

Implementation Report: Release of the System-centric Middleware Component for Universal Multicast

Matthias Wählisch, Thomas C. Schmidt,
Sebastian Meiling, Dominik Charousset
{waehlich, t.schmidt}@ieee.org

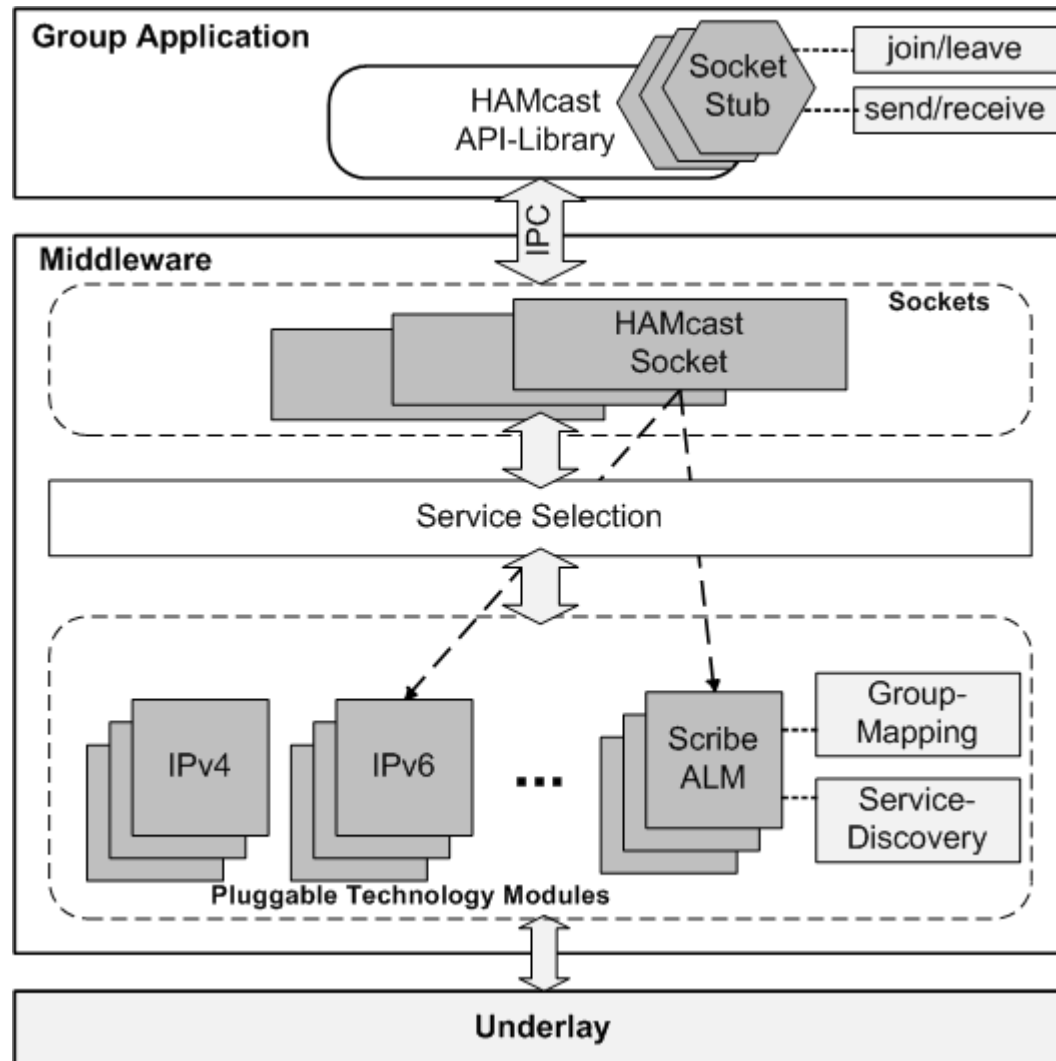
Agenda

1. Middleware Overview
2. Update Performance Evaluation
3. Mini-Tutorial
4. Conclusions & Outlook

Middleware Implementation at a Glance

- o Implements common multicast API
- o Implemented in C/C++ including boost library
- o Multi OS support:
 - Currently Mac OS and Linux
- o Several supported technologies
 - IPv4/v6, Scribe, ASM, SSM
- o Middleware runs as *user space daemon*
 - Implements transition between technologies
- o Application programmer just uses HAMcast library

