H∀Mcast: Evaluation of a High Throughput Middleware for Universal Multicast

Sebastian Meiling, Dominik Charousset, Thomas C. Schmidt
{sebastian.meiling,dominik.charousset}@haw-hamburg.de, t.schmidt@ieee.org
Internet Technologies 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*—The H∀Mcast-architecture provides a universal group-communication, the concept combines an abstract naming scheme for multicast groups, a common multicast API and a service-middleware for endsystems. In this work we present the implementation and evaluation of the H∀Mcast-middleware prototype. Besides an overview on our software-prototype and its components, we focus on the results of the performance evaluation.**

I. Introduction

Today, many Internet applications and services, such as IPTV, MMORGs, and social networks, are grounded on the principle of group communication. Data distribution within a group of multiple senders and receivers is most efficiently done using multicast. Nevertheless, despite the variety of existing multicast technologies, these application often rely on proprietary techniques based on IP-unicast, such as proxies or cache-servers. The two key problems are: (a) incompatible application interfaces between different multicast technologies (e.g. IPv4/ IPv6 and ASM/SSM), and (b) divergent deployment states of multicast services. This forces developers and programmers of group applications to choose a multicast-technology at compile-time, unaware of the availability at run-time. Often the result is, that multicast is not used at all.

To overcome these obstacles we proposed the H∀Mcast-architecture [1] for a universal multicast-service. Its concept combines an abstract naming scheme (based on URIs), a common multicast-API and a service-middleware on endsystems, as well as gateways to cross technological and administrative network borders. The focus of this paper lies on the performance evaluation of the H∀Mcast-prototype.

II. The H∀Mcast-Middleware Prototype

H∀Mcast aims to provide a universal group communication service based on a hybrid approach. The concept is independent of the availability of a certain multicast technology, such as IP-multicast [2] or overlay-multicast, and does not rely on a complete deployment of the H∀Mcast-architecture on every endsystem. On the contrary it allows for an incremental deployment within networks and attached nodes.

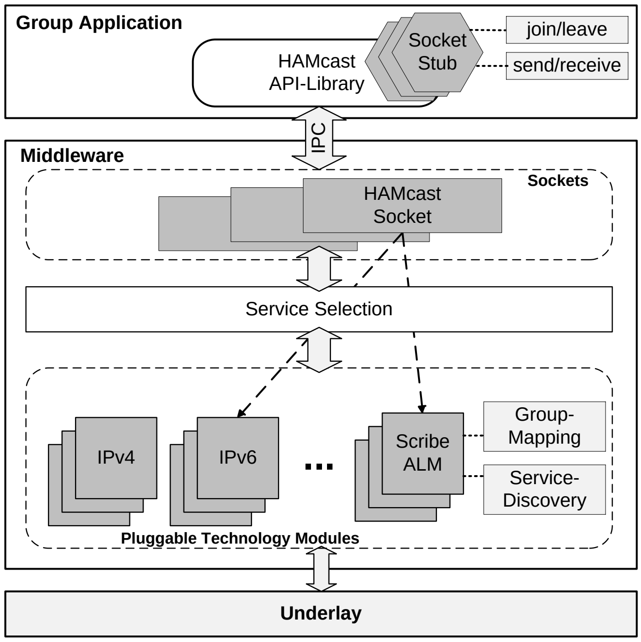
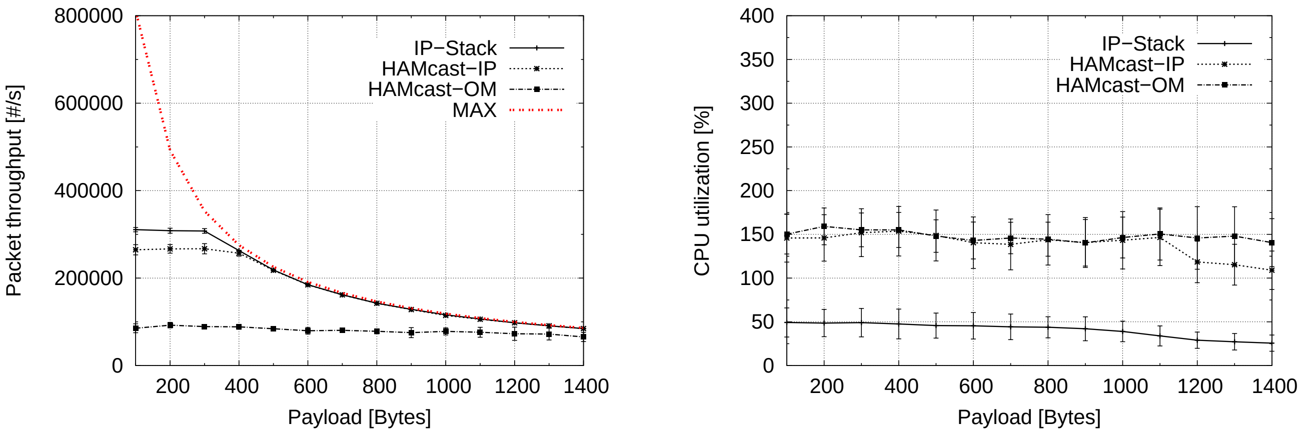


