

The logo for RTROS. It features a 3x3 grid of dots on the left, with the top-left dot missing. The dots are dark blue, except for the middle-left dot which is red. To the right of the dots, the letters 'RTROS' are written in a bold, sans-serif font. 'RT' is red, and 'ROS' is dark blue.

Real-Time Extension to the Robot Operating System

Jan Carstensen
Axel Rauschenberger

This work has been supported by the Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw).

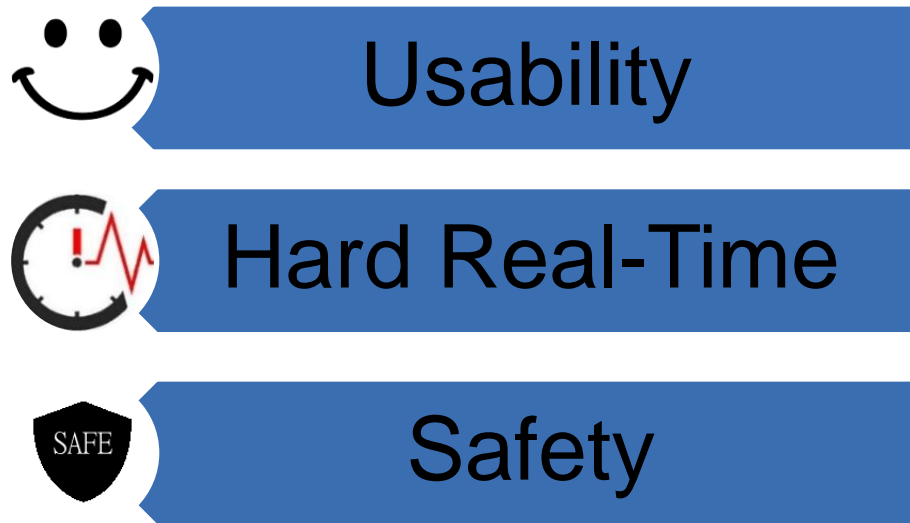
Motivation and Overview

- Introduction
- Architecture
- Communication
 - Publish/Subscribe
 - Services
- Real-Time Debugging
- Example Code
- Current Work