Middleware Architecture

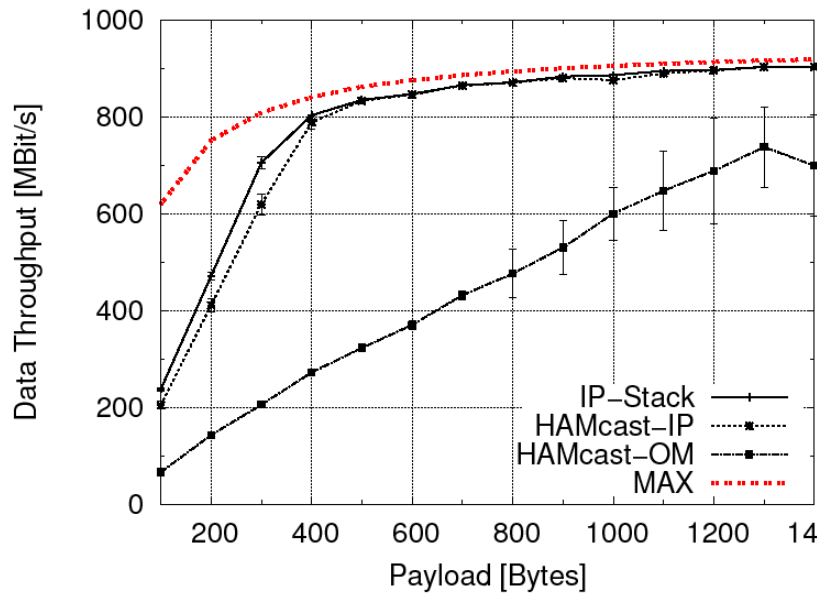


Evaluation Update

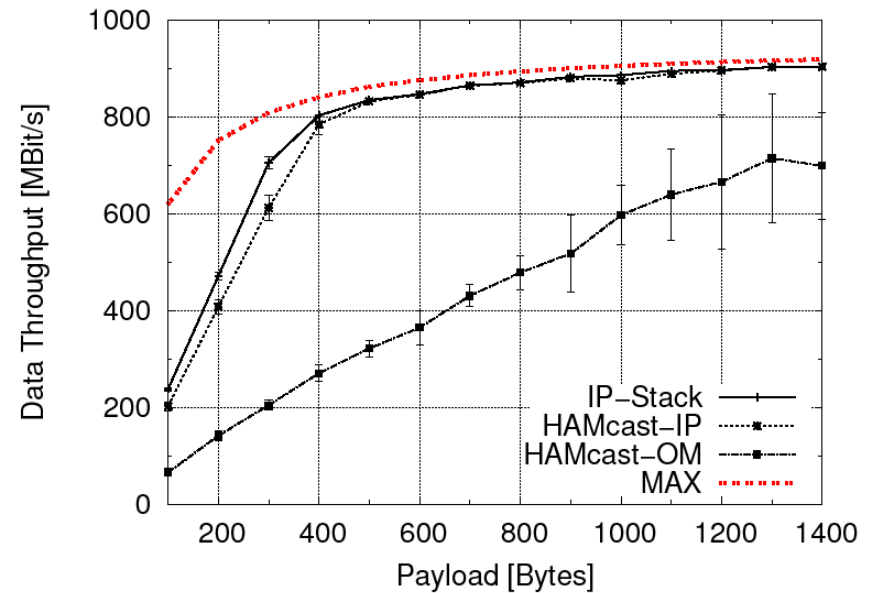
- o Analyzing system performance of HAMcast prototype
- o Single sender-receiver scenario
- o Hardware:
 - Hosts with QuadCore CPU, 8 GB RAM
 - Network link with bandwidth of 1 Gbit/s
- o Comparison of HAMcast-IP, HAMcast-OLM, and IP Multicast
- o Metrics: throughput, loss, and CPU usage
- o Packet payload size from 100 to 1,400 Bytes

