Large-Scale Measurement and Analysis of One-Way Delay in Hybrid Multicast Networks

Sebastian Meiling, Thomas C. Schmidt
smeiling@acm.org, t.schmidt@ieee.org
iNET Research Group – Department Informatik
Hamburg University of Applied Sciences
Berliner Tor 7, 20099 Hamburg, Germany

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

***Abstract*—Group communication plays an important role in the distribution of real-time data for IPTV, multimedia conferencing, or online multiplayer games, but IP multicast remains unsupported in today's global Internet. Hybrid solutions that bridge between overlay and underlay multicast are a promising escape from the deployment dilemma of multicast.**

In this paper, we examine the real-time capabilities of hybrid multicast in a globally distributed environment based on our adaptive architecture H∀Mcast within the Planet-Lab testbed. We present a large-scale measurement study and analysis of one-way packet delay distributions in several realistic group scenarios. The unique results in global traces of hybrid multicast data have been achieved by carefully tracking packets and continuously correcting clock offsets. Companion measurements of unicast-based distribution are part of our analysis, as well as the comparative discussion of our results with previous findings from theory and simulation. Our measurements reveal that about 50 % of global group members experience a real-time compliant service within the conversational time bounds of $150\ ms$.

***Index Terms*—Network measurement, global group communication, performance evaluation**

I. Introduction

Many popular Internet applications such as IPTV, MMORGS, online social networks, and Audio/Video conferencing rely on some form of group communication that can be efficiently provided by multicast. Multicast as a network service [1] simplifies application development and minimizes network load when implemented on the lowest possible layer. However, today's deployment of group services remains restricted to regional 'walled gardens' and layer two domains. Applications that are built to run globally are therefore forced to implement a group layer independent of network conditions.

Hybrid multicast is a promising approach to bridge the deployment gap and has been recently fostered in several characteristics. We proposed and developed a generic solution to hybrid adaptive multicast (H∀Mcast) that establishes group communication intelligence at a system level. Combined with a common API [2], applications can simply rely on a system library that dynamically selects the most beneficial communication service available at runtime. This approach enables 'write-once-run-everywhere' applications that take full advantage of network capabilities.

Hybrid network services impose a performance penalty, whenever networking is raised to the application layer. In particular, distribution delays increase on the overlay, which is critical for many real-time applications. In this paper, we present a methodology, large-scale measurements and analysis of the global performance characteristics for hybrid multicast communication in order to quantify these performance penalties in realistic settings. We chose different typical deployment scenarios, including native multicast regions on the IP and link layer, to explore the expected quality of service in comparison with a native distribution system. To the best of our knowledge, this is the first real-world measurement of one-way delay distributions for hybrid multicast on a global scale.

The remainder of the paper is organized as follows. In the next Section II we introduce the major challenges and related work on hybrid multicast approaches and network delay measurements. Section III outlines our methodologies for one-way delay measurements as well as for synchronizing packet timestamps. The experimental setup and measurement results are discussed in Section IV. We conclude in Section V and give an outlook on future work.

II. Global Group Communication: Challenges & Related Work

IP multicast [1] was proposed more than two decades ago for IPv4 and is supported in IPv6 by design. However, IP multicast suffers from inherent deployment issues [3] that led to isolated multicast islands – so called "walled gardens". Overlay Multicast (OLM) architectures such as the MBone [4] or AMT [5] aim to connect multicast edge networks by IP tunneling. An increased flexibility for end systems arose with the advent of peer-to-peer (P2P) overlay protocols, the idea of infrastructure-independent service deployment was also considered for group communication by Application Layer Multicast (ALM) protocols (e.g., CAN [6], NARADA [7], NICE [8], Bayeux [9], and Scribe [10]). Still these multicast technologies are bound to specific applications, and no common programming access nor service deployment concepts exist for the Internet.

The plurality of multicast flavors (ASM, SSM) and its technological instantiations (IP, OLM, ALM) raise the issue of connecting the various distribution systems in an interoperable way. The missing links are (a) an application programming interface to transparently use group communication and (b) a

comprehensive naming scheme to identify multicast groups in hybrid scenarios.

A. Hybrid Multicast

The key idea of hybrid multicast is to implement a global group communication service by integrating different multicast technologies. Following a dynamic selection from IP, OLM, and ALM, hybrid multicast approaches combine the efficiency of IP multicast and the easy deployment of overlays.

Universal Multicast (UM) [11] is a two-layered, hybrid multicast approach. In the upper layer, designated members interconnect multicast islands with a tree-based overlay multicast protocol. Each designated member is a special host within a native IP multicast network in the lower layer and acts as a gateway between the layers. UM further utilizes a hybrid group management protocol and global group IDs to provide group communication between all multicast islands. The authors evaluated the proposed architecture by simulation only. They compared UM with native unicast for different group sizes and metrics, i.e., cost ratio and relative delay penalty.

Hybrid Shard Tree (HST) [12] is a multi-layered architecture and routing approach to combine network- and subnetwork-layer multicast services in end-system domains with transparent, structured overlays on the inter-domain level. HST constructs shared distribution trees that remain transparent on the inter-domain level with the help of Inter-domain Multicast Gateways (IMGs). No changes to group management or IP routing are imposed. Instead, IMGs reactively adapt to the routing and group dynamic on different layers, and do not require an additional (stateful) control layer.

