches, which intend to optimize the performance of the BitTorrent (BT) protocol when applied over mobile ad hoc networks.

In [23], a cross-layer solution is presented whereby the BT tracker receives data from the IETF Application Layer Traffic Optimization (ALTO) infrastructure in the form of distance information of the Open Link State Routing (OLSR) proactive mesh routing protocol. As a result, the tracker returns a list of peers sorted after this distance information rather than returning random peers. However, the metric used for the cost of links is the Expected Transmission Count (ETX), which solely minimizes the expected total number of packet transmissions to successfully deliver a packet to the destination. Instead, we provide BT with information in the form of the ALM metric, which allows for inclusion of WLAN technology-specific parameters and thus is a better measure for the link quality. Moreover, as support of HWMP and ALM were defined as mandatory in the 802.11s standard, ALM will always be available on each compliant node, in contrast to other metrics.

The BitHoc project aims at adapting BT to wireless ad hoc networks mainly by restricting the communication to peers, which are only some hops away from each other. The authors suggest to realize the neighborhood scope restriction by using the TTL value for reducing routing overhead [24]. The results show that the overall download time can be reduced and the throughput can be improved when using a modified choking algorithm and piece selection strategy in combination with restricting neighborhood scope. Thereby, the authors still allow optimistic unchokes so that not only adjacent peers get the chance to download but also a few remote peers. However, the ALM metric represents a better quality metric than only the number of hops, as multi-hop paths may occasionally be preferable compared to overloaded one- or two-hop paths.

The work described in [25] is one of the few publications, which apply a P2P network on top of an 802.11s mesh network, and discuss possible performance optimizations by using cross-layer approaches. Thereby, the ALM metric is passed to a generic software framework that can be used as API for P2P applications. However, no specific P2P protocol, such as BT, is evaluated and no effect of ALM utilization is measured. Moreover, the 802.11s routing protocol HWMP is required to be heavily extended on each peer, e.g. by vendor-specific frame fields. On the contrary, we want to avoid MAC-layer modifications to ensure interoperability with standard 802.11s.

The work in [26] proposes a distributed management protocol for mobile P2P networks to, e.g., replace the centralized tracker in BT. As a result, the peers organize themselves in a shared tree dedicated for disseminating membership information. By using ad hoc routing information, peers construct and adapt their logical links in the tree with regard to the current network topology. However, this requires adding structure to unstructured P2P networks following routing characteristics. Furthermore, traffic overhead is created for maintaining this structure. Instead, in our approach the BT protocol is only slightly modified to select proximate peers by means of a new choking algorithm. Thereby, no modification of the construction algorithm is necessary.

IV. CROSS-LAYER APPROACH

As depicted in Section II-D, applying P2P protocols such as BT over WLAN mesh networks reveals a mismatch between logical overlay and physical underlay. To address this, we follow a cross-layer approach that performs an integration of 802.11s HWMP information (links and paths with ALM) into the BT protocol and its choking algorithm. As a consequence, the physical mesh topology is considered during logical peer selection. Figure 3 shows the design of our cross-layer solution. The software components *Mesh Management Framework* and *BitTorrent Client* are explained in the following.

A. *Mesh Management Framework*

The first part of our solution is based on a management framework for 802.11s mesh networks that we developed at the University of Rostock [27]. It was written in Java for flexible deployment on heterogeneous platforms. Running on each mesh node, the framework encapsulates the features of the