Throughput

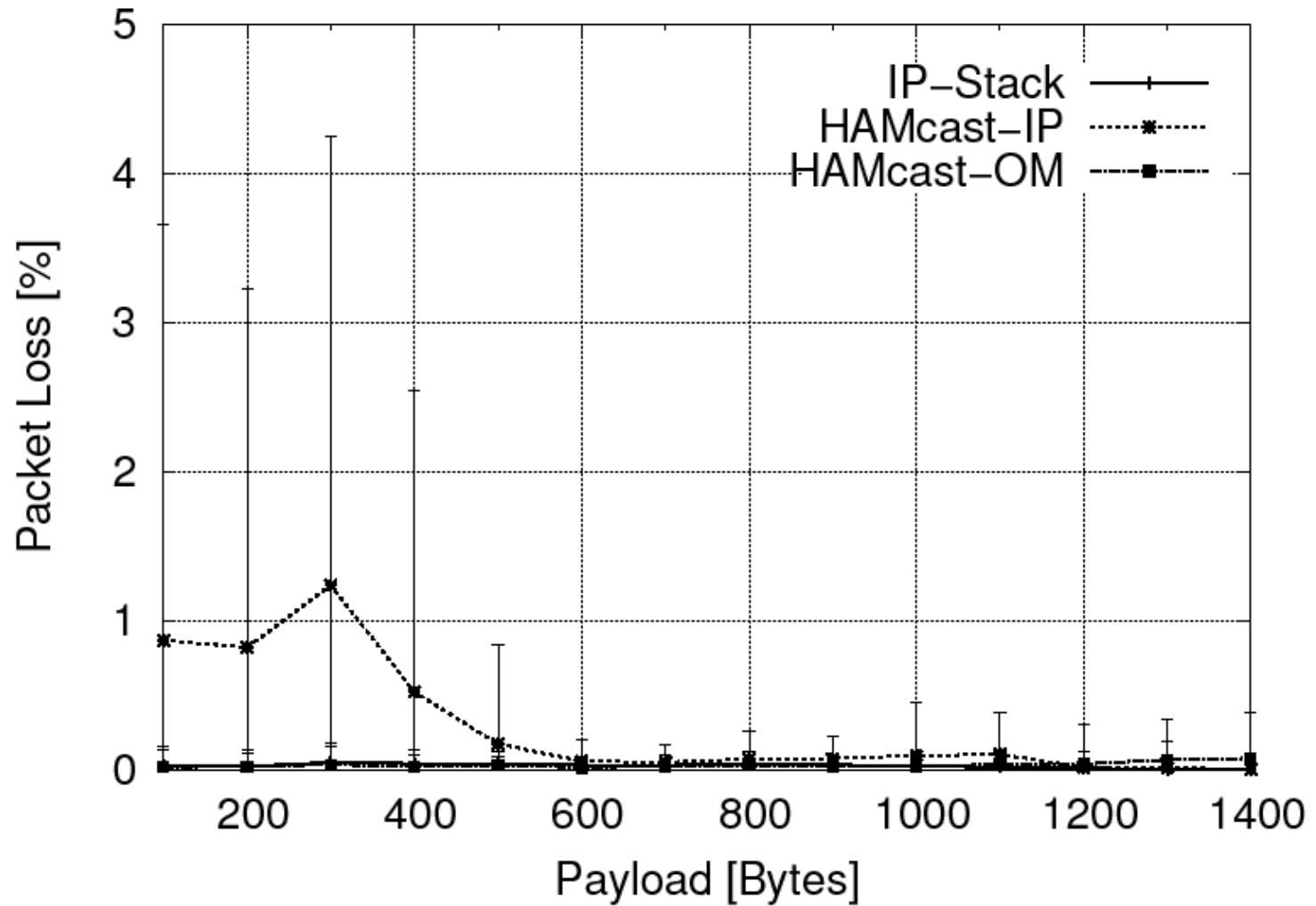
Sender



Receiver

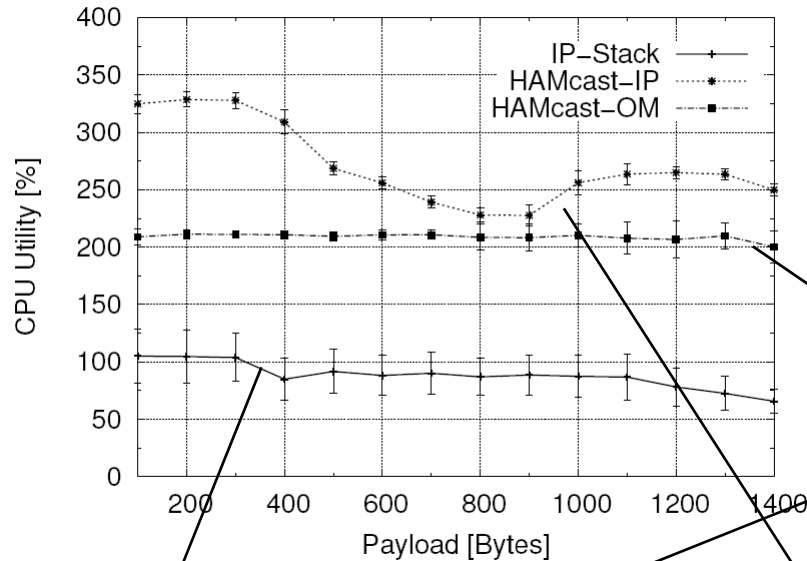


Packet Loss