Scalable Hybrid Multicast (SHM) [13] introduces another two-layered, hybrid multicast approach. In the upper layer, Domain Agents are connected through a distribution tree. The central control of the tree relies on a single rendezvous point, which introduces a potential bottleneck for group management functionality. Each domain agent is responsible for a local edge network and forwards data and control information between the upper and lower layer. In contrast to UM, SHM distinguishes between IP-multicast islands and node domains in the lower layer. In a node domain, group communication is then again enabled by an ALM protocol. The performance of SHM was evaluated based on simulation. The authors analyzed delay, link stress, stretch, and control overhead compared to IP multicast and an ALM protocol but no other hybrid approach.

Island Multicast (IM) [14] uses a two-layered topology concept for a hybrid multicast architecture. In the lower layer IM uses native IP multicast if available. To connect two IP multicast islands, IM establishes tunnels. The evaluation is based on simulation regarding link stress, relative delay penalty, and control overhead, as well as Planet-Lab deployment to analyze join/leave latencies, control overhead, and relative delay penalty.

Our current work is based on H∀Mcast [15], a generic approach to hybrid adaptive multicast that extends the HST architecture. H∀Mcast enables an evolutionary system-centric service of global multicast in the Future Internet. Its concept includes a common multicast API [2] with an abstract naming scheme for multicast groups. Combined with an adaptive system middleware[1] and transparent interdomain multicast gateways (IMGs) to overcome administrative and technological borders, H∀Mcast is able to utilize heterogeneous network technologies and to facilitate a global group communication service. It flexibly inter-connects any multicast-enabled network domain. In a preliminary performance analysis [16], we showed that our prototype implementation of a hybrid multicast architecture allows for similar performance in terms of packet and data throughput compared to the Linux Kernel. This analysis focused on local environments to evaluate the ground truth of the communication stack. Further measurements and analysis with a dedicated focus on inter-domain scenarios in real-world deployments are integral part of this paper (cf., Section IV).

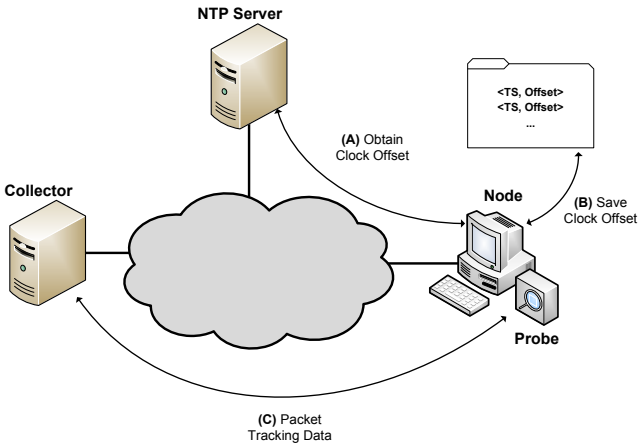


Fig. 1. Time Measurement Setup

B. Measurement and Evaluation

The evaluation of networking protocols and communication schemes benefits from common metrics to allow for comparison. In the context of hybrid multicast, mainly end-to-end delay, delay penalty, routing stretch, and link stress have been applied. All of these measurements require a unidirectional fine-grained analysis per link, which is challenging and usually approximated by the round trip time (RTT). Claffy et al. [17] discuss methodologies for one-way delay measurement in unicast communication. They argue that RTT measurements are insufficient and often misleading to determine unidirectional latencies. Lao et al. [18] present a comparative study of various multicast schemes, which have been implemented on different layers (IP, OLM, ALM). They evaluated end-to-end delay, multicast tree cost, and control overhead. However, this was limited to simulation. Castro et al. [19] concentrated on the evaluation of P2P multicast protocols. They simulated and analyzed CAN-style vs. Pastry-style and flooding vs. tree-based multicast schemes with respect to the relative maximum

[1] A public release of the H∀Mcast software can be downloaded at http://hamcast.realmv6.org/developers.

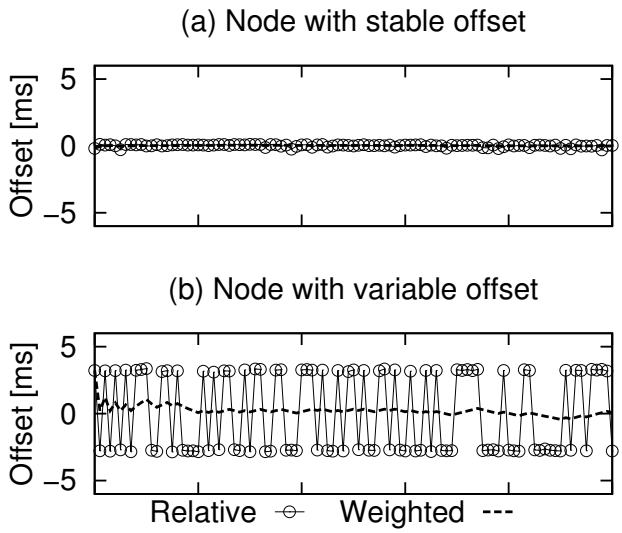


Fig. 2. Sample measurements of the deviation from mean offsets.

