Peer my Proxy – A Performance Study of Peering Extensions for Multicast in Proxy Mobile IP Domains

Thomas C. Schmidt, Sebastian Wölke
Hamburg University of Applied Sciences
iNET Research Group, Department Informatik
Berliner Tor 7, 20099 Hamburg, Germany
t.schmidt@ieee.org, sebastian.woelke@haw-hamburg.de

Matthias Wählisch
Freie Universität Berlin
CST Research Group, Institut für Informatik
Takustr. 9, 14195 Berlin, Germany
waehlisch@ieee.org

Abstract—**Proxy Mobile IPv6 (PMIPv6) and its multicast extensions have been designed by the IETF as a deployment friendly mobility scheme. Although easy to implement, the basic multicast proxy solution suffers from unwanted delay and jitter due to suboptimal traffic flows. In this paper, we recap recent IETF work on peering extensions for multicast proxies and make the following two contributions. First we introduce the design and implementation of a highly flexible, open proxy that allows for dynamic reconfiguration at runtime. In particular, the system can support a variety of functional extensions including peering. Second we report on extensive performance measurements of proxy peering in LTE and UMTS type networks. Our findings indicate that a transparent deployment of the peering option significantly smoothes handovers and chokes delay variations throughout the access network.**

***Keywords*—Seamless mobility management; handover measurement; mobile multicast; wireless multimedia networking**

I. INTRODUCTION

Prevalent infotainment offers like IPTV and other multimedia applications require scalable and efficient network services throughout the mobile realm. IP layer multicast [1] has been designed to support multimedia group communication in a highly scalable fashion. It is widely implemented for fixed networks. In the wireless world, neither typical flash crowds of mobile users, nor the dense dissemination of infotainment channels can be supported at reasonable cost without multicast. Still the IETF has only recently released work on standard protocols for multicast mobility.

Proxy Mobile IPv6 (PMIPv6) [2] refines Mobile IPv6 (MIPv6) [3] by network-based management functions that enable IP mobility for a host without its active participation in any mobility-related signaling. Additional network entities called the Local Mobility Anchor (LMA) and Mobile Access Gateways (MAGs) are responsible for managing IP mobility on behalf of the mobile node (MN). PMIPv6 can be extended to support multicast services by deploying MLD proxy instances [4] at MAGs [5].

Even though easy to deploy and simple, the multicast basic solution for listeners leaves challenges open, as it may exhibit suboptimal routes and can lead to unwanted delay and jitter throughout the wireless access network. Handovers in particular may change the service quality at the receiver and thereby alienate users. Conversely, mobile multicast sources can cause a major service degradation for thousands of listeners, when changing from an efficient topological attachment to a disadvantageous network position. It is the objective of the present paper to present and analyze a deployment-ready solution that prevents service instabilities and enables an optimized traffic scheme.

In this paper, we take up the peering proposal for MLD Proxies presented in [6] and introduce Mcproxy[1], a fully compliant open source implementation of such an extended MLD Proxy, which has deployment in industry production systems. We show that this software has near-unicast forwarding performance and use it to experimentally validate the service gain of our peering scheme. Our findings indicate that a direct peering among multicast proxies in the access network will eliminate service fluctuations and unwanted traffic flows throughout a PMIPv6 domain. Operators who deploy PMIP multicast with peering proxies are thus enabled to offer homogeneously fast and stable multimedia services to their customers.

The remainder of this paper is structured as follows. We summarize the problem of multicast in PMIP and discuss related work in Section II. Architectural and implementation concepts of the Mcproxy solution and its evaluation are introduced in Section III. Section IV presents the performance measurements of proxy peering and discusses the results. We conclude with an outlook on future work in Section V.

II. THE PROBLEM OF MULTICAST IN PMIPV6 AND RELATED WORK

Multicast mobility has been subject to intensive studies and research proposals since the early days of the mobile Internet design [7], [8]. Various solutions have been proposed which were either built on top of multicast routers, or conjointly managed by mobility anchors. Over the years, not any one of these dedicated multicast mobility management schemes was adopted in standardization work nor in practical deployment — except for the elementary approach of bidirectional tunneling of multicast traffic via the Home Agent [9]. Reasons for this disappointment must be considered twofold, first in the hesitant

---

[1] See http://mcproxy.realmv6.org.

adoption of MIPv6, which is an end-node-centric operator-agnostic protocol. The second reason for ignoring the protocol work for mobile multicast is motivated by proposals of unrealistic complexity. Only the simple exception of bidirectional tunneling survived.

The major motivation for developing PMIPv6 was driven by an operator-friendly deployment. PMIP mobility management is centered around the anchors LMA and MAG, while mobile nodes are liberated from any protocol involvement. These anchors expose an exceptional access topology towards the static Internet for a mobile node: the MAG introduces a routing hop in situations, where the LMA architecturally acts as the next hop (or designated) router for the mobile. In the particular case of multicast communication, group membership management, as signaled by the Multicast Listener Discovery (MLD) protocol [10], requires explicit treatment on the network side. Such multicast awareness in the access can be most simply achieved by deploying MLD Proxy instances at a MAG, one for each tunnel uplink to an LMA (see Fig. 3). Multicast listener reports from a mobile thus arrive immediately at a MAG proxy instance, and aggregated membership reports can be forwarded up the tunnel to a multicast capable LMA. Likewise, the more delicate problem of supporting sender mobility–even for source-specific multicast [11]–is solved by the generic proxy behaviour of forwarding all data of locally attached sources to its upstream tunnel that reaches the LMA. The simplistic elegance of this approach decouples multicast from mobility-related signaling, while all multicast routing operations (e.g., PIM [12]) remain bound to the static LMA and unaffected by handovers.

