Demo Abstract: A Showcase on Usage and Monitoring of the HΛMcast-Architecture 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 principle of group communication can be found in many applications, that exchange data between multiple senders and receivers. While multicast technologies provide efficient data distribution mechanisms, they also raise a variety of issues concerning service deployment, availability and usage. HΛMcast provides an architecture to overcome existing problems of current multicast technologies and offers a universal group-communication service. The design of the HΛMcast-architecture combines the concepts of an abstract naming scheme for multicast groups, a common multicast API and a service-middleware for endsystems. In this demo proposal we present a software prototype of the HΛMcast-architecture. For our demo presentation we build a live-lecturing software that uses the multicast API and service of HΛMcast to send and receive a video-stream and group-chat. Further, we introduce a multicast monitoring framework, that we developed to visualize network state, multicast trees and node information.**

I. Introduction

Many Internet services and applications, such as IPTV, online multiplayer games, video conferencing, and even social networks, are based on the principle of group communication. In contrast to classic client-server scenarios that are based on unicast, group communication is most efficiently realized using multicast. Multicast enables data distribution between one or more senders and a group of receivers, while effectively saving network bandwidth - opposed to flooding the network with broadcasts. Nevertheless, despite the variety of existing multicast technologies, most services and application often rely on IP-unicast enhanced with proprietary techniques, such as proxies or cache-servers. The major issues of multicast 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 group application developers to choose a multicast-technology and corresponding API at compile-time, unaware of actual availability at run-time. Moreover, usage of multicast has conflicting incentives between ISPs (transit), content providers (sender) an users (receiver) [1]. As a result of these issues, multicast is hardly used by applications developers and most ISPs provide only restricted multicast services within their network domains. However, multicast is often available in campus and enterprise networks. Unfortunately, there is no general interconnection of these edge-networks over the Internet, thus they remain isolated *islands* (so called *walled gardens*).

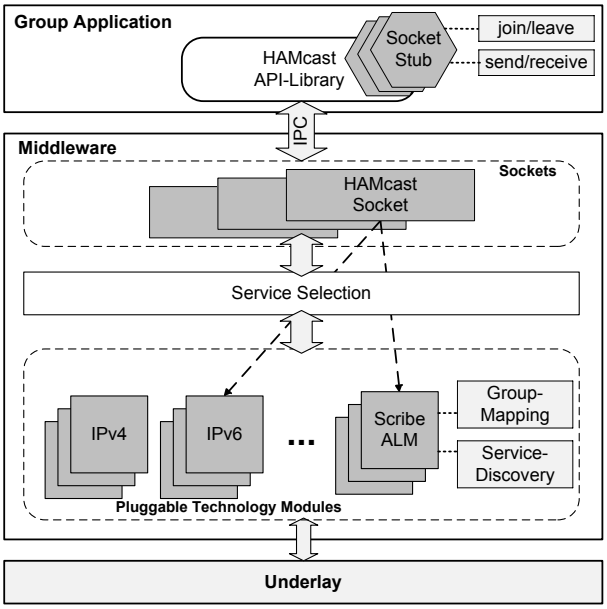
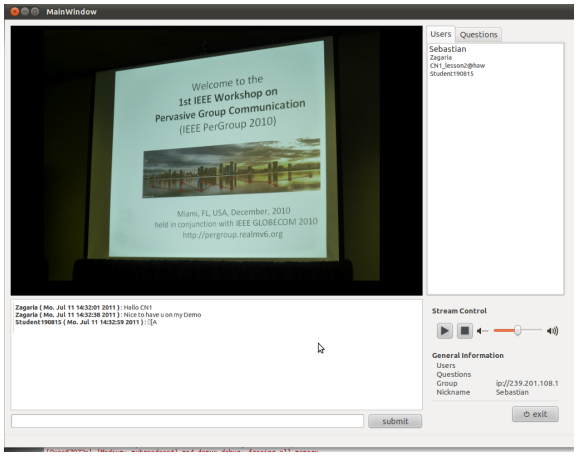
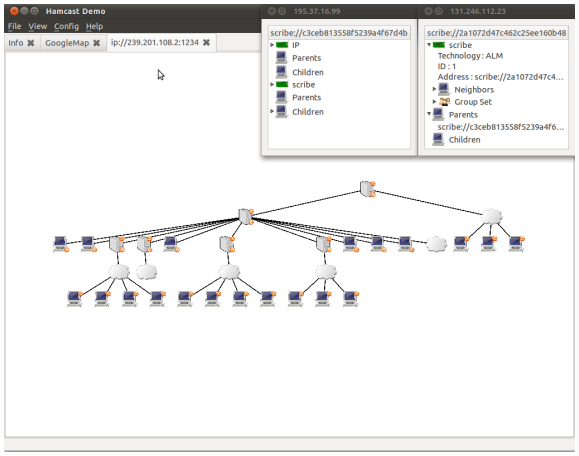


Fig. 1. Overview on system architecture.

The HΛMcast-architecture [2] overcomes these obstacles and provides concepts to enable a universal multicast service. HΛMcast combines an abstract naming scheme, a common multicast API and system-centric middleware component as well as gateways (IMGs) to cross administrative or technological borders. The architecture is independent of the availability of a certain multicast technology, such as IP-multicast or overlay-multicast, and does not rely on a complete deployment of the HΛMcast-middleware. On the contrary, HΛMcast allows for an incremental deployment within networks and on attached endsystems.



(a) Live Lecturing Software



(b) Monitoring Software

Fig. 2. Screenshots of the two demo applications, demonstrating usage of the prototype implementation.