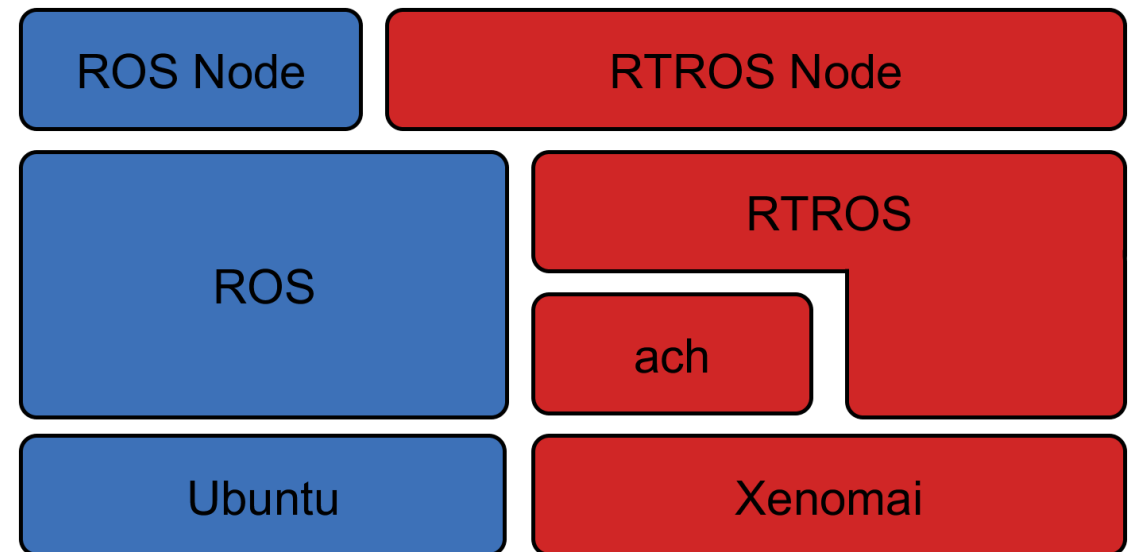


RTROS – Goals and Architecture

Design Goals

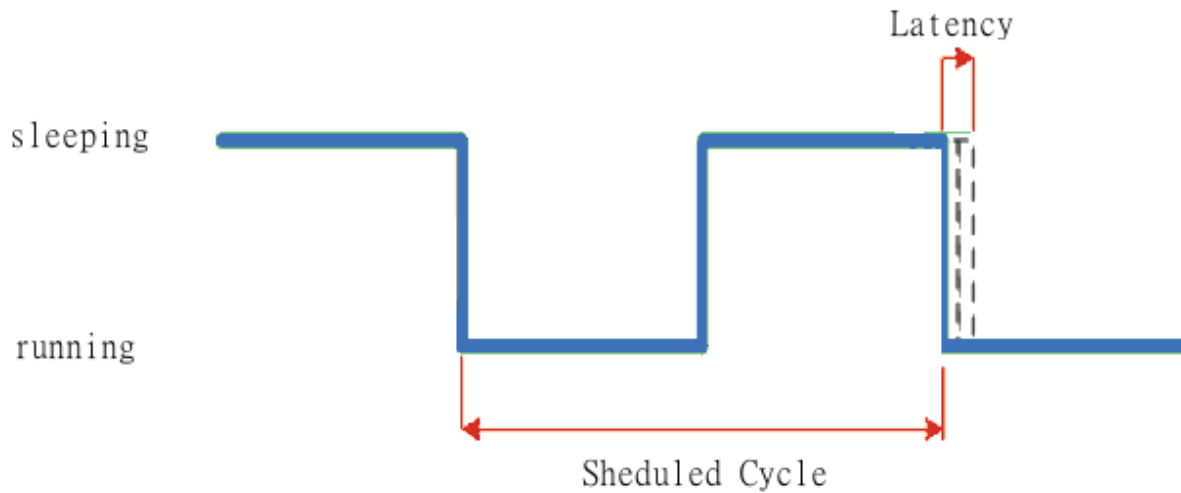


Architecture