The deployment simplicity of this base solution comes at the price of possible performance flaws. Isolated proxy instances at MAGs can only interchange traffic via its upstreams towards the LMA. Consequently, for a mobile receiver and a source that use different LMAs, the traffic has to go up to one LMA, cross over to the other LMA, and then traverse via another tunnel back to the same MAG, causing redundant flows in the access network and at the MAG. More severely, traffic delays and jitter experienced by a mobiles listener or induced by a mobile source strongly depend on the topological point of attachment within the access network and may change significantly in the event of a handover.

Several approaches have been presented to optimize performance. Jeon et al. [13] propose a designated PIM router in the access network, which requires an (untypical) flat access topology. The ROPT [14] approach extends this idea to a designated multicast tunnel head (M-LMA) that delivers multicast streams to all MAGs of a PMIP domain simultaneously. Fast handover negotiations between MAGs are proposed in [15], [16]. These protocols build on a rapid multicast context transfer between access routers, and [16] is in a final state of standardization. Multicast fast handover, though, requires unicast fast handover protocols in place, and thus does not allow for universal deployment. Context transfer via the LMA is the core idea of the RAMS protocol [17]. RAMS extends PMIP signaling by multicast state records that are centrally held at the LMA, thereby introducing scalability issues for the sake of limited performance gains. Improvements of multicast source mobility are in the focus of Nguyen and Bonnet [18]. An improvement of handover performance is achieved by transferring forwarding contexts between MAGs. In contrast, Wang et al. [19] propose a proxy extension for multiple upstreams that distribute multicast traffic according to dynamic filter rules. In the remainder of this work, we will concentrate on the peering extensions for MLD proxies as defined in the pre-final IETF standards document [6].

A peering interface at MLD proxies can be seen as a preferred data exchange for locally attached multicast sources. Peerings can be deployed between MAG-local proxy instances, as well as between remote systems interconnected by tunnels. An MLD proxy in the perspective of a sender will see peering interfaces as restricted downstream interfaces, while they appear as preferred upstream links at the listener side. No additional signaling is needed to keep peering instances in sync. Data exchange is done on a direct path, but restricted to local sources, only, to prevent multicast forwarding loops. In this way, the MLD peering approach keeps the simplicity and transparency (w.r.t. receivers and senders) of the base solution, but provides a significant performance optimization as presented in Section IV.

III. MCPROXY

Mcproxy is an open source implementation of a multicast proxy daemon for Linux. It is compliant with *IGMP/MLD Proxying* [4] and supports the group membership protocols *IGMPv3* [20] and *MLDv2* [10] for IPv4 and IPv6 respectively. Mobility of PMIPv6 multicast listeners is provided by standard extensions [5]. To the best of our knowledge, this is the first implementation which fulfills all requirements to deploy multicast mobility in PMIPv6 domains.

Our implementation allows for multiple proxy instances in parallel on a single machine. Each instance interacts with a different routing table of the Linux kernel to activate isolated fast forwarding. Furthermore the proxy introduces a dynamically configurable data filter at kernel level to handle *multiple upstream interfaces* and *peering interfaces*. This is required by current *PMIPv6 source mobility* approaches [6].

All source code of the Mcproxy is released under GNU GPLv2 and available via GitHub https://github.com/mcproxy/mcproxy.

A. Architecture

The software architecture of Mcproxy is modular (cf., Figure 1). The most abstract module *Mcproxy* covers the proxy and acts as a management module. This management module loads and parses a given configuration file with the help of the *Configuration* module. It contains all necessary settings such as the IP version, the setup of each single proxy instance with their up- and downstream interfaces, and all desired filter rules.

After basic initial configuration, the proxy loads the required *Proxy Instances* and passes on their specific configuration. Each proxy instance creates and maintains an IGMP/MLD raw socket with a unique multicast routing table flag (`mrt` flag) to interact with the Linux kernel. Then the proxy instance starts a multicast *Querier* for every downstream interface and one *Routing* module. A *Querier* is responsible for the group memberships of its assigned subnet and the *Routing* module processes routing events (e.g., new multicast sources)

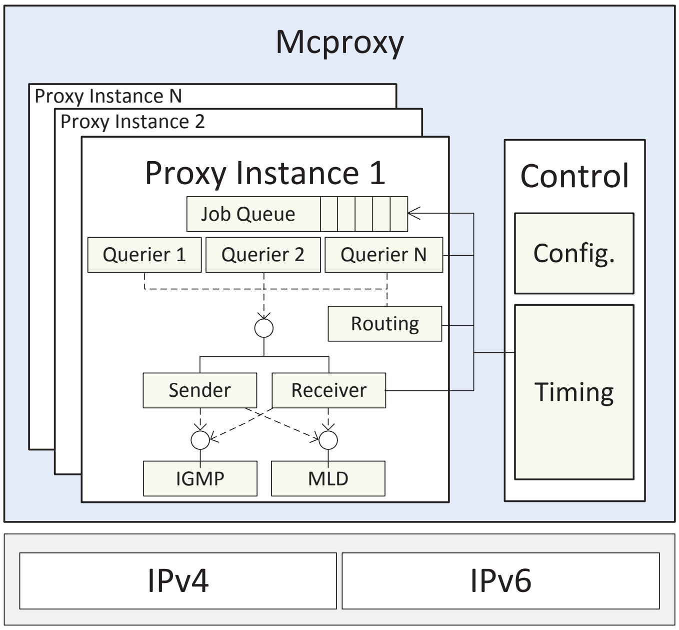


Figure 1. Architecture of Mcproxy

and updates the routing table with respect to the configured filter rules.