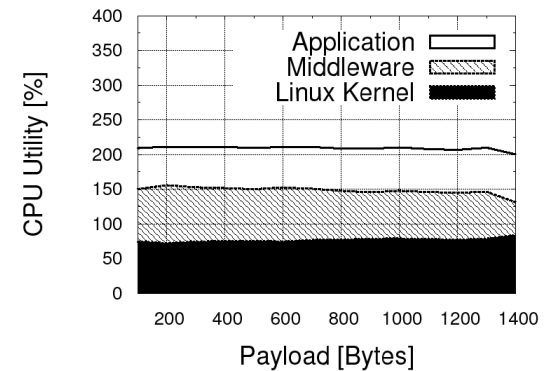
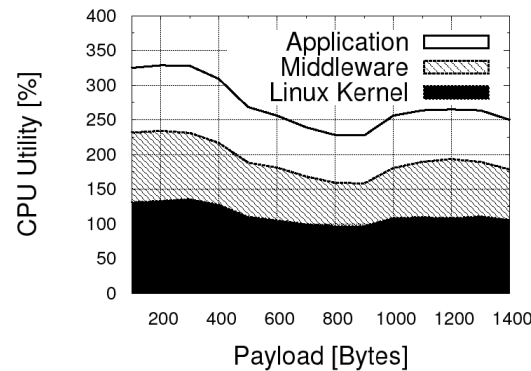
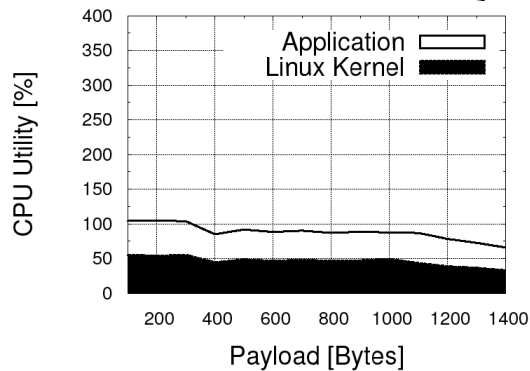
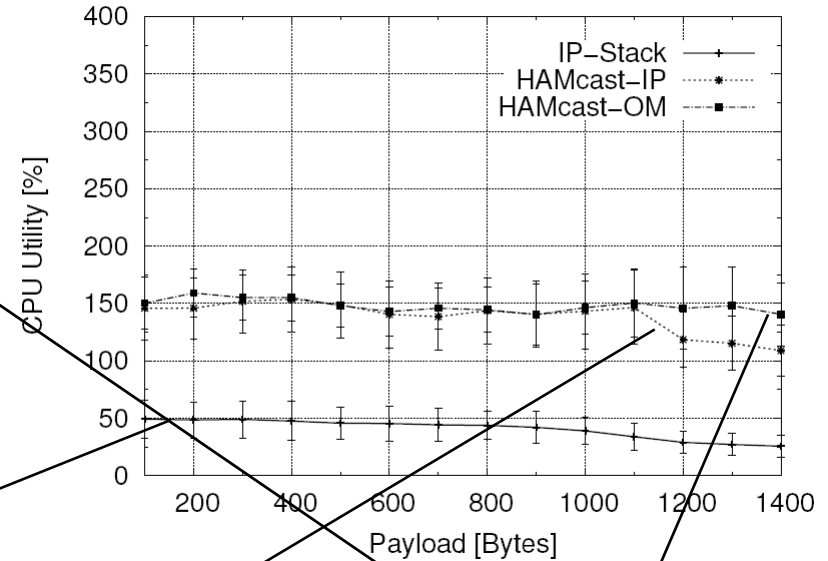