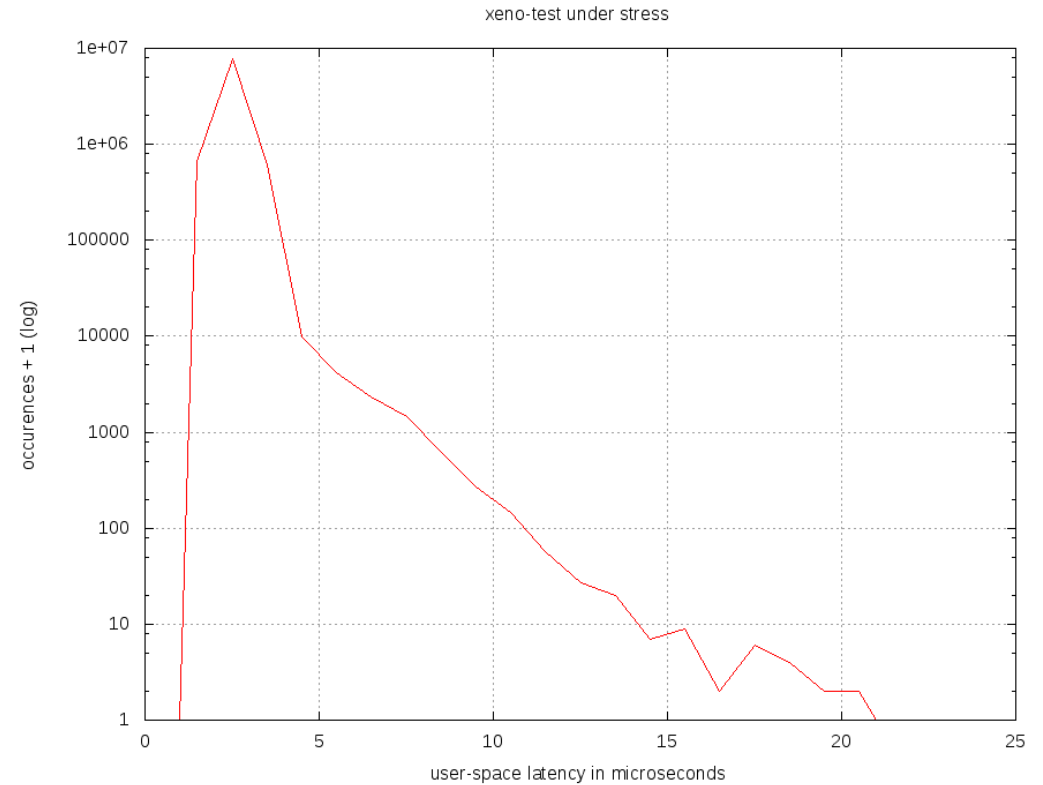


Rate – Cyclic Behaviour

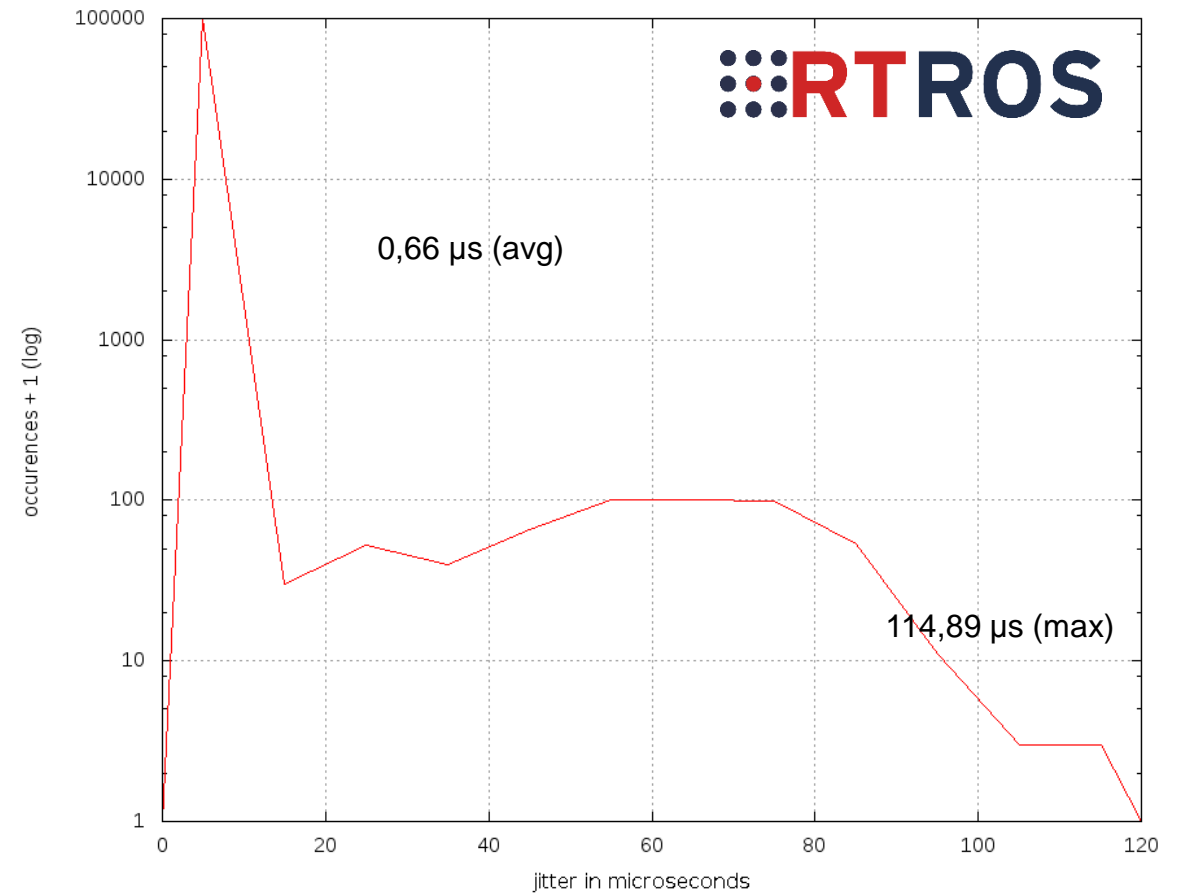
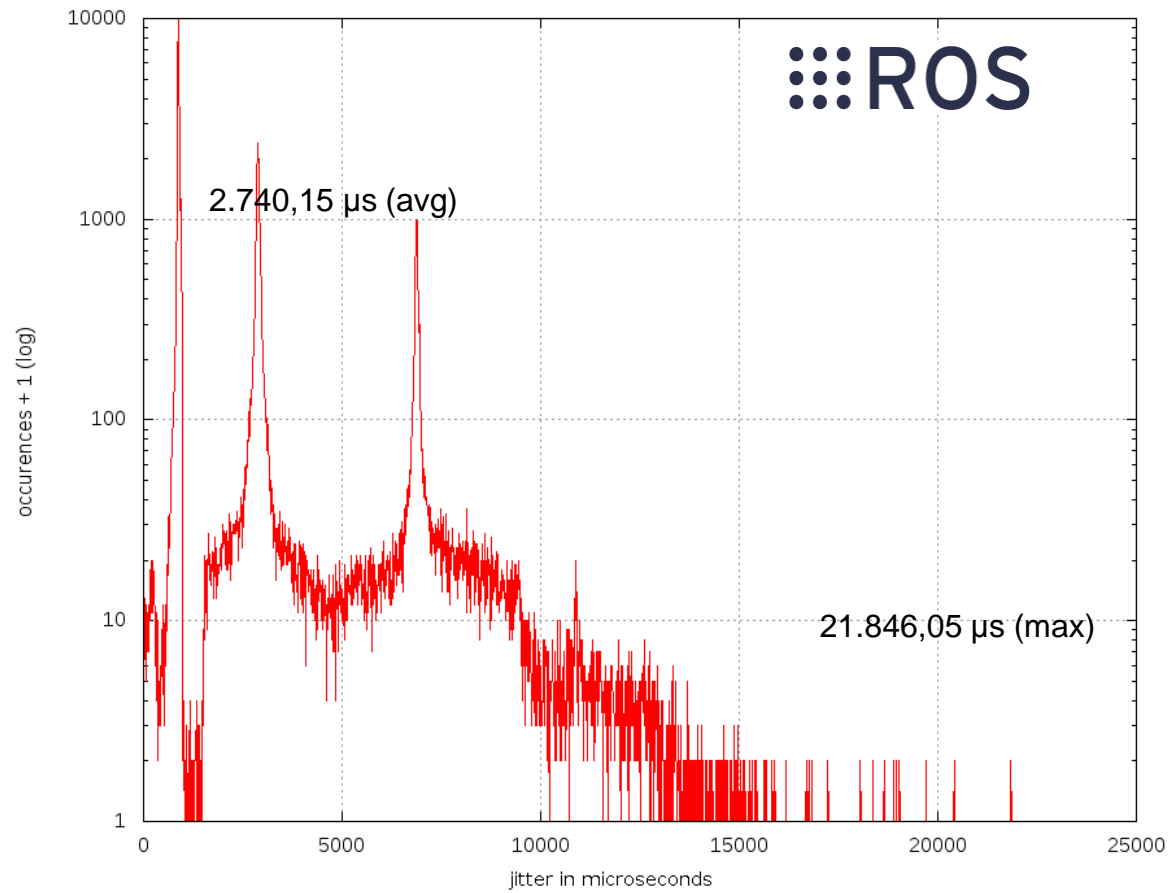
- Cyclictest