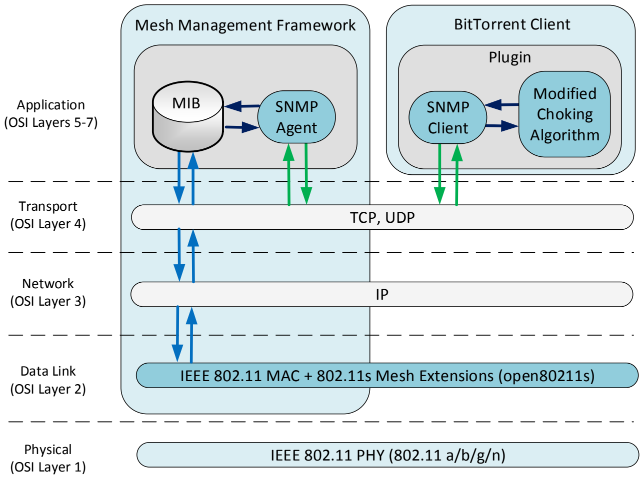

Fig. 3. Proposed cross-layer approach

Linux kernel-level 802.11s implementation *open80211s* (see Section II-B) and several CLI tools in its platform-dependent part, which is interchangeable for possible future 802.11s implementations. In a platform-independent part, status information and configuration functions are represented by Java classes. On this abstraction level, our framework addresses automatic mesh network initialization. The bootstrapping process includes MAC- and IP-layer auto-configuration to ensure node availability and to enable communication between overlying applications. Each node is equipped with fall-back procedures to repeat certain configuration steps on occurrence of errors. Furthermore, every node runs an integrated agent for SNMP (Simple Network Management Protocol), a widespread industry network management protocol. Local status information and configuration functions are provided as SNMP data model and can be queried by a corresponding SNMP client. Thus, SNMP serves as interface to easily manage the mesh nodes remotely and to combine their otherwise limited network view (see Section II-A) to a global scope. Our framework further differentiates between an *Agent* and *Manager* role. The latter additionally supports the complementary SNMP client side, including the remote query of status information and configuration of nodes. It also provides a DHCP server for IP address distribution and NTP server for time synchronization.

The Linux kernel 802.11s implementation on every node maintains data structures for link establishment and HWMP path selection [17]. The *station list* and *mesh path list* contain all physical links and logical paths from a mesh node's viewpoint. While every station list entry holds the information of a direct link to a node's one-hop neighbor, any forwarding rule in the mesh path list includes next hop, target node, and ALM value. When using default settings, path information are updated every second. A path expires after five seconds, e.g., path establishment is re-triggered. These information are reflected as SNMP objects in our management framework. They can be queried either by an external or any local application, as long as it supports SNMP client functionality. Thus, SNMP is used as an interface to pass the MAC-layer link and path information, gathered by the mesh management framework, to a BT client application, running on the same machine (see Figure 3).

B. BitTorrent Client and Plugin

The open source software *Vuze* (former Azureus) is one of the most popular BT clients available [28]. It is written in Java and can be easily extended, as it provides a flexible plugin API. We used a legacy program version (Vuze 4.3.1.4) to be compatible with an existing open source plugin that served as template for integrating our peer selection optimization. The Vuze plugin called ”BASS” originates from an external research project [29]. Its vanilla version already includes dummy code to replace the default peer selection of Vuze. However, the initial BASS plugin state only performs logging functionality and makes no changes to the BT choking algorithm. We integrated our modifications into the plugin skeleton code. When the plugin is loaded, it replaces the default *Unchoker* implementation of Vuze. To access the physical mesh link and path information inside the local mesh management framework on a node, we integrated an SNMP client into the plugin. Up-to-date mesh status information are queried periodically every 8 seconds, to be available each time the BT choking algorithm is triggered (every 10 seconds).

C. Modified Choking Algorithm

As depicted in Section II-C the choking algorithm handles BT peer selection. The algorithm periodically grants a predefined number of upload slots to interested peers. We replaced BT's default unchoking criterion, the upload rate of peers, by the ALM metric of the mesh path to it. ALM (see Section II-A, equation 1) represents the frame transmission time cost in microseconds. Thus, smaller metric values are preferred and the list of interested BT peers is now sorted in ascending order. According to equation 1, ALM considers parameters of the wireless channel as well as the current frame transmission and error rate. Its cumulative nature also reflects the path length, but additionally enables the consideration of fast multi-hop and overloaded single-hop paths. Thus, we assume an improvement in overall download performance already when directly applying ALM as unchoking criterion in an 802.11s mesh network. Especially in larger mesh setups with many multi-hop paths, this approach is expected to keep network strain local by choosing proximate peers to upload pieces to.

Apart from that, we did not change the default timing of the BT choking algorithm (10 s and 30 s for optimistic unchokes, respectively) and number of upload slots (4 = 3 + 1 optimistic unchoke) [22]. The fourth slot for the randomly chosen optimistic unchoke is explicitly left untouched because it facilitates piece diversity and parallelism in the BT swarm, as stated in [24].