CPU Usage

Sender



Receiver



MINITUTORIAL
LET'S DO SOME PRACTICAL

Download & 3rd Party Dependencies

- o http://www.realmv6.org/hamcast_dev_downloads.html
- o Required tools and libs
 - build tools (gcc ...)
 - automake + libtool
 - Boost library (1.42 or higher)
 - SSL developer library
 - PCAP developer library
- o Almost standard libs ...
 - Packages are available for Ubuntu and Debian

Installation

- o Implementation consists of four parts
 1. Multicast API (`libhamcast/`)
 2. Middleware component (`middleware/`)
 3. Technology modules (`modules/<technology>`)
 4. Examples (`programs/<software>`)
- o Two alternatives:
 - `./build.sh` compiles the complete implementation
 - **Per subdirectory:** `automake -i && ./configure && make`
- o It's quite easy to install the HAMcast middleware 😊

Start the HAMcast Middleware

- o Create configuration file for the middleware

- `cp middleware/middleware.ini.example
middleware/middleware.ini`

- o Select modules of interest

- Linux *.so, MacOS *.dylib
- Example configuration →

- o Start the middleware

- `./run_hamcast.sh`

```
[global]  
log_level=error
```

```
[void_module]  
file=../modules/void/.libs/libvoid_m  
odule.dylib
```