Abstracting the general functions of the multicast proxy from specific IP versions is very important to decrease code redundancy and ease code maintenance. Up to this hierarchy level, all modules are independent of the IP version, as all processed IP addresses are embedded in a transparent data structure. The IP version-dependent source code is hidden in the modules *Sender* and *Receiver* which cover socket calls, create group membership messages, receive and process them. Finally, the module *Timing* is responsible for all aspects of time-dependent behavior.

Proxy instances are decoupled and isolated in different threads. For interaction they provide a message passing service that keeps their internal states thread-safe. This service is used for example by the *Receiver* to notify about arriving join and leave messages, and by the module *Timing* to notify about expired timers.

B. Multicast Data and Signalling Filter

Mcproxy provides a dynamically configurable multicast filter at kernel level, which controls multicast group management signalling and the forwarding of application data. In detail, the filter maintenances group membership at the querier side, the group membership aggregation to the upstreams, and the multicast data flow. This filter is fundamental for two operations: (a) protecting SSM against ASM receivers, (b) handling of multiple upstreams and peering interfaces. The filter follows a flexible design. It is possible to configure a black or whitelist filter, as well as an input and output filter for specific up- or downstream interfaces.

a) Control multicast subscriptions: The filter function enables parallel operation of SSM and ASM. Typically, there are two scenarios, which might lead to conflicts:

S1 SSM as well as ASM multicast listeners join the same multicast group. In this case the multicast proxy needs to ensure that the ASM subscription is signalled to the upstream node, otherwise the SSM listeners would limit the multicast data for ASM peers to specific sources.

S2 ASM listeners are restricted to specific sources per proxy domain. Such a scenario is useful in mobile environments, where operators want to control data capacity by restricting data sources. For example, a downstream interface is restricted to the use of the SSM multicast channel (S, G). If this interface receives the ASM membership report $EXCLUDE(\{\}, G)$, i.e., a multicast listener is interested in all sources of the group G, then the ASM membership subscription can be converted to the state $INCLUDE(S, G)$.

Using Mcproxy, an administrator can adapt the configuration to the actual deployment scenario.

b) Handling multiple upstreams and peering interfaces: The support of this function is beneficial to implement load balancing, fallback mechanisms, or simultaneous interaction with disjoint multicast networks. The exact behaviour of a multicast proxy with multiple upstreams depends on the use case and must be treated separately by the direction of the data flow.

S1 A data flow from a downstream node is distributed to selected upstreams. Based on configuration, an upstream subscription is emulated, i.e., the multicast proxy creates group states for upstream interface. When data arrives from the downstream interface, the Mcproxy matches the data flow and forwards the data only to the corresponding upstream peers. Alternatively, priorities are assigned to upstream peers, and data is only sent to the node with the highest priority.

S2 The Proxy receives the same data from multiple upstreams. To avoid duplicate data transmission to multicast listeners, the multicast proxy needs to restrict downstream transmission. There are two options for this. The multicast proxy joins the requested multicast channel only at the upstream interface with the highest priority. Alternatively, the proxy joins the multicast channel at all upstream interfaces with matching filter rules, but at the moment of arriving data it unsubscribes on all interfaces except the incoming data interface. The latter approach thus automatically adapts to the delay of the upstream peers.

Peering interfaces defined by the *PMIPv6 source draft* [6] are considered as restricted down- and upstream interfaces. This means a multicast proxy with a peering interface reflects a multiple upstream proxy. It only has to be configured with the appropriate upstream behavior and filter rules.

C. Evaluation of Routing Performance

We new examining the forwarding behaviour of the Mcproxy. Our goal is to provide insight on the routing performance and the scalability related to multiple downstreams.