In our demonstration we present a showcase on the software prototype of the HΛMcast-architecture. Therefore, we implemented an application for live lecturing with video-streaming and chat, and also developed a monitoring framework to analyze and visualize the multicast network state. Both software tools utilize the API and multicast service of HΛMcast, and provide insights on usage of its components.

II. Prototype Implementation

We implemented the HΛMcast-architecture as a software prototype in C/C++, relying on some extra libraries, such as *boost* [3]. Our prototype consists of three core components:

- a common API, for transparent multicast access
- a user-space middleware, running once per host
- dynamic loadable multicast technology modules

The multicast API of HΛMcast is provided as a C++ library and is in full conformance with the IRTF draft [4]. The HΛMcast-middleware provides abstraction from the available multicast-technologies and connects via a self developed, lightweight IPC-protocol with the client application. Group applications can access multicast by utilizing service-calls from the API-library. Multicast technology modules are dynamically loaded by the HΛMcast-middleware at runtime as specified in a local configuration file. Each module may provide a service discovery method, to investigate the local network environment and to ensure necessary service requirements. If service discovery fails, i.e. corresponding multicast technology is not available, the module self terminates and is unloaded by the middleware, saving hardware resources. For example, enabling IP-multicast makes only sense if a IGMP or MLD-querier exists.

Currently, there are two multicast technology modules available for HΛMcast, one for IP-multicast [5] and the other for overlay-multicast based on Scribe [6]. Figure 1 gives an overview on the HΛMcast-architecture and its components.

The HΛMcast-prototype is available as a compressed archive (tar.gz) for download on http://hamcast.realmv6.org.

III. Demo Presentation

Our demo presentation consists of two parts. First, we present a live-lecturing software as a sample use-case of the HΛMcast-architecture. The software allows to send and receive a video-stream via multicast utilizing the common multicast API for group communication provided by HΛMcast. Further, it offers a multicast chat to pose questions to the tutor and answer them within in the group of participants. Fig. 2a shows a screenshot of the application.

Second, we demonstrate a framework to monitor multicast nodes and visualize the distribution tree of multicast groups. The framework consists of a monitoring viewer and a daemon for endhosts. The latter is a small probe, that sends node specific information, e.g. joined groups, to the monitoring viewer. Fig. 2b shows a screenshot of the monitoring viewer with the visualization of small group tree and node details.

IV. Demo Software Components

For our demo setup we create an overlay-multicast domain using globally distributed nodes from the Planet-Lab [7] and G-Lab testbeds [8]. Within this overlay-domain we deploy multiple IMGs to interconnect multicast-enabled edge-networks, namely the sender site at HAW Hamburg and a receiver at the demo site.

For our demo presentation we make use of the following software tools:

- HΛMcast-middleware – runs on all nodes, to enable universal multicast service
- IMG daemon – to connect sender/receiver domain with overlay-multicast domain
- VideoStreamer – uses the HΛMcast-API to send/receive video stream and chat messages
- Monitoring dameon – a small probe, that runs on endhosts to discover multicast distribution tree and node information
- Monitoring viewer – a software to collect and visualize data from monitoring daemons

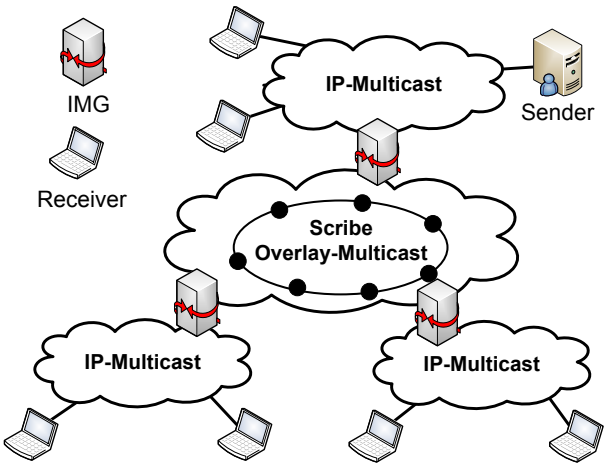


Fig. 3. Network topology example for setup of demo presentation.

V. Demo Requirements

The local demo setup consists of three interconnected laptops. One works as an IMG to connect the local network to the overlay-multicast domain. Another is a receiver for the lecturing software and the third runs our monitoring viewer to visualize multicast network topology and state. Setup will take about 30 to 60 min. Further, we have the following requirements for our demo:

- a table with space for 3 Laptops (around 2-3 m^2)
- Internet access, with at least 2 public IP-addresses
- power supply for hardware equipment
- a poster board, to illustrate demo details

Acknowledgment

The authors would like to thank Sebastian Zagaria for his contributions on implementing software tools for this demo presentation. Further we would like to thank Fabian Holler, Fabian Jäger, and Sebastian Wölke for their supporting work.

This work is funded by the Federal Ministry of Education and Research (BMBF) of Germany within the project H∀Mcast, see http://hamcast.realmv6.org. H∀Mcast is part of the G-Lab initiative.

References

[1] 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.

[2] 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.

[3] Boost, "Website of the Boost portable C++ source libraries," 2011. [Online]. Available: http://www.boost.org/

[4] M. Waehlisch, T. Schmidt, and S. Venaas, "A Common API for Transparent Hybrid Multicast," IETF, Internet-Draft – work in progress 03, July 2011.

[5] 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.

[6] 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.

[7] Planet-Lab, "Website of the Planet-Lab research network," 2011. [Online]. Available: http://www.planet-lab.org

[8] German-Lab, "Website of the German-Lab experimental facility," 2011. [Online]. Available: http://www.german-lab.de