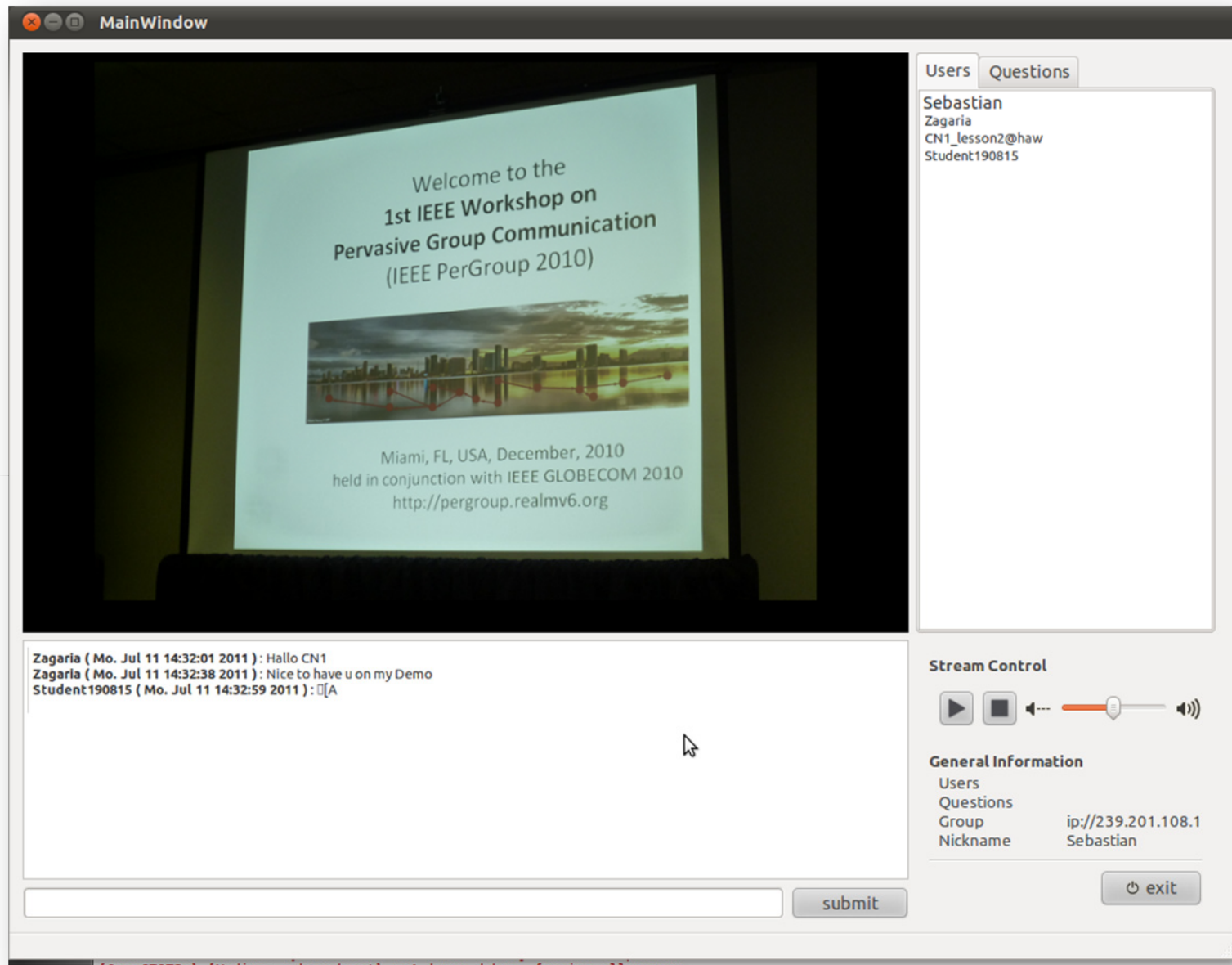
```
[loopback_module]  
file=../modules/loopback/.libs/liblo  
opback_module.dylib
```

```
[ip_module]  
file=../modules/ip/.libs/libipv4modu  
le.dylib
```

Code Example 1: Join & Receive

```
hamcast::uri group;
group = "ip://239.0.0.1:1234";
hamcast::multicast_socket s;
s.join(group);
hamcast::multicast_packet mp;
while(true) {
    if (s.try_receive(mp, 50)) {
        std::string tmp;
        const char* msg = reinterpret_cast<const
char*>(mp.data());
        std::copy(msg, msg + mp.size(),
            std::back_inserter_iterator<std::string>(tmp));
        tmp += '\n';
        cout << tmp;
        cout.flush();
    }
}
```

