HAVMcast: A High-Throughput Middleware for a Universal Future Internet Multicast Service

Sebastian Meiling
Dominik Charousset, Thomas C. Schmidt, Matthias Wählisch

iNET RG, Department of Computer Science
Hamburg University of Applied Sciences

August 2nd, 2011
1. Introduction
2. Architecture
3. Prototype
4. Evaluation
5. Conclusion
Motivation

- Network nodes are equipped with enhanced resources
  - advanced endsystem intelligence
  - support for complex operations
- This offers new service deployment options
- For example hybrid group communication
  - combine native IP and overlay multicast
  - endsystems participate in routing and forwarding
Objectives

- Decouple application development from service deployment
- Universal service access through standardized APIs
- Adaptive service instantiation at runtime, depending on local network environment and node capabilities
- Provide incremental deployment and service evolution
- HAVMcast exemplary implements a universal multicast service
Issues for Future Internet Services

- Globally available network services rely on:
  - uniform deployment within networks and endsystems
  - widely available (standardized) service APIs
- Requires support by vendors of hardware and operating systems
- Multicast specific issues:
  - Divergent deployment states of multicast technologies
  - Incompatible APIs for various multicast flavors
  - Conflicting incentives for usage and deployment
HVMcast Architecture

- System oriented multi service architecture
- Building blocks for new services:
  - technology transparent, universal service API
  - extended middleware functionality on endsystems
  - evolutionary, incremental service deployment
- HVMcast multicast service consists of:
  - an abstract naming scheme based on URIs (LocID split)
  - the common multicast API, conforms to IRTF draft [1]
  - a middleware component for endsystems
  - Interdomain Multicast Gateways (IMGs)
Incremental Deployment Scenario

- IMGs inter-connect heterogeneous multicast domains
- Group members (F, G) independent of domain or technology
- Coexistence of standard and HAMcast network stack
Overview

- Prototype implementation to demonstrate concepts of the H\(\forall\)Mcast architecture
- Utilizes hybrid group communication to provide a universal multicast service
- Late binding of multicast technology at runtime
- Implemented in C/C++ including *boost* library
- Multi OS support, currently runs on Linux and Mac OS
Components

- **Common multicast API**
  - Transparent multicast calls
  - Implemented as client library

- **Middleware Component**
  - User space daemon
  - Instantiated once per host

- **Service Modules**
  - Implement specific technology
  - e.g. IP multicast, Scribe
Evaluation

- Analyzing system performance of HAVMcast prototype
- Single sender-receiver scenario
- Hardware:
  - Hosts with QuadCore CPU, 8 GB RAM
  - Network link with bandwidth of 1 Gbit/s
- Comparison of HAVMcast-IP, HAVMcast-OM and IP multicast
- Metrics: throughput, loss and CPU usage
- Packet payload size from 100 to 1400 Bytes
Packet Throughput

![Packet Throughput Graph]

- IP-Stack
- HAMcast-IP
- HAMcast-OM
- MAX

Payload [Bytes] vs. Packet Throughput [Pakete/s]
Data Throughput

![Graph showing data throughput vs payload for different protocols](image)

- **IP-Stack**
- **HAMcast-IP**
- **HAMcast-OM**
- **MAX**

The graph illustrates the data throughput in MBit/s for varying payload sizes in bytes, comparing different protocols such as IP-Stack, HAMcast-IP, HAMcast-OM, and MAX.
Packet Loss

The diagram shows the packet loss percentage (%) for different payload sizes in bytes. The x-axis represents the payload size in bytes, ranging from 0 to 1400, while the y-axis represents the packet loss percentage, ranging from 0 to 5. Three lines are plotted for different protocols:

- **IP-Stack**: Represented by a solid line.
- **HAMcast-IP**: Represented by a dotted line.
- **HAMcast-OM**: Represented by a dashed line.

The graph indicates that the packet loss varies with the payload size and the protocol used.
CPU Utility

![Graph showing CPU Utility vs Payload for different protocols: IP-Stack, HAMcast-IP, HAMcast-OM. The x-axis represents Payload [Bytes] ranging from 0 to 1400, and the y-axis represents CPU Utility [%] ranging from 0 to 400. The graph illustrates the performance comparison across different payload sizes.](image-url)
Conclusion

- Prototype demonstrates feasibility of HAVMcast architecture
- Design enables extension and integration of new features
- Promising evaluation results verify prototype performance
- First deployment of prototype enables a hybrid group communication service in G-Lab testbed environment
- Active participation in IRTF SAM RG:
  - Standardization of common multicast API
  - Cooperation within research community
- Further information and download of prototype at: http://hamcast.realmv6.org
- Visit demo presentation here at EuroView
Thank you for your attention. Questions?
References I


References II