In smaller mesh setups with only few and rather short multi-hop paths, peer proximity and robustness to small metric deviations between different paths should be further enforced within the choking algorithm. Consequently, we added the

TABLE I
VUZE BITTORRENT SETTINGS

Parameter	Value
File size	65 MB
Piece size	256 kB
Unchoke period	10 s
Optimistic unchoke period	30 s
Upload slots	4 (3+1)

TABLE II
OPEN80211S HWMP SETTINGS

Parameter	Value
HWMP frame TTL	31
Max. PREQ retries	4
Path refresh time	1000 ms
Min. discovery timeout	100 ms
Active path timeout	5000 ms
PREQ min. interval	10 ms
HWMP net diameter traversal time	50 ms

option to limit possible peers to only those in one-hop distance, while still ordering them by ALM. This filtering step can be performed on every node. In fact, each mesh node can differentiate between direct neighbors (contained both in *station list* and *mesh path list*) and nodes in multi-hop distance (only contained in the *mesh path list*) out-of-the-box, as given by the 802.11s standard's default path selection protocol HWMP. This already enables one-hop limitation solely by relying on standard features. Complete path knowledge for a more fine-grained hop count filtering would require aggregation of all distributed path information on every node. This was not considered due to the implied network overhead.

For evaluation of the three choking algorithm variants (default BT, ALM-based, ALM-based & limited to single-hop peers), we integrated them in the Vuze plugin code and made them switchable by an external configuration file.

V. EVALUATION IN A REAL-WORLD TEST BED

We established a real-world test bed with 8 nodes to realistically evaluate our solution. It comprised a notebook (1.6 GHz Core i5-4200U dual-core CPU, 8 GB RAM, Ubuntu 14.04 LTS, Kernel v3.13) and 7 Raspberry Pi (RPi) model B single-board computers (700 MHz ARMv6 CPU, 512 MB RAM, Raspbian Linux, Kernel v3.12) [30]. Each node was equipped with an USB WLAN adapter (Buffalo WLI-UC-GNM), operating in the 2.4 GHz ISM band in 802.11g mode. The adapters rely on the Ralink chipset driver *rt2800usb* that supports the open80211s mesh extensions [31], [32].

A measurement run consisted of the distribution of a 65 MB video file to all nodes in the BT swarm, which is a sufficient size to show the effect of our modifications to the BT choking algorithm. Measurements were performed for each of the three choking algorithm variants (see Section IV-C). Thus, the modified Vuze client and BASS plugin (see Section IV-B) were installed on every device. The notebook always operated as initial BT seed and tracker, the RPis as distributed leechers. Every leecher was configured to become a seed as soon as it has received all pieces of the file. The lingering time was set to its maximum value to ensure a seed remains in the BT swarm for the complete measurement duration.

Measurements were initially performed comparatively on different underlays (see Section V-A). In a second step, a multi-hop mesh setup was used (see Section V-B). The 802.11s mesh operation and bootstrapping, including IP address configuration, were performed by our management framework running on each node (Manager role on the notebook, Agent role on the RPis). Table I shows the basic BT parameters that were used for all measurements. Apart from that, no upload/download speed limitations were defined on BT client side. Table II shows the HWMP related open80211s settings that were used in the mesh setups.

Additionally, *tshark*, a CLI version of *wireshark*, was executed on all nodes to capture incoming and outgoing TCP packets for BT traffic analysis [33]. After each completed file distribution to all peers in the BT swarm, the Vuze log files and traffic captures of all nodes were collected. Using these, the required time to receive the whole file was determined for every peer.

A. Comparison of different underlay technologies

Initial measurements were performed on different underlays to retrieve reference times for the achievable BT performance, depending on the kind of physical LAN setup. First, all peers were connected by a Fast Ethernet switch (D-Link DES-1008D, 10/100 Mbit/s gross bandwidth). In a second setup, the switch was replaced by a WLAN AP (ASUS WL-500gP, 54 Mbit/s gross bandwidth in 802.11g mode). Here, all nodes were placed within 1 m distance to the AP, their WLAN adapters configured in common infrastructure mode. Naturally, only the unmodified BT choking algorithm was used in these first two setups. As our optimized choking alternative relies on ALM, it is only applicable in a mesh underlay.

Finally, a small 802.11s mesh network was set up. All nodes were still placed as in the AP case, i.e., within direct radio range to each other, to cover the same area as the former WLAN infrastructure. Due to the small inter-node distance, only single-hop mesh paths were established with little divergence in ALM values. In the case of the single-hop mesh network, measurements were performed both for the default BT choking algorithm and the optimized variant with ALM as peer sorting criterion.