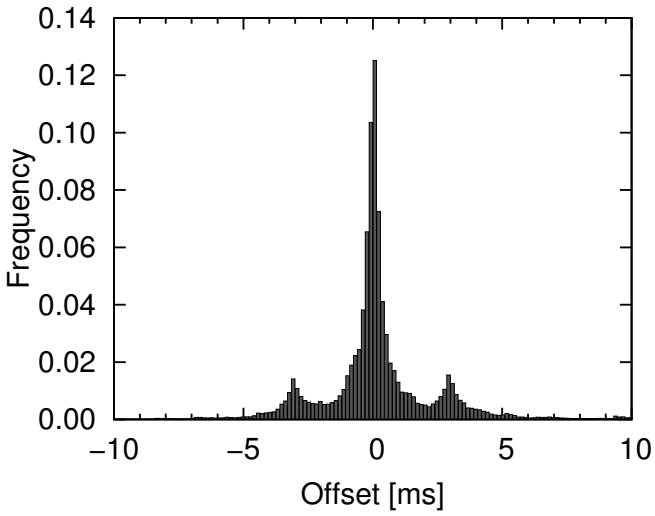


Fig. 3. Temporal errors: Deviation from mean offset across all nodes.

and average delay penalty. Wählisch and Schmidt [20] derived an á priori estimator for delay distributions in hybrid multicast schemes. Based on an analytical model, they compared IP multicast, hybrid multicast with Scribe and hybrid multicast with CAN. Confirming earlier simulation results of Castro et al., they found that Scribe-based hybrid multicast largely outperforms CAN-based services. The use of efficient overlay schemes can further enable a near to optimal (IP multicast) delay when applied to interconnecting multicast networks on an upper Internet tier.

To the best of our knowledge, there is no large-scale hybrid multicast experimental study available that analyzes the effects in real-world settings. Still it is highly important to quantify the performance of hybrid multicast thoroughly (i.e., considering the one-way delay) as hybrid architectures are a promising direction for global group communication in the common Internet.

III. Measurement Methodologies

A. One-Way Packet Delays in a Global Distribution System

Multicast decouples senders from receivers in a connectionless, one-way communication without feedback channel. Thus, delays can only be observed at intermediate forwarders along the path and at receiving endpoints. To calculate the *one-way delay* (OWD) in a globally distributed system like the Internet, it is necessary to identify and uniquely track data packets on their way from the source to *all* destinations. This requires a measurement of selected probe packets using distributed clocks, i.e.,

1) sender timestamp of a packet obtained from the source
2) receiver timestamp of the same packet at all endpoints.

While common round trip time (RTT) measurements can be calculated from a single clock, OWD measurements rely on the accuracy and synchronization of the internal clocks of at least two nodes. This synchronization on a fine-grained base is challenging and requires realistic error estimates.

In the following subsections, we outline the basic methods used to track packets in a hybrid multicast network environment and to approach a synchronous time of controlled errors for a large number of nodes.

B. Packet Tracking

To trace selected multicast packets along paths in a hybrid multicast network from source to all receivers, we used a modified version of the packet tracking framework developed by the Fraunhofer FOKUS group [21]. The framework reduces measurement traffic and synchronizes the sampling fractions. It consists of three components:

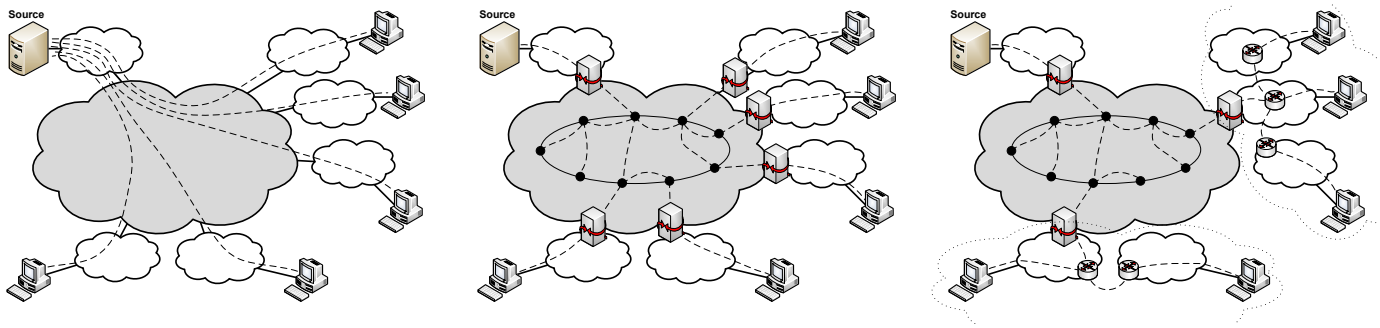
- a measurement probe deployed on each tracking node,
- a collector matching data from all probes,
- and a viewer to visualize per packet paths.

The measurement probe passively captures packets at a node and records timestamps of send and receive operations. The probe periodically reports packet traces to the collector that preprocesses data from all probes and writes trace files. To allow for multiple receivers connected via different distribution technologies, we extended the framework by a new signature scheme. For our evaluation, we further processed the traces from the collector to extract the relevant multicast data packets.

C. Clock Synchronization & Correction

Synchronized internal clocks at all nodes as well as temporal error estimates are mandatory for our one-way packet delay measurements. Our large-scale measurements have been deployed on the Planet-Lab testbed environment. All Planet-Lab (PL) nodes run NTP [22] to synchronize system clocks, but clock offsets up to a magnitude of seconds would still influence our OWD measurements negatively.

We found that 89 % of the selected Planet-Lab nodes synchronize with a stratum 1 NTP server and have an average



(a) N-times Unicast (b) Hybrid Multicast (switched) (c) Hybrid Multicast (routed)

Fig. 4. Overview on the three measurement scenarios deployed on Planet-Lab.

absolute clock offset of $1,344$ s (MIN = 0.011 s, MAX = 25.475 s). Another 8 % of the nodes use a stratum 2 server and have an average offset of $0,471$ s (MIN = 0.014 s, MAX = 2.312 s). The remaining 3 % of all selected nodes do not synchronize their internal clocks at all. Moreover, restricted user rights and Planet-Lab policy [23] prevent an adaptation of the NTP server configurations or to manually adjust the internal clock of a Planet-Lab node. Thus, we had to re-synchronize the packet timestamps at all nodes by adding a correction term to each measurement.