Application Example: Video Lecture



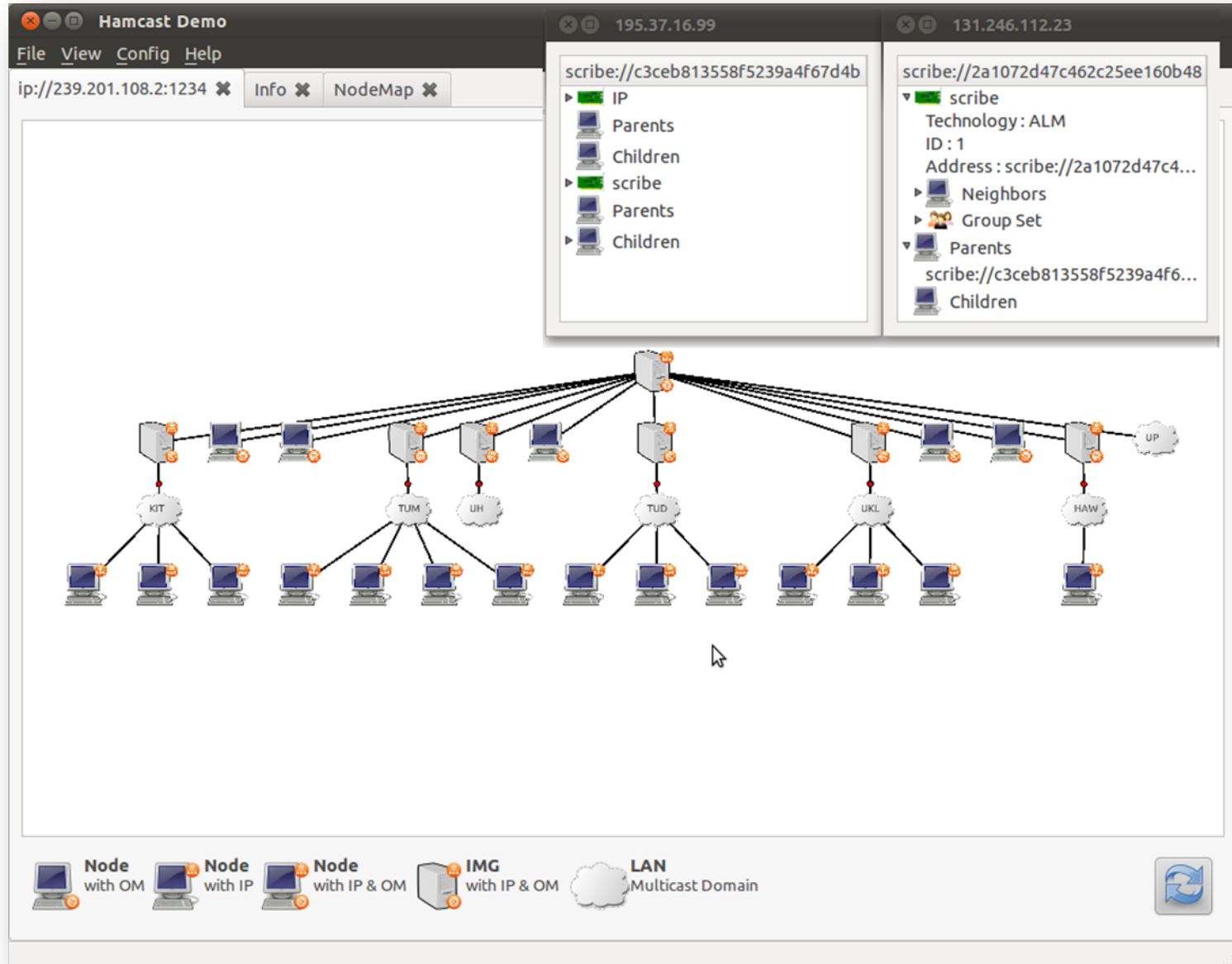
Code Example 2: Service Calls

```
#include <limits>
#include <iostream>
#include "hamcast/hamcast.hpp"
#include "hamcast/ipc.hpp"
#include <boost/thread.hpp>

using std::cout;
using std::endl;

int main(){
    std::vector<hamcast::ipc::interface_property> vec;
    vec = hamcast::ipc::get_interfaces();
    cout << "list of middleware interfaces:" << endl << endl;
    for (size_t i = 0; i < vec.size(); ++i) {
        cout << "interface[id = " << vec[i].id << "]: " << endl
             << "\tname = " << vec[i].name << endl
             << "\taddress = " << vec[i].address << endl
             << "\ttechnology = " << vec[i].technology << endl << endl;
    }
    return 0;
}
```

Application Example: Monitoring



Future Release

- o Passive service discovery
 - Based on kernel tables
- o Scribe module
 - Based on Chimera implementation
 - Public bootstrap node hosted @ HAW Hamburg
- o More example applications included
 - Multicast monitoring & chat will be published
- o Improved installation script

Conclusions & Outlook

- Improved performance for HAMcast middleware
- Middleware is easy to install and easy to use
- New distribution technologies available soon

Download, Install, and have Fun!

→ <http://www.realmv6.org/hamcast/hamcast.tar.gz>