For any of the described combinations of underlay and choking algorithm, 5 measurement runs were performed and the required download time was averaged over all peers. Figure 4 displays the results, including the 95 % confidence intervals. In the Ethernet setup, it took an average of 100 s on each node to download the 65 MB file. As expected due to lower net bandwidth and additional wireless channel access overhead, download time increased for the AP setup to an average of 435 s, i.e. it was more than 4 times slower than

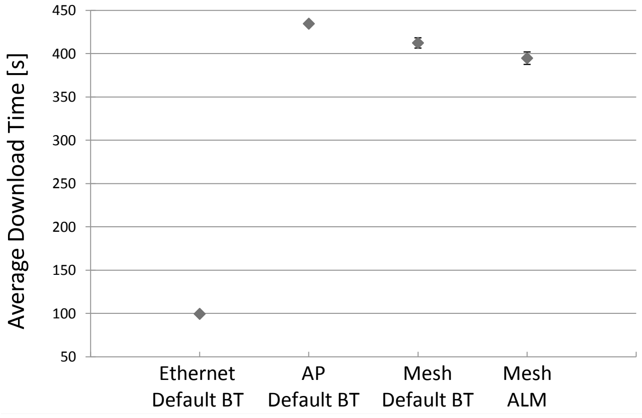

Fig. 4. Average download time for different underlay networks

in the wired network. Subsequently replacing the WLAN infrastructure by an 802.11s mesh network without changing the node placement, download time improved again by 5 % to an average of 412 s. This reveals the competitiveness of 802.11s to the likewise range-limited, AP-based setup. Eventually, using plain ALM as criterion for peer selection already resulted in further improvement by 4 % to an average of 395 s, even for the mere presence of one-hop paths with small link quality variations that are otherwise not considered by the default BT algorithm.

B. Measurements in a multi-hop mesh setup

Consequently, as this is the common usage scenario for WLAN mesh networks, we performed similar measurements in an 802.11s setup of wider network diameter, that is not coverable by a single AP anymore. Nodes were placed in different rooms spread over two floors of our institute building, as depicted in Figure 5. The higher inter-node distance and signal loss due to walls and objects lead to considerable ALM fluctuations and the establishment of multi-hop paths. On the one hand, this setup shows the flexibility of extending the coverable area of the wireless backbone merely by node placement. On the other hand, the delay caused by frame forwarding on intermediary nodes and the varying link quality also imply a degradation in network performance.

We evaluated the three choking algorithm variants described in Section IV-C: default BT mode (1); peers ordered by ALM (2); only one-hop peers allowed, ordered by ALM (3). Again, 5 measurement runs were performed for each variant and the required download time was averaged over all peers. Results are displayed in Figure 6.