To evaluate the correction term and estimate the error, we periodically measured the NTP offset at each node in parallel to the probe of the packet tracking framework. Each pair of the local system time and discovered NTP offset ($\widehat{T}_j(k)$, $\widehat{O}_j(k)$) was written to a file for later resynchronization. Figure 1 gives an overview of the measurement setup with the packet tracking framework and scripted offset retrieval.

The resynchronization approach works as follows (Table I gives an overview of the symbols). A packet p_i sent by the source is captured at node j at time $\widehat{T}_j(p_i)$. Assuming the clock of node j has an offset $O_j(p_i)$ according to the reference NTP server when receiving packet p_i, the correct timestamp $T_j(p_i)$ for p_i can be written as

$$T_j(p_i) = \widehat{T}_j(p_i) + O_i^j \tag{1}$$

The exact offset O_i^j for each packet p_i captured at node j is unknown. From the offsets continuously recorded at events k, we can obtain $\widehat{T}_j(k)$ and $\widehat{T}_j(k+1)$ so that

$$\widehat{T}_j(k) \leq \widehat{T}_j(p_i) \leq \widehat{T}_j(k+1) \tag{2}$$

TABLE I
OVERVIEW OF USED VARIABLES

Variable	Meaning in Measurement
$\widehat{T}_j(\cdot)$	Sampled timestamp at event $\cdot$ and node j
$T_j(\cdot)$	Reference timestamp at event $\cdot$ and node j
$\widehat{O}_j(\cdot)$	Sampled offset at event $\cdot$ and node j
$O_j(\cdot)$	Offset to reference time at event $\cdot$ and node j
$\overline{O}_j(\cdot)$	Average offset at event $\cdot$ and node j

Based on this, we can estimate the offset $O_j(p_i)$ for the packet p_i at node j using the measured offset samples $\widehat{O}_j(k)$ and $\widehat{O}_j(k+1)$ as follows

$$O_j(p_i) \approx \widehat{O}_j(k) + \frac{\widehat{T}_j(p_i) - \widehat{T}_j(k)}{\widehat{T}_j(k+1) - \widehat{T}_j(k)} \cdot (\widehat{O}_j(k+1) - \widehat{O}_j(k)) \tag{3}$$

The quality of these offset samples depends on the topological distance of node j to the reference NTP server. Fluctuating or asymmetric travel times between the node and the NTP server induce errors in the offset calculation of the NTP protocol. Figure 2 compares the deviation from the mean offset for a node close and a node distant to the reference NTP server, which results in almost stable and variable offsets respectively. In the latter case, the offset varies about $\pm$ 3 ms around the mean offset. To estimate these errors, we calculated the weighted mean offset $\overline{O}_j(k)$ over the last 32 ($= u$) samples,

$$\overline{O}_j(k) = \frac{u-1}{u} \cdot \overline{O}_j(k-1) + \frac{1}{u} \cdot \widehat{O}_j(k) \tag{4}$$

To estimate the accuracy of the proposed method, we analyzed the observed clock offsets for all nodes. Fig. 3 shows the distribution of errors, i.e., the deviation from the mean offset taken over all nodes. We observed an average deviation of 2.009 ms from the mean offset, with a standard variation of 4.660 ms. Note the accumulation points at around $\pm$ 3 ms, these correspond to the unstable offset estimation by the reference NTP server as observed for nodes distant to the reference NTP server (see Fig. 2(b)).

IV. Measurement Setup and Results

A. Metrics

1) One-way delay: The one-way delay is the unidirectional latency of a packet to travel from source to receiver. For multicast distribution, a single packet experiences variable delays at different receivers after being distributed and replicated along the distribution tree.

2) Relative delay penalty: The relative delay penalty is the ratio of end-to-end latency for packets sent between a pair of nodes using hybrid transmission and the corresponding unicast latency. To account for the entire group of receivers

in multicast, we calculated the *relative average delay penalty* (RAD) and *relative maximum delay penalty* (RMD) [19]. RAD (RMD) is the ratio between the average (maximum) one-way packet delay using a hybrid multicast scheme and the average (maximum) delay using IP unicast.

3) Link stress: Link stress is given by the number of replicated packet traversing the same physical link, when sending a message to multiple receivers. Native IP multicast has an optimal link stress of 1 on all links. Hybrid multicast and N-times unicast on the other hand have average link stress greater 1. Thus, link stress indicates the load and bandwidth consumption induced on a physical link by a certain group communication scheme.

4) Out-degree: The out-degree represents the number of packet fan-out at a single node. In the N-times unicast scenario, packet replication is done by the source that sends one packet to each receiver. For multicast transmission, packet replication takes place within the network along the group distribution tree at the latest point possible to reach all receivers. Characteristic multicast trees in the Internet are rather tall than wide, leading to a higher probability at moderate fan-outs.

5) Path length: Path length is the number of routing hops between a sender and receiver. For an unbiased comparison of all measurement scenarios, we counted IP hops on the network layer as the absolute path length. Thus, we had to resolve all Scribe overlay and tunnel links to the corresponding number of IP hops in the underlay.

6) Stretch: Routing stretch is the ratio of absolute end-to-end path length (hops) between two nodes in (hybrid) multicast and unicast. While a unicast route (often) follows the shortest path, the construction of multicast distribution trees can result in larger distances between the source and receiver nodes.