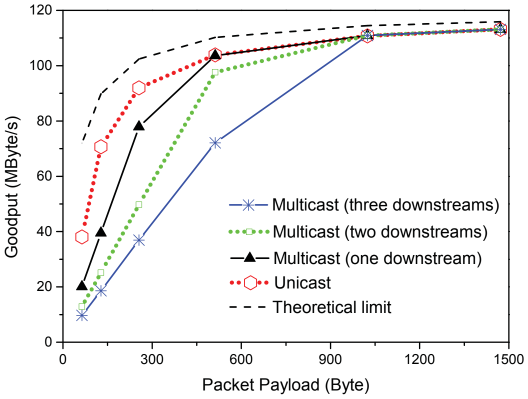


Figure 2. Comparison of unicast and multicast goodput

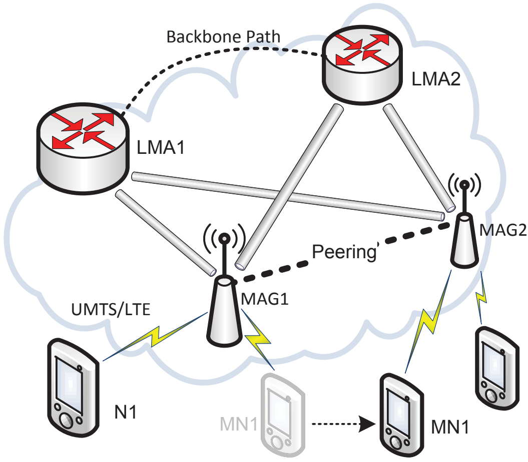


Figure 3. Setup of the experimental testbed

In detail, we compare the goodput to the unicast routing performance of the Linux kernel. Our testbed consists of one source and multiple receivers, each equipped with a Gigabit Ethernet network card directly connected to the test router (AMD Opteron 6376 Processor with 2.3 GHz clock frequency, Linux kernel version 3.11.10). The sender attempts to transmit packets of variable size at full network speed.

Figure 2 shows the unicast and multicast goodput as a function of payload sizes. We determine a maximum of 113.1 Mbyte/s in all cases, which approaches the network capacity. For smaller payload sizes, forwarding of packets exceeds the processing capacity of our test router, which leads to packet drops. This reduces the data rate for unicast and multicast. Multicast capacity is further reduced by the Linux kernel. This behaviour could be due to different implementations of multicast and unicast routing tables in the Linux kernel. For multiple receivers, the router has to copy packets prior to forwarding, which further degrades the overall packet throughput as visible in Figure 2.

IV. PEERING PERFORMANCE ANALYSIS

A. Experimental Setup and Measurements

Our evaluation of PMIPv6 multicast is driven by two aspects, (a) an analysis under realistic conditions but (b) in a clean measurement environment. Consequently, we deploy Mcproxy in a network setup that allows for concentrating on multicast-specific mechanisms without side effects.

1) Basic Network Setup: The network topology is built by Mininet[2], a network emulation tool. We take three advantages of this. First, it incorporates real Linux multicast routing tables into the emulated network. We thus can easily deploy Mcproxy in a complex testbed. Second, it controls link delays to evaluate different types of access networks. Third, we can use a single clock across all nodes to accurately measure one-way delays.

Figure 3 depicts the basic network setup. It covers all basic variations of topological deployment for group communication in PMIP environments by using fixed Nodes (N), Mobile Nodes (MN), Local Mobility Anchors (LMA), and Mobile Access Gateways (MAG). Each node (M)Ni is assigned to an LMAi and MAGi respectively. Each MAG runs one multicast proxy instance for every attached LMA. An LMA represents the upstream router. The downstream interfaces of the multicast proxy instances are connected to the MNs of the corresponding LMAs. Furthermore all proxy instances of the MAGs maintain peering interfaces between each other. The end devices are connected either via an emulated UMTS or via an LTE link to the MAGs. The link delays between the entities vary and are summarized in Table I.

The multicast source sends a constant bit stream of 100 packets per second. We focus on a single group scenario, as the number of groups does not affect the performance of the multicast scheme.

2) Mobility and Mobile Operator Scenarios: The experiments consider sender and receiver mobility, as well as single and multi LMA scenarios.

Sender mobility: The multicast source moves from MAG1 to MAG2, and the multicast receiver is connected to MAG2.
Receiver mobility: The multicast receiver moves from MAG1 to MAG2, and the multicast source sends data via MAG1.

A mobile node moves between the access networks after 500 ms. We do not account for specific layer 2 handover delays, because these delays are independent of the multicast scheme in use.

Single LMA scenario: MAGs and (M)Ns are assigned to a single LMA. Peering mechanisms thus will not be applied.
Multi LMA scenario: Two LMAs operate as upstreams for the mobile access gateways. A backbone network interconnects the LMAs. Data delivery can be bypassed via the peering link between both MAGs.

3) Performance Measurements: In the subsequent sections, we evaluate the PMIPv6 multicast scheme based on the one-way packet delivery delay and jitter between source and receiver(s). The source adds timestamps to the multicast data

---

[2] See http://mininet.org.

Table I. LINK DELAYS AND VARIATIONS IN THE TESTBED

Links	Delay (ms)	± Jitter (ms)
UMTS ((M)N ⟷ MAG)	200	± 50
LTE ((M)N ⟷ MAG)	5	± 3
Tunnel (MAG ⟷ LMA)	20–40	± 6
Peering (MAG ⟷ MAG)	5–15	± 3
Backbone Path (LMA ⟷ LMA)	40–60	± 12

packets, which will be evaluated by the receivers. We will analyze the delay in different granularity:

Delay space: The delay space measures the relative frequency of packet travel time between the multicast source and two receivers, each receiver is connected to a different MAG. It is worth noting that no mobile node is involved. All nodes are fixed to analyze the basic delays.

Handover performance: The handover performance is evaluated based on the continued transmission delay as well as the corresponding interarrival jitter. We present measurements between source and receiver per packet.

We conduct ten experiments for each scenario to account for statistical variations. We verified that the results are converged. Each experiment lasts for at least one second.

B. Results

In the following, we present selected results of our extensive measurements. We focus on the core service parameters *delay* and *interarrival jitter*. *Packet loss* only occurs from handover management and will be discussed below with handover traces. First we examine the simple deployment case of a single LMA. Second we take a closer look at the more intricate case of senders and receivers associated with different LMAs.