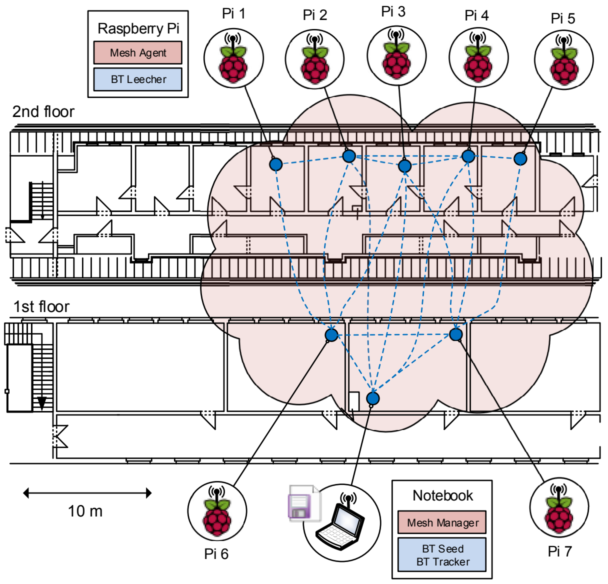

Fig. 5. Floor plan of the multi-hop mesh setup

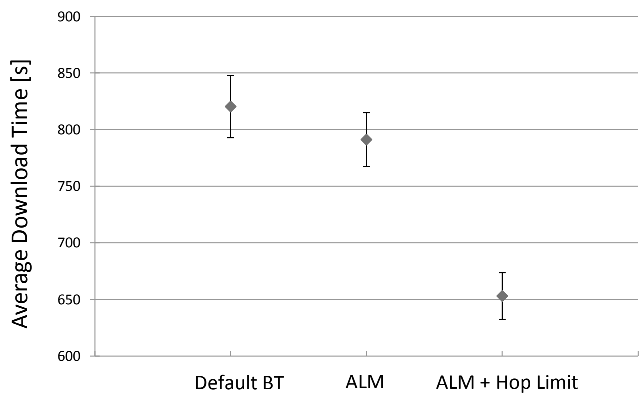

Fig. 6. Average download time in the multi-hop mesh setup

In the multi-hop setup, we could clearly observe the need for optimization of the default BT choking algorithm. As expected, average download time severely increased from 412 s (default BT mode, single-hop mesh) to an average of 820 s (default BT mode, multi-hop mesh), i.e. it nearly doubled. By applying ALM as peer sorting criterion, required time decreased again by around 4 % to an average of 791 s. Additionally limiting upload only to peers in one-hop distance, still ordered by ALM, yielded an average download time of 653 s. This means a reduction of around 20 % compared to running the default choking algorithm in this setup.

Thus, the approach of keeping BT traffic local leads to considerable improvement already for a small mesh network size of only 8 nodes and average paths lengths of 2 hops. As ALM expresses both length and overall link quality of a path, BT peers are chosen with regard the topology and quality of the mesh underlay. This reduces stress on intermediary nodes and redundant piece transmissions (see Section II-D).

Our optimization relies solely on standard features provided by 802.11s, i.e., it uses plain ALM and conducts one-hop limitation simply by comparing the list of direct neighbors with a path's target node. We expect this to perform even better in large-scale mesh setups with higher node count and longer paths. Transmission over many hops is very costly due to the required data forwarding on each intermediary node (see Section II-D) and thus peer selection has an even greater effect on BT performance.

VI. CONCLUSION

In the presented work, we use our mesh management framework as a middle-ware for exploring cross-layer optimization strategies within the specification boundaries of the 802.11s standard. We pursue a bottom-up approach that performs integration of the 802.11s default MAC-layer Airtime Link Metric (ALM) into the application layer to improve the BitTorrent (BT) P2P protocol and its choking algorithm, running on top of the physical mesh network. Defined as mandatory default metric in 802.11s, ALM is guaranteed to be available on every standard-compliant mesh node. To the best of our knowledge, the presented cross-layer solution is the first approach to integrate ALM into BT while no changes to the 802.11s routing protocol HWMP are required. As a logical starting point, we pass plain ALM to BT's choking algorithm to replace the upload rate as default peer selection criterion. Directly using ALM shows improvement already in our small mesh setup and average file download time on each node is reduced by around 4 %. Consequently, we combine ALM with a neighborhood scope limit. In our test bed, the second approach reduces average download time by up to 20 %, by still maintaining interoperability to the 802.11s standard and HWMP. The presented mesh setup represents a decisive step towards the mitigation of the mismatch between logical overlays and wireless physical underlays and hence forms the basis for establishing an even more comprehensive test bed. The trend of achievable download time reduction is clearly visible using this setup and fixed parameter set (BitTorrent defaults). In our future work, we will investigate our solution in larger and more dynamic scenarios. Both a 40 node real-world test bed and a simulation environment are currently prepared. We will evaluate the influence of varying parameters, such as choking period, number of upload slots, and piece size.

ACKNOWLEDGMENT

The authors would like to thank the German Research Foundation (DFG), Research Training Group 1424 (Multimodal Smart Appliance Ensembles for Mobile Applications) for their financial support.

REFERENCES

[1] "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 802.11-2012 (Revision of IEEE Std 802.11-2007)*, 2012.

[2] M. Portmann and A. A. Pirzada, "Wireless mesh networks for public safety and crisis management applications," *Internet Computing, IEEE*, vol. 12, no. 1, pp. 18–25, 2008.

[3] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: The WLAN mesh standard," *Wireless Commun.*, vol. 17, no. 1, pp. 104–111, Feb. 2010.

[4] S. Bari, F. Anwar, and M. Masud, "Performance study of hybrid wireless mesh protocol (HWMP) for IEEE 802.11s WLAN mesh networks," in *Computer and Communication Engineering (ICCCE), 2012 International Conference on*. IEEE, 2012, pp. 712–716.

[5] J. Eberspächer and R. Schollmeier, "First and second generation of peer-to-peer systems," in *P2P Systems and Applications*. Springer, 2005.

[6] B. Biskupski, J. Dowling, and J. Sacha, "Properties and mechanisms of self-organizing MANET and P2P systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 2, no. 1, p. 1, 2007.

[7] M. Bisignano, G. Di Modica *et al.*, "P2P over MANET: a comparison of cross-layer approaches," in *Database and Expert Systems Applications, 2007. 18th International Workshop on*. IEEE, 2007.