B. Network Scenarios

1) N-times Unicast: The unicast scenario (see Fig. 4(a)) considers a source node that successively sends each packet to all receiving nodes in individual flows. This simple approach roughly resembles the minimal implementation of group communication in the current Internet.

2) Hybrid Multicast with switched edge domains: In the first hybrid multicast scenario (see Fig. 4(b)), we connected all selected Planet-Lab sites (that is edge networks) through an overlay multicast domain. At each Planet-Lab site, one node was configured as an *Interdomain Multicast Gateway* (IMG) forwarding group data from the overlay to its local IP multicast domain. A second node acts as IP multicast receiver and measurement endpoint.

3) Hybrid Multicast with routed edge domains: The second hybrid scenario (see Fig. 4(c)) is an enhancement of the first, in which we grouped multiple Planet-Lab sites and their nodes to form larger IP multicast islands. The nodes for each IP multicast domain were chosen based on closeness defined by the geographical coordinates of the corresponding Planet-Lab sites. For each IP multicast domain, we chose one node to act as an IMG again, other former IMGs were configured as IP multicast routers to forward traffic between Planet-Lab sites. The forwarding nodes have been inter-connected by tunnels.

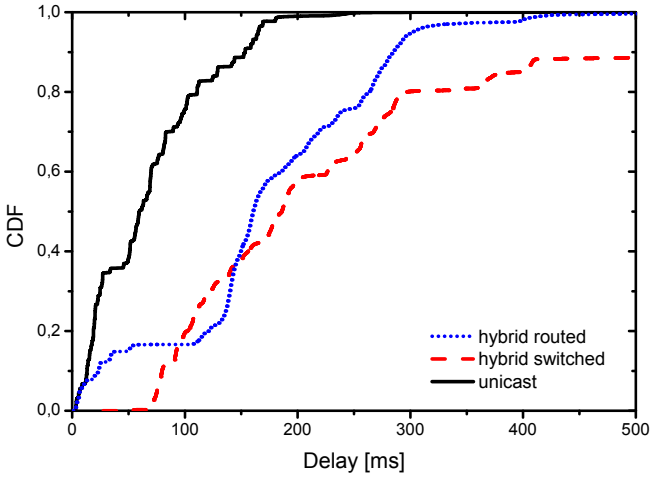


Fig. 5. End-to-end one-way delay distribution for unicast and both hybrid multicast scenarios (switched, routed).

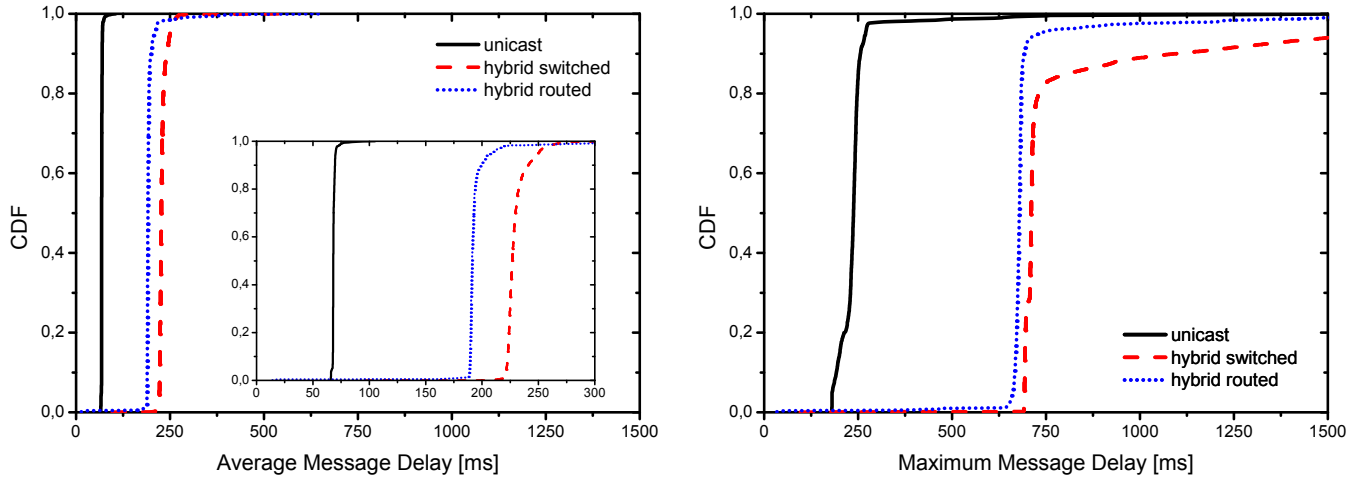
C. Measurement Setup

We extensively used the Planet-Lab testbed environment for our measurements. We selected more than 200 nodes from over 100 different Planet-Lab sites. For each scenario, we configured 100 nodes as receivers; for the hybrid scenarios, further nodes were configured as needed to host IMGs or multicast routers respectively. In general, each Planet-Lab site hosts only two nodes. We do not consider that as a problem, as additional nodes in all scenarios would experience identical delays. On each node, we deploy the H∀Mcast middleware, as well as the packet tracking probe, and the script for timestamps resynchronization. We used release 0.4 of H∀Mcast that supports IPv4/6 multicast, IGMP/MLD Proxy-based tunneling, and Scribe ALM. The source and all receivers were configured to use IP unicast or multicast, while the IMGs joined a Scribe overlay network to inter-connect the multicast edge networks and forward group data between multicast domains. For our experiments, we sent packets with a payload of 1000 Bytes and in an interval of 1 second. Note, the actual packet size sent on the wire depends on the multicast technology in use. IP multicast adds IP + UDP headers and Scribe multicast adds IP + UDP headers as well as a header for the P2P overlay protocol, for example.