- System Latency

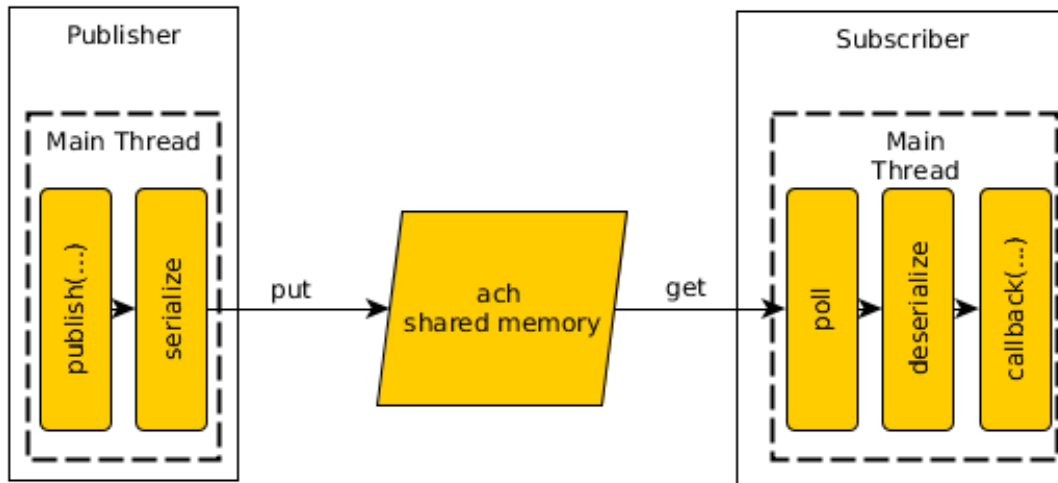


Rate Benchmark

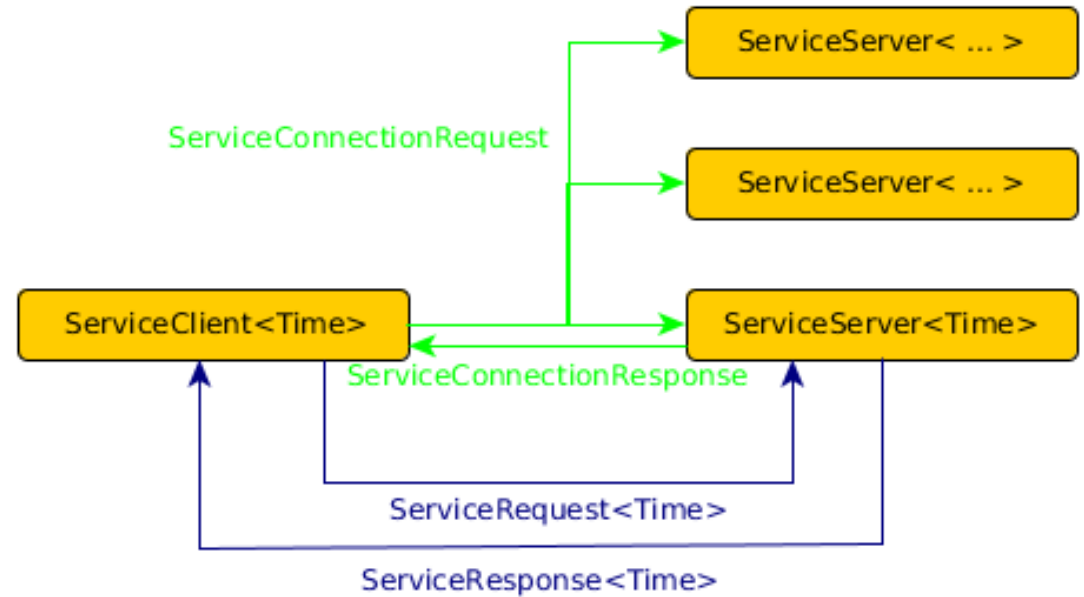


Communication