[8] "BitTorrent." [Online]. Available: https://wiki.theory.org/BitTorrentSpecification

[9] M. K. Sbai, C. Barakat, J. Choi, A. Al Hamra, and T. Turletti, "Adapting BitTorrent to wireless ad hoc networks," in *Ad-hoc, Mobile and Wireless Networks*. Springer, 2008, pp. 189–203.

[10] R. C. Carrano, L. C. S. Magalhães, D. C. M. Saade, and C. V. N. Albuquerque, "IEEE 802.11s multihop MAC: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 52–67, 2011.

[11] S. Kim, O. Lee, S. Choi, and S.-J. Lee, "Comparative analysis of link quality metrics and routing protocols for optimal route construction in wireless mesh networks," *Ad Hoc Netw.*, vol. 9, no. 7, Sep. 2011.

[12] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector routing," RFC 3561, Internet Engineering Task Force, Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[13] M. Guesmia, M. Guezouri, and N. Mbarek, "Performance evaluation of the HWMP proactive tree mode for IEEE 802.11s based wireless mesh networks," in *Communications and Networking (ComNet), 2012 Third International Conference on*. IEEE, 2012, pp. 1–7.

[14] R. G. Garroppo, S. Giordano, and L. Tavanti, "A joint experimental and simulation study of the IEEE 802.11s HWMP protocol and airtime link metric," *International Journal of Communication Systems*, vol. 25, no. 2, pp. 92–110, 2012.

[15] K. S. Nagegowda, H. R. Ranganath, C. Puttamadappa, and T. G. Basavaraju, "Performance evaluation of scalable routing protocols using routing metrics for wireless mesh networks under different network scenarios," *IRACST - International Journal of Computer Networks and Wireless Communications (IJCNWC)*, vol. 4, no. 6, December 2014.

[16] S. Islam, M. M. Alam, M. A. Hamid, and C. S. Hong, "High throughput path selection for IEEE 802.11s based wireless mesh networks," in *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*. ACM, 2010.

[17] "open80211s." [Online]. Available: http://open80211s.org/open80211s/

[18] "Linux wireless." [Online]. Available: http://wireless.kernel.org/

[19] "Minstrel RCA." [Online]. Available: http://wireless.kernel.org/en/developers/Documentation/mac80211/RateControl/minstrel

[20] R. G. Garroppo, S. Giordano, and L. Tavanti, "Experimental evaluation of two open source solutions for wireless mesh routing at layer two," in *Proceedings of the 5th IEEE international conference on Wireless pervasive computing*, ser. ISWPC'10. IEEE Press, 2010.

[21] Sandvine Intelligent Broadband Networks, "Global internet phenomena report 1H 2014," Tech. Rep., May 2014.

[22] B. Cohen, "Incentives build robustness in BitTorrent." First Workshop on the Economics of Peer-to-Peer Systems, June 2003.

[23] F. P. D'Elia, G. Di Stasi, S. Avallone, and R. Canonico, "BitTorrent traffic optimization in wireless mesh networks with ALTO service," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*. IEEE, 2011, pp. 1–6.

[24] A. Krifa, M. K. Sbai, C. Barakat, and T. Turletti, "BitHoc: A content sharing application for wireless ad hoc networks," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–3.

[25] A. Marques, F. Mira da Silva, and R. Rocha, "P2P over mobile ad-hoc networks," in *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops' 09. 6th Annual IEEE Communications Society Conference on*. IEEE, 2009, pp. 1–3.

[26] M. K. Sbai, E. Salhi, and C. Barakat, "A membership management protocol for mobile P2P networks," in *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*. ACM, 2009.

[27] M. Rethfeldt, P. Danielis, G. Moritz, B. Konieczek, and D. Timmermann, "Design and development of a management solution for wireless mesh networks based on IEEE 802.11s," in *Integrated Network and Service Management (IM), 14th IFIP/IEEE Symposium on*. IFIP/IEEE, 2015.

[28] "Vuze BitTorrent client." [Online]. Available: http://www.vuze.com

[29] "BASS." [Online]. Available: https://code.google.com/p/bass-plugin/

[30] "Raspberry Pi." [Online]. Available: http://www.raspberrypi.org/

[31] "rt2x00 project." [Online]. Available: http://rt2x00.serialmonkey.com

[32] "Linux wireless - list of drivers." [Online]. Available: http://wireless.kernel.org/en/users/Drivers

[33] "Wireshark." [Online]. Available: http://www.wireshark.org/