D. Results & Discussion

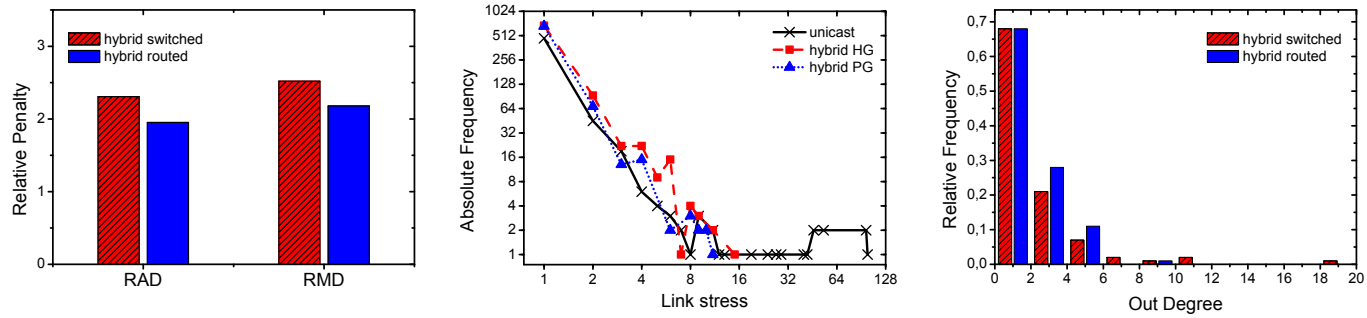
We first analyzed the end-to-end one-way delay. Fig. 5 compares the delay distribution for the unicast, hybrid multicast switched, and hybrid multicast routed scenarios. It is worth noting that the error intervals derived in Section III are small when compared to the measurement results.

As naturally expected, unicast achieves an overall lower one-way delay when compared to hybrid multicast. Hybrid multicast packet distribution on a global scale adds an extra



(a) (b)

Fig. 6. Comparison of average and maximum one-way packet delay for unicast and both hybrid multicast schemes.



(a) (b) (c)

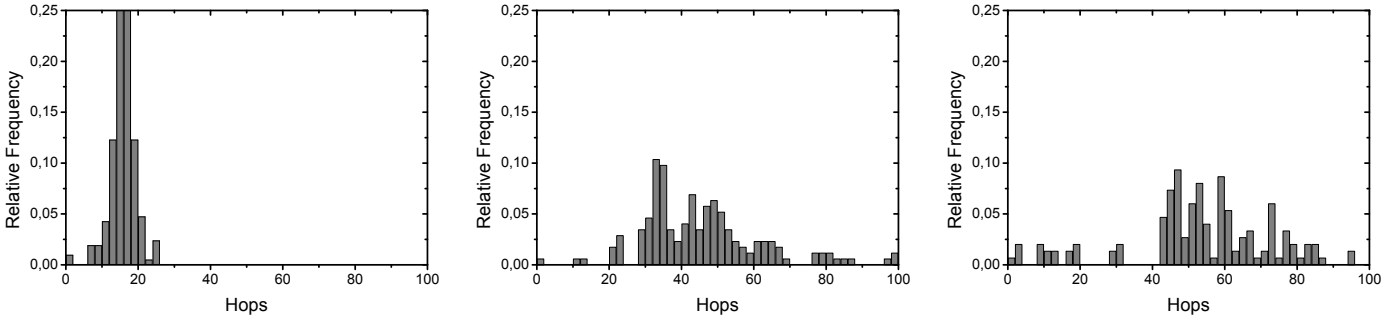
Fig. 7. Comparison of link stress, out degree, and delay penalty.

average delay of 100 ms to the packets. Still about 50 % of the packets reach its receivers within 150 ms, the critical bound for conversational real-time applications identified by the ITU [24]. Remarkably, for the hybrid routed scenario 75 % of the packets are received within 250 ms, versus 350 ms for the hybrid switched scenario. We observed that due to the larger IP domains consisting of multiple Planet-Lab sites in the routed hybrid multicast scenario, nodes close to the source are within the same IP multicast domain and have a delay close to the unicast scenario. But nodes distant to the source are located in another IP domain and multicast data is forwarded through the Scribe multicast domain. Depending on the position of the rendezvous point relative to the source, this induces an extra increase in delays at around 100 ms, see Fig. 5. These observations indicate that routing in the overlay via a rendezvous point chosen independent of topological constraints imposes a severe performance penalty. In previous work [25], we have shown how to mitigate this specific penalty.

In comparison with our previous semi-empirical studies [20], we find that the average delay for Scribe-based hybrid multicast of about 175 ms is reasonably well represented, whereas shortest unicast paths are 68 ms on average and faster than expected. The delay variation of 53 ms for unicast and 132 ms in the hybrid routing exceeds the a priory estimators uniformly by a factor of two. These jitter measurements include scheduling effects of overloaded Planet-Lab nodes and may be enhanced therefore.

In Fig. 6, we compare the average and maximum one-way delay per packet. We also calculated the relative average (RAD) and relative maximum delay (RMD) penalty for hybrid multicast compared to unicast (cf., Fig. 7(a)). While both hybrid scenarios stay close with the average delay, the routed scenario limits the maximum packet delay to 800 ms. This is another indication of the strong effect imposed by the cross-rendezvous-point routing. The maximum packet delay is an important metric for real-time group applications as it indicates the maximum latencies for a message to reach all group members. Relative delay penalties as observed in Fig. 7(a) are typical for a Scribe overlay network.