Fig. 1. System-architecture of the H∀Mcast-prototype. Showing the multicast-API, service-middleware and pluggable technology modules.

The system-architecture of H∀Mcast (see fig. 1) consists of three core components:

- a technology-transparent API [3]
- a middleware abstraction layer
- multicast technology modules

We implemented a prototype of the H∀Mcast-middleware in C++ (using the Boost library) as a user-space process, running once on an endsystem. The multicast-API was implemented as a library in C++, but is also available as a Java package. At the moment, there are two multicast technology modules available for H∀Mcast, a module for IP-multicast (H∀Mcast-IP) and one for overlay-multicast based on Scribe [4] (H∀Mcast-OM). To connect the H∀Mcast-middleware process with a client application, we developed an IPC-protocol, that uses *localhost* sockets.



(a) Packet throughput of receiver.

(b) CPU utilization of receiver

Fig. 2. Performance results of the H∀Mcast-middleware.

III. Evaluation

To analyze the system performance of our middleware prototype on endsystems we compared different multicast technologies (IP and OM[1]) using H∀Mcast against IP-multicast using the Linux IP-Stack. For the evaluation we setup two nodes in a sender-receiver scenario, they were equipped with a quad-core CPU and 8 Gb of RAM, connected over 1 GBit/s network link. For each test we run 25 iterations with a duration of 40 seconds. Using a measurement interval of 1s we recorded throughput, packet loss and CPU utilization for packets with a payload between 100 to 1400 Bytes.

Fig. 2 presents an excerpt of the promising results of our performance evaluation for the H∀Mcast-middleware. It shows the packet throughput (fig. 2a) and CPU utilization (fig. 2b) at the receiver. Though CPU-utilization is higher compared to the IP-stack, our middleware-prototype can achieve similar throughput for payloads $>$ 500 Bytes (H∀Mcast-IP). The overall performance of H∀Mcast is correlated with the IPC throughput, but also depends on the implementation of multicast technologies. For instance, throughput of H∀Mcast-OM is influenced by its underlying P2P network protocol.

Currently we are conducting further experiments in the G-Lab testbed to evaluate the performance of our H∀Mcast-prototype in hybrid multicast scenarios measuring additional metrics such as RTT and delay. Therefore we apply the packet tracking framework [5] developed by the Fraunhofer FOKUS group, a partner within the G-Lab initiative.

Acknowledgment

The authors would like to thank Fabian Holler, 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.

[1] *Overlay Multicast*, here Scribe.

References

[1] 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.

[2] 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.

[3] M. Waehlisch, T. Schmidt, and S. Venaas, “A Common API for Transparent Hybrid Multicast,” IETF, Internet-Draft – work in progress 01, March 2011.

[4] 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.

[5] T. Santos, C. Henke, C. Schmoll, and T. Zseby, “Multi-hop Packet Tracking for Experimental Facilities,” in *Proc. of the ACM SIGCOMM 2010 conference (SIGCOMM’10)*, ser. SIGCOMM ’10. New York, NY, USA: ACM, 2010, pp. 447–448. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851256