1) Single LMA Scenario: Figures 4 and 5 visualize the delay space distribution throughout provider networks for LTE and UMTS access radio respectively. The empirical distributions represent the relative frequencies of delay values that occur in an evenly populated provider network (i.e., the mobile clients are evenly scattered among MAGs).

Without proxy peering, the delay follows a multi-modal distribution with two peaks, one from local traffic and another from remote reception via the LMA. This is clearly visible for LTE (s. Fig. 4(a)), while the slow UMTS access blurs the shapes, but leads to a similar increase in delay and delay variation (s. Fig. 5(a)). As soon as peering is activated, delay distributions reduce to the mono-modal shape that is characteristic for the latencies of network and radio access technologies (s. Figs. 4(b), 5(b))—all service degradations due to triangular routing disappear.

The results of our handover analysis are displayed in Figures 6 and 7. We restrict figures to receiver measurements, as the transmission in this simple scenario is symmetric with almost identical results for source handovers. After the mobile receiver changed from MAG1 to MAG2, transmission delays increase according to the extended data paths. Without peering, about 50 ms are added to the latency, which reduces the overall performance of the fast LTE network by a factor of five. Correspondingly, the interarrival jitter boosts up to about 80 ms causing a noticeable disturbance to real-time applications. In contrast, LTE handovers significantly improve in the peering case, smoothening the jitter below the critical mark of 50 ms. Delays remain within the performance bound of ≈ 30 ms characteristic for the access network.

Similar absolute differences are seen for the slow UMTS network, but are again blurred by the largely fluctuating performance of the radio layer. The overall interarrival jitter is enhanced accordingly. Packet loss during about 400 ms in the UMTS case is due to mobility-related MLD signaling in the access. This essentially is caused by a single packet exchange on the radio link and cannot be changed by peering. Much enhanced LTE performance reduces packet loss at handovers to about three packts or 30 ms.

In summary for the simple deployment scenario, a significant improvement of the QoS for mobile multimedia services became apparent from peering between the proxies. This effect becomes dominant for LTE networks, which reach compliance with conversational services by peering. The lower performance of UMTS radio links dominates performance and keeps service improvements by peering less pronounced.

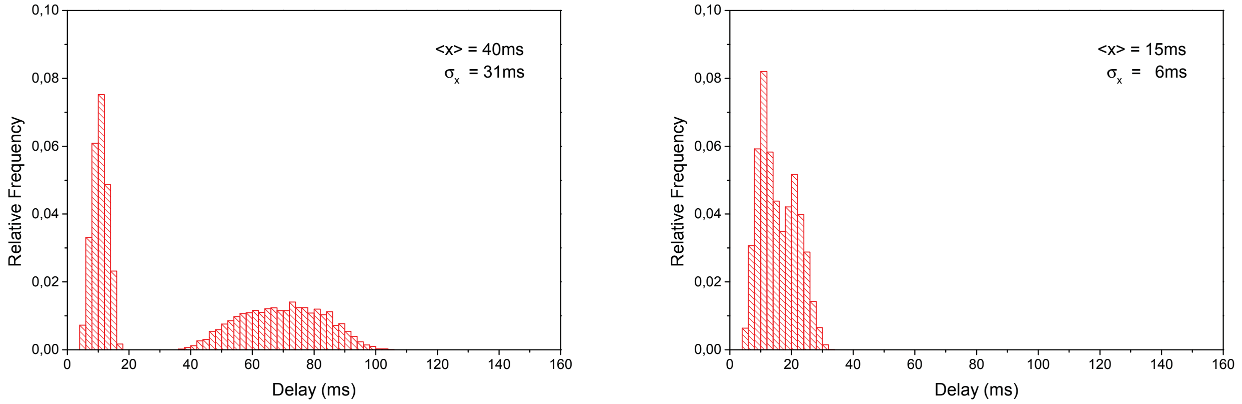
2) Multiple LMA Scenario: The multi LMA scenario adds the complexity of inter-LMA traffic handling and must be seen as the more intricate, but also more realistic use case. For space constraints, we restrict the presentation to results for the LTE case, which shows clearer visibility of the peering effects.

The delay space distribution with and without peering is visualized in Fig. 8. In the non-peering case (Fig. 8(a)), two modes (LMA1 and LMA2) of the distribution overlap. It is noteworthy that our setting omitted the trivial case of local delivery, which would have added a third mode of delays centered around 10 ms. The peering case (Fig. 8(b)) reduces complexity and modality of the distribution to the results known from the single LMA scenario (Fig. 4(b)).

Sender handover in the multi LMA scenario exhibits several topological dimensions. In Fig. 9, we display the expressive case of a source that moves from the receiver-local MAG1 to a remote MAG2. Without peering, the initially minimal delay jumps up drastically under handover to values of triangular paths with a collateral jitter explosion. With peering, the delay after handover is only slightly enhanced by the inter-MAG communication path with the result of smooth QoS values under handover. As receivers are permanently subscribed and no additional signaling is involved in the handover, almost no packet loss occurs.

The receiver handover in this multi LMA scenario admits a different behaviour as visualized in Fig. 10. The source associated with LMA1 sends its data upstream in the absence of peering. Prior to handover, the receiver can only obtain traffic from its associated LMA2, why the delay remains above 100 ms, even though the nodes are topologically close. As a consequence, jitter and packet loss (≈ 60 ms) are significantly pronounced and spoil the handover performance. Peering reduces the overall delay drastically and guaranties an overall handover performance very similar to the simple scenario with only one LMA (Fig. 6(b)).