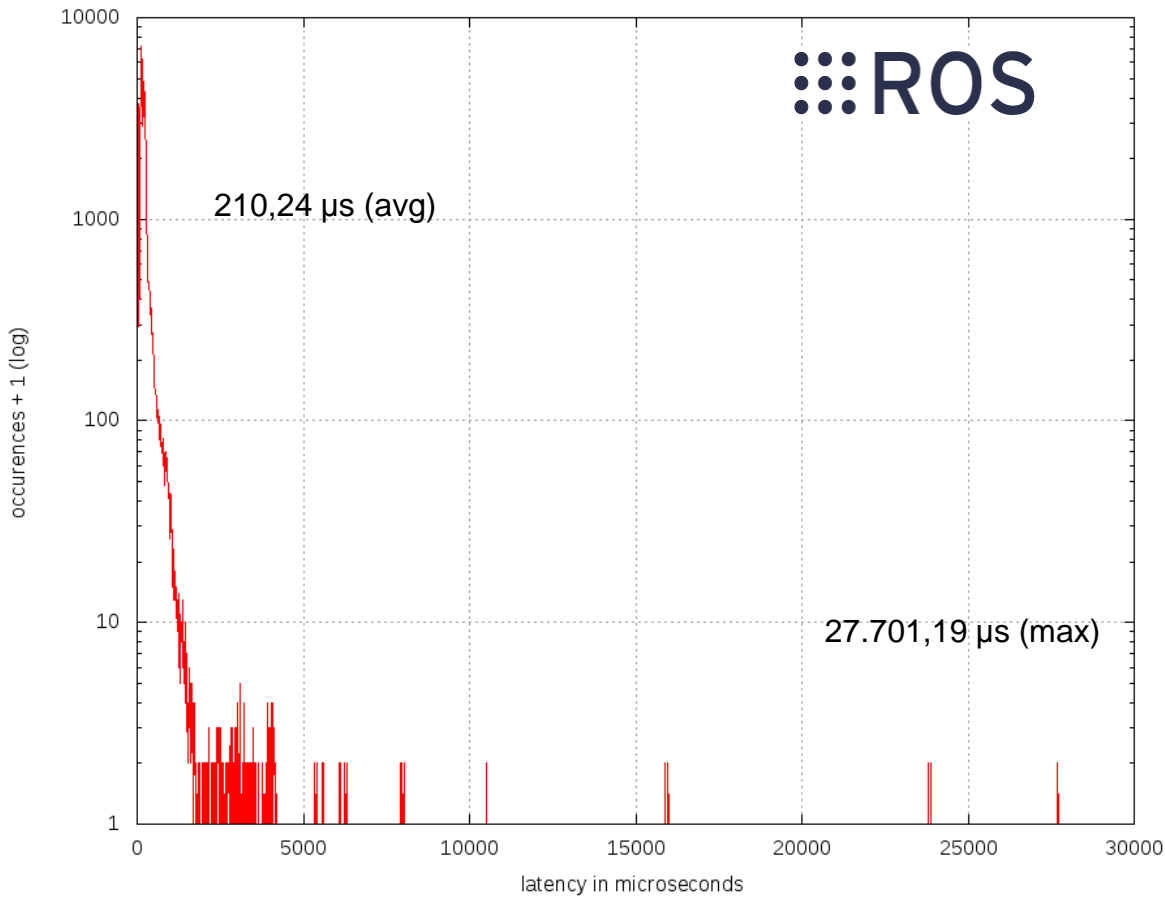
Publish/Subscribe



Services

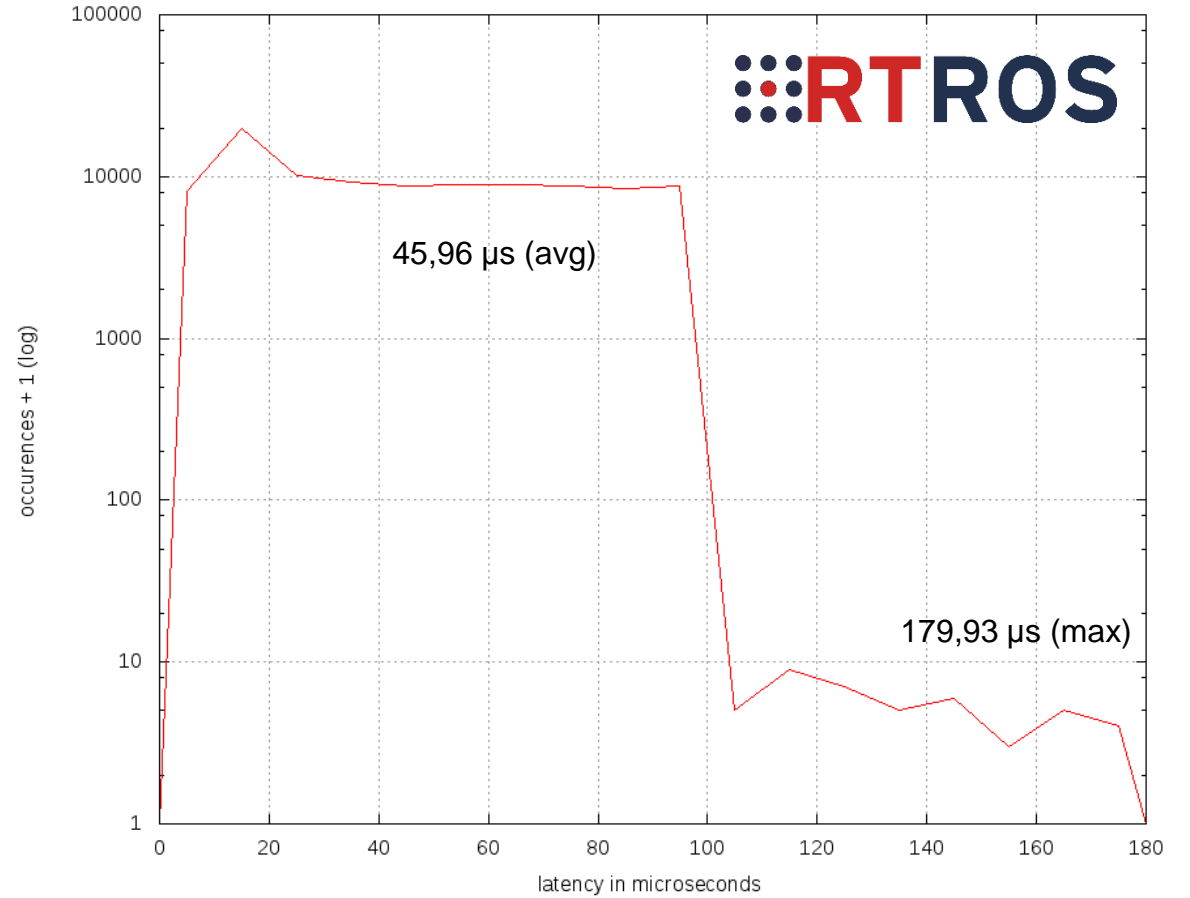