Fig. 7(b) compares the link stress induced by unicast and both hybrid multicast scenarios for a group of 100 receivers.



(a) Unicast (b) Hybrid Multicast (switched) (c) Hybrid Multicast (routed)

Fig. 8. Distribution of path length (hops) from source to receivers.

Even though the dominant majority of links exhibit a link stress of 1 in all scenarios, unicast packets traverse a number of links at a very high redundancy. These links are close to the source, where the link stress amounts to the number of receivers. Stress gradually declines with the distance from the source, as expected. In total, unicast exhibits an average link stress of 2.5 with a large standard deviation of 8.5. In contrast, both hybrid multicast schemes limit the load at all physical links with averages and standard deviations close to 1. Link stress reaches its maximum at 15 (switched) and 11 (routed) respectively.

A comparable measure of infrastructure load is given by the fan-out degree attained at intermediate nodes. Thereby, a moderate and fairly uniform replication load is desired at all forwarding nodes. Fig. 7(c) shows the out-degree distribution for both hybrid multicast scenarios. Routed hybrid multicast shows a more balanced distribution shape than the switched scenario, which limits replications to the overlay. Unfavorable balance is a common phenomenon of the Scribe overlay that – due to asymmetric routing – exhibits an enhanced fan-out at the rendezvous point with little replication in the subsequence. This is well reflected in almost doubling standard deviations, i.e., 1.3 for the hybrid routed versus 2.4 in the switched case.

The distributions of path lengths for the three measurement scenarios are shown in Fig. 8. As expected, unicast path length is shorter than both hybrid multicast scenarios and reaches a maximum of 25 IP routing hops. The average path length is 15.3 hops for unicast, 47.61 hops for hybrid multicast switched and 54.52 hops for the routed scenario respectively. It is noteworthy to mentioned that the path lengths in the hybrid multicast routed scenario are also affected by connecting multiple Planet-Lab sites by tunneling to constitute larger IP multicast domains.

A further structural analysis of the logical multicast distribution trees of both hybrid multicast scenarios revealed that the group tree of hybrid multicast switched has a depth of 5, with a depth of 4 for the tree in Scribe overlay and a last hop in the switched edge network. The group tree of hybrid multicast routed has an overall depth of 7, with a depth of 3 for the overlay part and up to 4 in the IP multicast domains. Note that these values correspond to the logical depth of the group

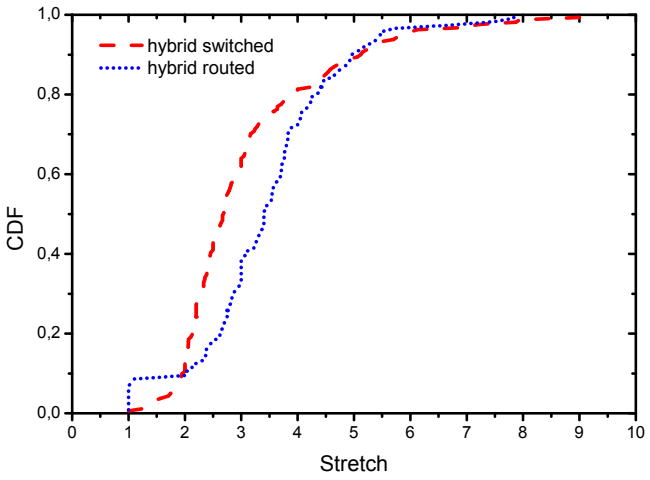


Fig. 9. Relative routing stretch for hybrid multicast compared to unicast

distribution tree without resolving corresponding IP hops in the underlay. Thus, a Scribe overlay link and an IP multicast tunnel link likewise count as one logical tree hop. On the one hand, an increased tree depth leads to a lower link stress for the hybrid routed case. On the other hand, it also raises the average path length from source to receivers and consequently amounts to a higher routing stretch as shown in Fig. 9. The average routing stretch of hybrid multicast switched is 3.1 with a standard deviation of 1.5. Hybrid multicast routed has an average stretch of 3.5 with a standard deviation of 1.3. It is worth mentioning that nearly 10 % of all links in the routed hybrid multicast scenario have a stretch of 1, these correspond to nodes within the IP multicast domain of the source.

V. Conclusion and Outlook

This work reported on large-scale measurements of hybrid multicast. Built upon a careful tracking of individual packets, we evaluated the one-way delays of data frames within a global distribution system. A central part of our methodology was a continuous monitoring and correction of the distributed clocks in combination with in-band error estimations. We used the adaptive system-centric approach of H∀Mcast to deploy

various multicast technologies transparently on Planet-Lab.

Our findings in the heterogeneous, unclean Planet-Lab environment revealed an overall increased delay of about 100 ms that pushes quality of experience (QoE) in conversational multimedia applications to its limits. Still about 50 % of globally distributed nodes experience a transmission delay below 150 ms, which is the ITU recommendation for real-time compliance in this realm. Performance results easily comply to streaming applications like IPTV and largely outperform current peer-to-peer solutions, thus supporting the use of hybrid multicast solutions in today's dominant application area of group distribution.

In future work, we will concentrate on a further analysis of the performance impacts of individual components of the hybrid system. We expect significant performance potentials from improving interdomain multicast gateway processing and placement, as well as from the underlying overlay multicast schemes.

Additional deployment scenarios are on our schedule, as well. In particular, we will investigate settings that include cloud-based network services for spanning wider parts of the Internet. Borrowing network services from selected, widely distributed ASes of cloud providers yields the promise of fast and reliable backbone components in our global setting.