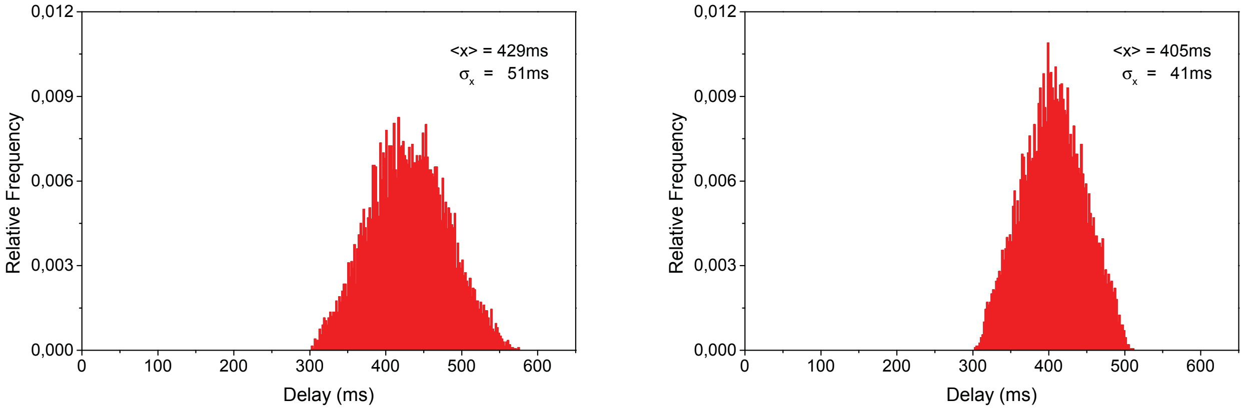
Overall, the introduction of peering between MLD proxies could clearly demonstrate its effectiveness on the path towards a smooth multicast service layer in the mobile Internet.



(a) without peering

(b) with peering

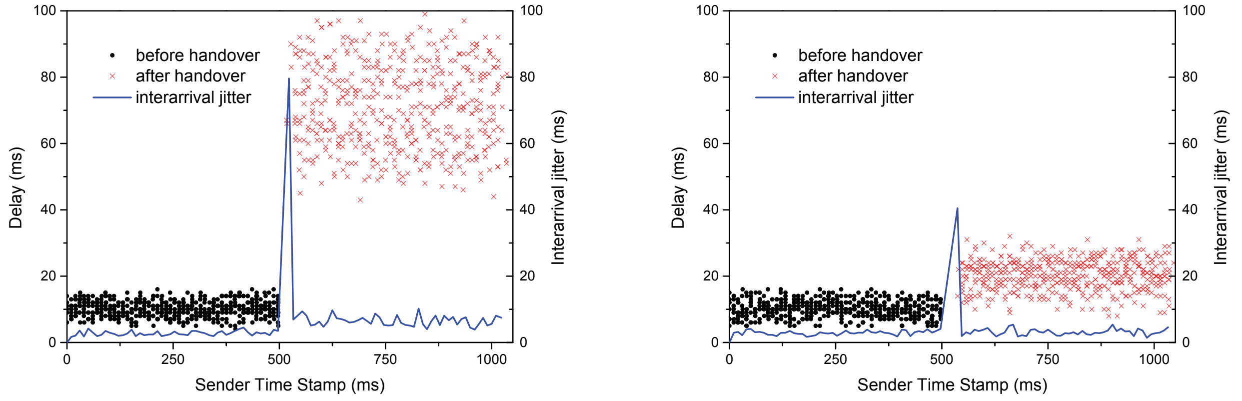
Figure 4. Delay space distribution in a single LMA scenario (LTE)



(a) without peering

(b) with peering

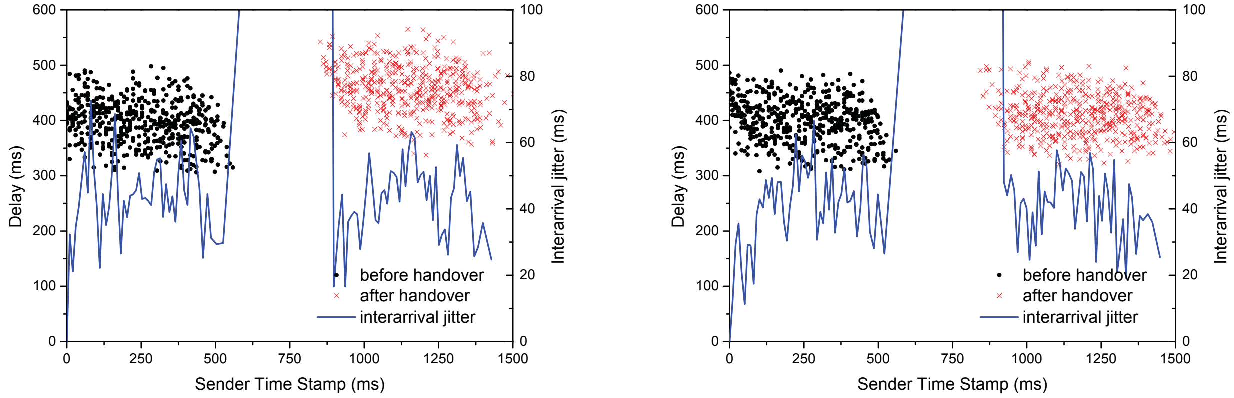
Figure 5. Delay space distribution in a single LMA szenario (UMTS)