Publish/Subscribe Benchmark



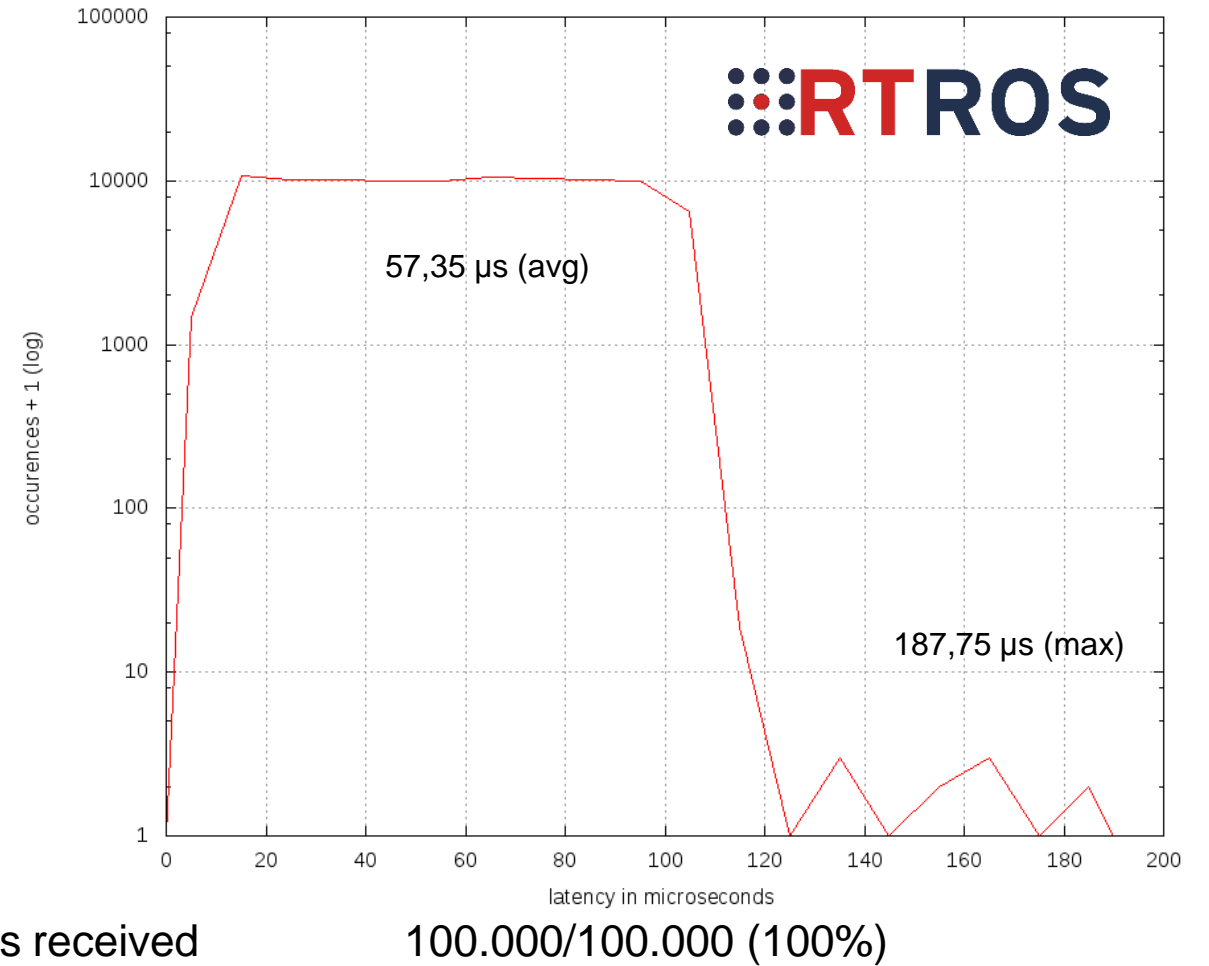
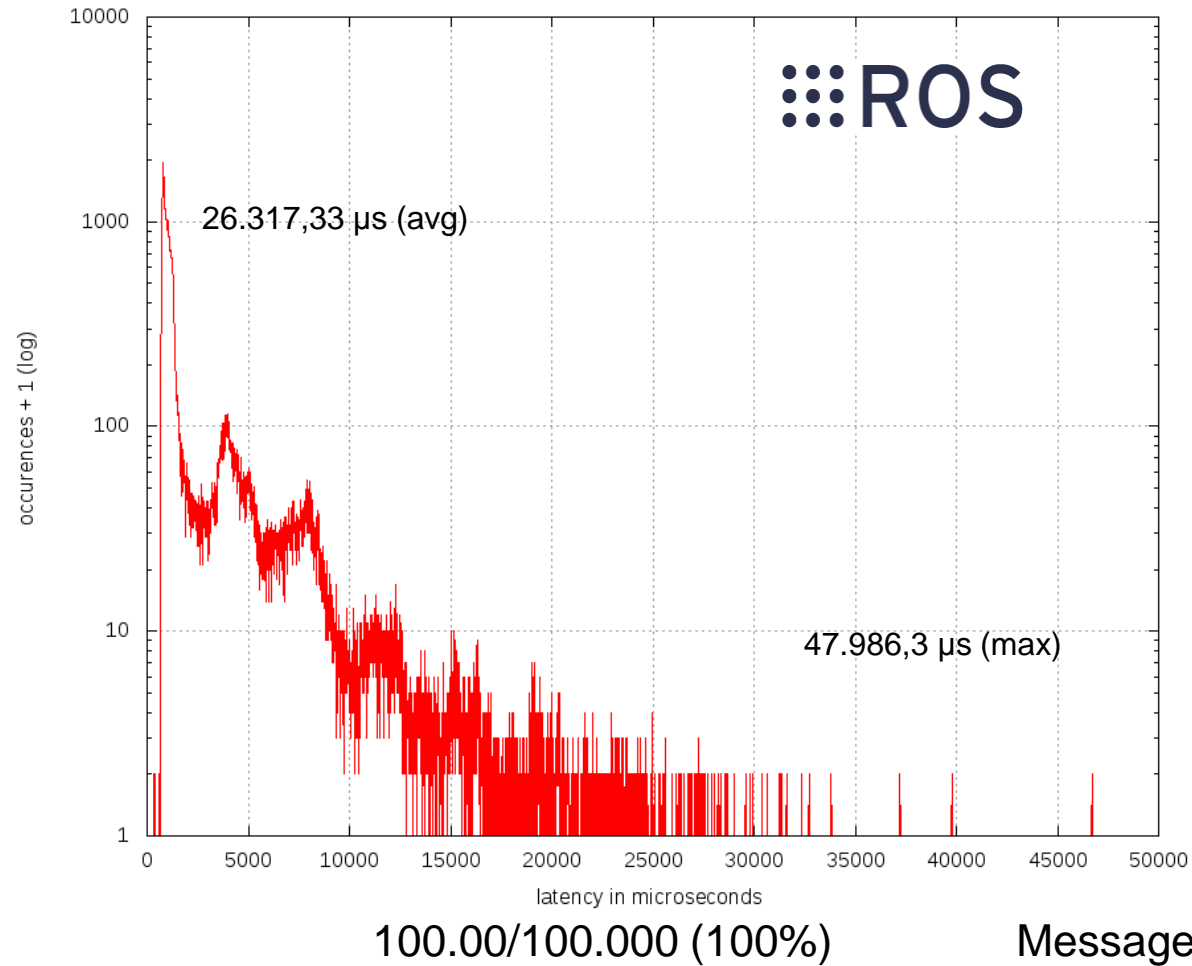
99.843/100.000 (99%)