ACKNOWLEDGMENT

The authors would like to thank Nora Berg, Dominik Charousset, Sebastian Wölke and Sebastian Zagaria for their supporting work. This work is funded by the Federal Ministry of Education and Research (BMBF) of Germany within the project H∀Mcast and the G-Lab initiative, see http://hamcast.realmv6.org.

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] M. Wählisch, T. C. Schmidt, and S. Venaas, "A Common API for Transparent Hybrid Multicast," IRTF, IRTF Internet Draft – work in progress 05, July 2012. [Online]. Available: http://tools.ietf.org/html/draft-irtf-samrg-common-api

[3] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network Magazine*, vol. 14, no. 1, pp. 78–88, 2000.

[4] K. Almeroth, "The evolution of multicast: from the MBone to interdomain multicast to Internet2 deployment," *IEEE Network*, vol. 14, no. 1, pp. 10–20, Jan/Feb 2000.

[5] G. Bumgardner, "Automatic Multicast Tunneling," IETF, Internet-Draft – work in progress 14, June 2012.

[6] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker, "Application-Level Multicast Using Content-Addressable Networks," in *Proc. of 3rd Intern. Workshop on Network Group Communication (NGC'01)*, ser. LNCS, J. Crowcroft and M. Hofmann, Eds., vol. 2233. London, UK: Springer–Verlag, Nov. 2001, pp. 14–29.

[7] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and Modeling of Computer Systems*. New York, NY, USA: ACM Press, 2000, pp. 1–12.

[8] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," in *Proc. of SIGCOMM '02*. New York, NY, USA: ACM Press, 2002, pp. 205–217.

[9] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination," in *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '01)*, J. Nieh and H. Schulzrinne, Eds. New York, NY, USA: ACM, 2001, pp. 11–20.

[10] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 100–110, 2002.

[11] B. Zhang, W. Wang, S. Jamin, D. Massey, and L. Zhang, "Universal IP multicast delivery," *Computer Networks*, vol. 50, no. 6, pp. 781–806, 2006.

[12] M. Wählisch and T. C. Schmidt, "Between Underlay and Overlay: On Deployable, Efficient, Mobility-agnostic Group Communication Services," *Internet Research*, vol. 17, no. 5, pp. 519–534, November 2007. [Online]. Available: http://www.emeraldinsight.com/10.1108/10662240710830217

[13] S. Lu, J. Wang, G. Yang, and C. Guo, "SHM: Scalable and Backbone Topology-Aware Hybrid Multicast," in *16th Intern. Conf. on Computer Communications and Networks (ICCCN'07)*, August 2007, pp. 699–703.

[14] X. Jin, K.-L. Cheng, and S.-H. G. Chan, "Island multicast: combining IP multicast with overlay data distribution," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 1024–1036, 2009.

[15] S. Meiling, D. Charousset, T. C. Schmidt, and M. Wählisch, "System-assisted Service Evolution for a Future Internet – The HAMcast Approach to Pervasive Multicast," in *Proc. of IEEE GLOBECOM 2010, Workshop MCS 2010*. Piscataway, NJ, USA: IEEE Press, Dec. 2010, pp. 913–917.

[16] ——, "HAMcast: Evaluation of a High Throughput Middleware for Universal Multicast," in *11th Würzburg Workshop on IP: Joint EuroNF, ITC, and ITG Workshop on Visions of Future Generation Networks (EuroView2011)*, Würzburg, Aug. 2011.

[17] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Measurement Considerations for Assessing Unidirectional Latencies," *Internetworking: Research and Experience*, vol. 4, pp. 121–132, 1993.

[18] L. Lao, J. hong Cui, M. Gerla, and D. Maggiorini, "A comparative study of multicast protocols: Top, bottom, or in the middle?" in *In Proc. of the 8th IEEE Global Internet Symposium (GI'05)*. Piscataway, NJ, USA: IEEE Press, 2005, in conjunction with IEEE INFOCOM'05.

[19] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An Evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays," in *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003)*, vol. 2. Washington, DC, USA: IEEE Computer Society, 2003, pp. 1510–1520.

[20] M. Wählisch and T. C. Schmidt, "An a Priori Estimator for the Delay Distribution in Global Hyrid Multicast," in *Proc. of the ACM SIGCOMM CoNEXT. Student Workshop*. New York: ACM, Dec. 2009, pp. 19–20.

[21] T. Santos, C. Henke, C. Schmoll, and T. Zseby, "Multi-hop Packet Tracking for Experimental Facilities," in *Proc. of the ACM SIGCOMM 2010*. New York, NY, USA: ACM, 2010, pp. 447–448. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851256

[22] D. L. Mills, "On the Accuracy and Stablility of Clocks Synchronized by the Network Time Protocol in the Internet System," *SIGCOMM Comput. Commun. Rev.*, vol. 20, no. 1, pp. 65–75, Dec. 1989. [Online]. Available: http://doi.acm.org/10.1145/86587.86591

[23] N. Spring, L. Peterson, A. Bavier, and V. Pai, "Using PlanetLab for Network Research: Myths, Realities, and Best Practices," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 17–24, 2006.

[24] ITU, "G.114 - One-way transmission time," ITU, Recommendation - Telecommunication Union Standardization Sector, 05 2003.

[25] M. Wählisch, T. C. Schmidt, and G. Wittenburg, "On Predictable Large-Scale Data Delivery in Prefix-based Virtualized Content Networks," *Computer Networks*, vol. 55, no. 18, pp. 4086–4100, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2011.07.020