(a) without peering

(b) with peering

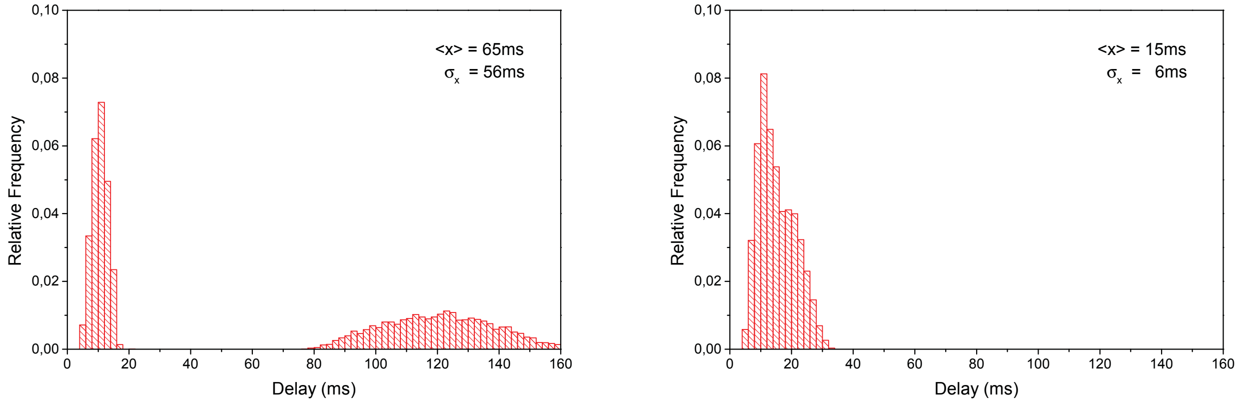
Figure 6. Receiver handover in a sinlge LMA szenario (LTE)



(a) without peering

(b) with peering

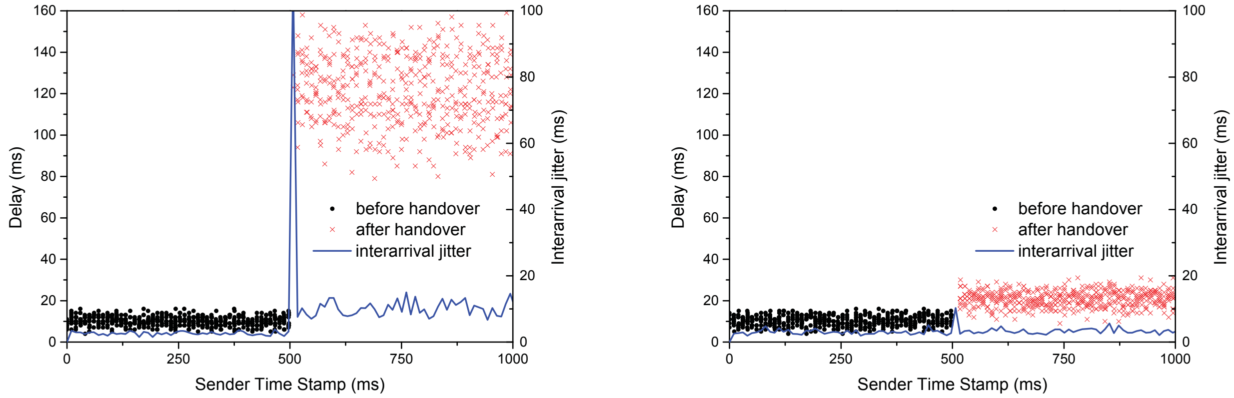
Figure 7. Receiver handover in a sinlge LMA szenario (UMTS)



(a) without peering

(b) with peering

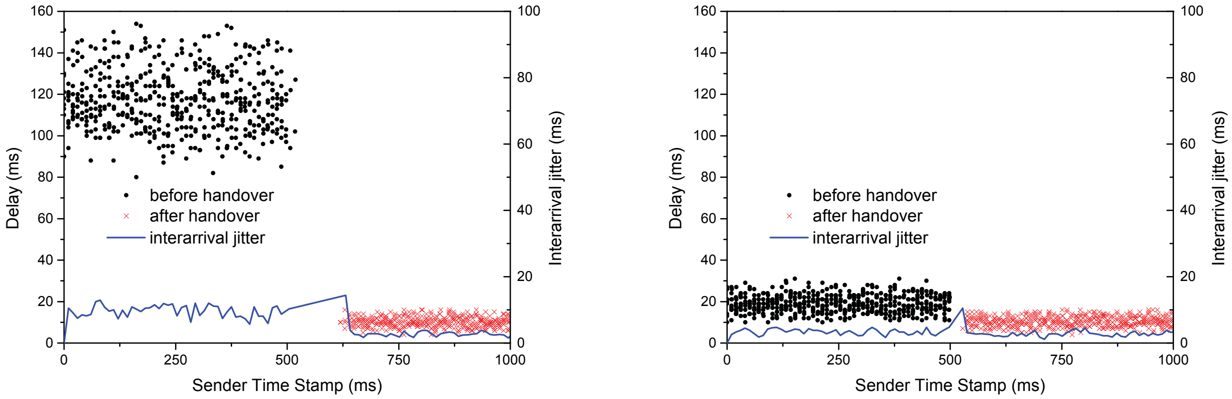
Figure 8. Delay space distribution in a multi LMA szenario (LTE)