Messages received



100.000/100.000 (100%)

Services Benchmark



Real-Time Debugging

```
carstensen@carstensen-xeno-14: ~  
[ RT-INFO] [1457968417.799695] The default way to output some Information!  
[RT-DEBUG] [1457968417.799701] Information vital to debug  
[ RT-WARN] [1457968417.799703] Something is not going well, maybe this is worth a look!  
[RT-ERROR] [1457968417.799704] Something is wrong!  
[ RT-WARN] [1457968417.799864] Mode switch (reason: invoked syscall). Backtrace:  
/home/carstensen/catkin_ws/devel/lib/librtros.so(_ZN5rtros16sigdebug_handlerEiP9siginfo_t  
/lib/x86_64-linux-gnu/libc.so.6(+0x36d40)[0x7fdbe94cfd40]  
/lib/x86_64-linux-gnu/libc.so.6(__write+0x2d)[0x7fdbe958488d]  
/opt/ros/indigo/lib/libroscpp.so(_ZN3ros7PollSet6signalEv+0x5a)[0x7fdbeae0136a]  
/opt/ros/indigo/lib/libroscpp.so(_ZN3ros12TopicManager7publishERKSsRKN5boost8functionIFNS  
0x234)[0x7fdbead9c614]  
/home/carstensen/catkin_ws/devel/lib/rtros_test/modeSwitch_talker(_ZNK3ros9Publisher7publ  
vRKT_+0xb4c)[0x40f12e]  
/home/carstensen/catkin_ws/devel/lib/rtros_test/modeSwitch_talker(main+0x295)[0x40c5a2]  
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf5)[0x7fdbe94baec5]  
/home/carstensen/catkin_ws/devel/lib/rtros_test/modeSwitch_talker[0x40c249]
```

Example Code



```
#include "ros/ros.h"
#include "std_msgs/Time.h"

int main(int argc, char **argv)
{
    ros::init(argc, argv, "ros_talker");
    ros::NodeHandle nh;
    ros::Publisher pub_rts = nh.advertise<std_msgs::Time>("time", 10);

    std_msgs::Time msg;
    ros::Rate loop_rate(1000);
    ros::Time rosTime;

    while (ros::ok())
    {
        rosTime = ros::Time::now();
        msg.data = rosTime;
        pub_rts.publish(msg);

        ros::spinOnce();
        loop_rate.sleep();
    }

    return 0;
}
```

```
#include "rtros/rtros.h"
#include "std_msgs/Time.h"

int main(int argc, char **argv)
{
    rtros::init(argc, argv, "rtros_talker");
    rtros::NodeHandle nh;
    rtros::Publisher pub_rts = nh.advertise<std_msgs::Time>("time", 10);

    std_msgs::Time msg;
    rtros::Rate loop_rate(1000);
    ros::Time rosTime;

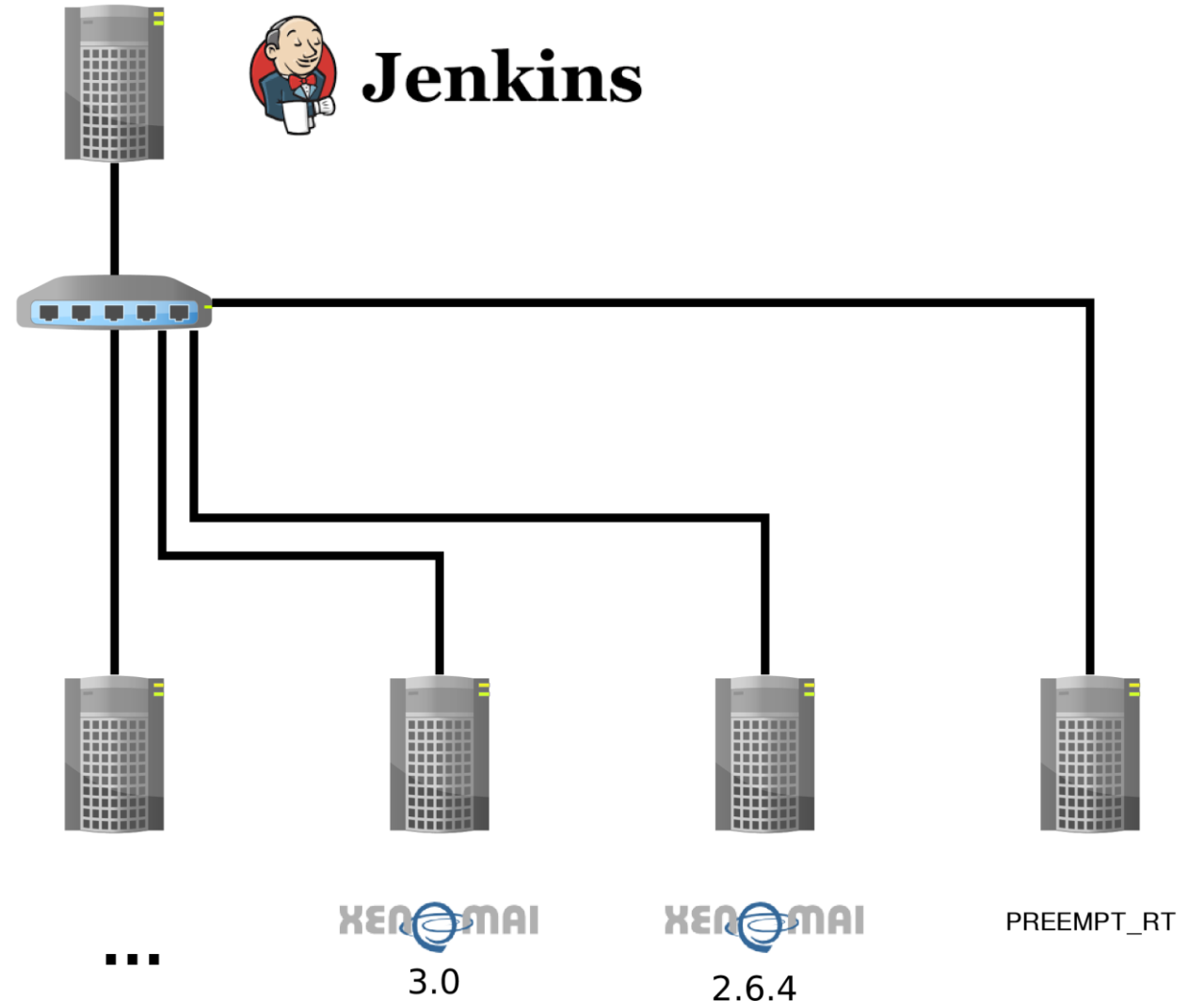
    while (rtros::ok())
    {
        rosTime = rtros::Time::now();
        msg.data = rosTime;
        pub_rts.publish(msg);

        rtros::spinOnce();
        loop_rate.sleep();
    }

    return 0;
}
```

Current Work

- Testing and Improving
- RTROS – Compatibility
 - Preempt_RT
 - Xenomai 3.0
- External Evaluation
 - Open-Source Release
- New Features
 - Network Communication
 - Time Synchronisation



Conclusion

Features

- Publish/Subscribe
- Services
- Time
- Transform (TF)
- Console

Problems

- Drivers
- Real-Time Restriction
- Sensors