(a) without peering

(b) with peering

Figure 9. Sender handover in a multi LMA szenario (LTE)



(a) without peering

(b) with peering

Figure 10. Receiver handover in a multi LMA szenario (LTE)

V. CONCLUSIONS AND OUTLOOK

In this work, we have taken up current IETF standardization efforts on multicast mobility management in Proxy Mobile IPv6. We contributed the design along with an open source implementation of an enhanced IGMP/MLD Proxy component that is capable of supporting all related current and the major emerging standards. This software was used to experimentally analyze key performance features of the peering approach between multicast proxies. Our measurement results could clearly demonstrate that this simple and easily deployable solution does enhance the quality of service in characteristic mobility use cases, while no additional signaling nor overhead was introduced.

In future work, we will extend our approach and implementation to include general fast handover operations [16] into the proxy peering. This will lead to the fastest handover management for multimedia multicast services that is achievable on the network layer.

REFERENCES

[1] S. E. Deering and D. R. Cheriton, “Multicast Routing in Datagram Internetworks and Extended LANs,” *ACM Trans. Comput. Syst.*, vol. 8, no. 2, pp. 85–110, 1990.

[2] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy Mobile IPv6,” IETF, RFC 5213, August 2008.

[3] C. Perkins, D. Johnson, and J. Arkko, “Mobility Support in IPv6,” IETF, RFC 6275, July 2011.

[4] B. Fenner, H. He, B. Haberman, and H. Sandick, “Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding (’IGMP/MLD Proxying’),” IETF, RFC 4605, August 2006.

[5] T. Schmidt, M. Waehlisch, and S. Krishnan, “Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains,” IETF, RFC 6224, April 2011.

[6] T. C. Schmidt, S. Gao, H.-K. Zhang, and M. Wählisch, “Mobile Multicast Sender Support in Proxy Mobile IPv6 (PMIPv6) Domains,” IETF, IETF Internet Draft – work in progress 09, March 2014. [Online]. Available: http://tools.ietf.org/html/draft-ietf-multimob-pmipv6-source

[7] I. Romdhani, M. Kellil, H.-Y. Lach, A. Bouabdallah, and H. Bettahar, “IP Mobile Multicast: Challenges and Solutions,” *IEEE Comm. Surveys & Tutorials*, vol. 6, no. 1, pp. 18–41, 2004.

[8] T. C. Schmidt, M. Wählisch, and G. Fairhurst, “Multicast Mobility in Mobile IP Version 6 (MIPv6): Problem Statement and Brief Survey,” IRTF, RFC 5757, February 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5757.txt

[9] G. Xylomenos and G. C. Polyzos, “IP Multicast for Mobile Hosts,” *IEEE Comm. Mag.*, vol. 35, no. 1, pp. 54–58, 1997.

[10] R. Vida and L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6,” IETF, RFC 3810, June 2004.

[11] T. C. Schmidt and M. Wählisch, “Extending SSM to MIPv6 — Problems, Solutions and Improvements,” *Computational Methods in Science and Technology*, vol. 11, no. 2, pp. 147–152, November 2005. [Online]. Available: http://www.man.poznan.pl/cmst/papers/11_2/CMST11-08.pdf

[12] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised),” IETF, RFC 4601, August 2006.

[13] S. Jeon, N. Kang, and Y. Kim, “Mobility management based on proxy mobile IPv6 for multicasting services in home networks,” *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 3, pp. 1227–1232, 2009.

[14] J. Zuniga, L. Contreras, C. Bernardos, S. Jeon, and Y. Kim, “Multicast Mobility Routing Optimizations for Proxy Mobile IPv6,” IETF, RFC 7028, September 2013.

[15] J.-H. Lee, T. Ernst, D.-J. Deng, and H.-C. Chao, “Improved pmipv6 handover procedure for consumer multicast traffic,” *IET communications*, vol. 5, no. 15, pp. 2149–2156, 2011.

[16] T. C. Schmidt, M. Wählisch, R. Koodli, G. Fairhurst, and D. Liu, “Multicast Listener Extensions for MIPv6 and PMIPv6 Fast Handovers,” IETF, IETF Internet Draft – work in progress 05, March 2014. [Online]. Available: http://tools.ietf.org/html/draft-ietf-multimob-fmipv6-pfmipv6-multicast

[17] L. M. Contreras, C. J. Bernardos, and I. Soto, “Rams: A protocol extension to PMIPv6 for improving handover performance of multicast traffic,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 2, pp. 67–82, 2011.

[18] T.-T. Nguyen and C. Bonnet, “Performance Optimization of Multicast Content Delivery in a Mobile Environment based on PMIPv6,” in *Proc. of WCNC 2013*. IEEE, 2013.

[19] L. Wang, S. Gao, H. Zhang, T. C. Schmidt, and J. Guan, “Mobile Multicast Source Support in PMIPv6 Networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, p. 152, 2013. [Online]. Available: http://dx.doi.org/10.1186/1687-1499-2013-152

[20] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, “Internet Group Management Protocol, Version 3,” IETF, RFC 3376, October 2002.