# Robotics-as-a-Service Benchmarks over ROS2

## GOALS

Robotics-as-a-Service (RaaS) platforms can significantly enhance the value of Autonomous Mobile Robots (AMRs).

We designed and implemented a system Quality of Service (sQoS) tool to benchmark our RaaS architecture, ROS2 middlewares (RMW) and ROS2 QoS settings performance using Fed4Fire+ testbeds with our AMR prototype.

The Fed4Fire+ VirtualWall2, Grid'5000, Emulab Utah and ExoGENI testbeds were used for the experiments.

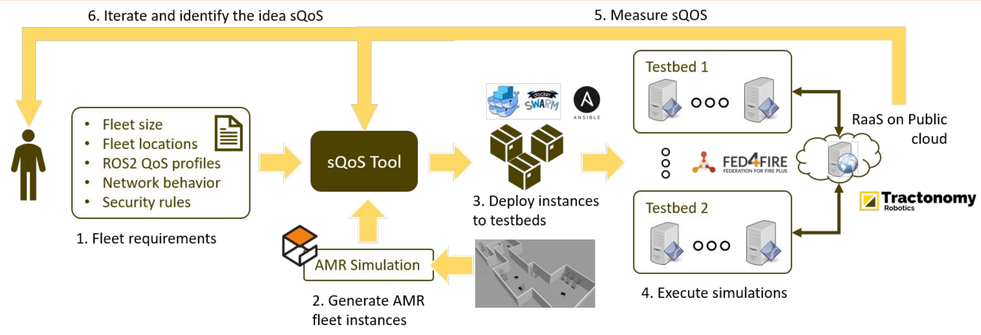
## CHALLENGES

To connect to the cloud with ROS2 we need to route multicast and unicast traffic in both directions over the WireGuard VPN tunnel

Limited information available on network performance of ROS2 nodes working in a distributed setup. Validating, benchmarking and testing multiple configurations on a distributed setup is expensive and hard to get right.

Before going into production we need performance metrics when scaling connected AMRs.

## DEMO SETUP



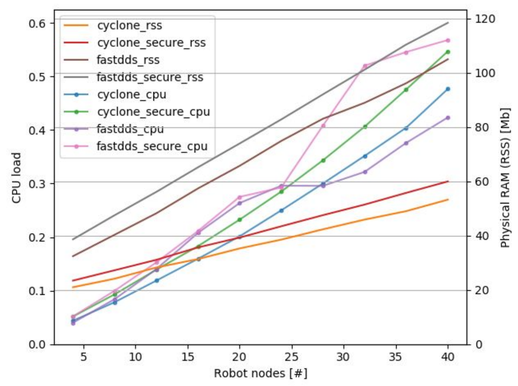

A 6 bare-metal nodes setup was configured on the VirtualWall2 infrastructure (Cloud, Manager, and 4 workers).

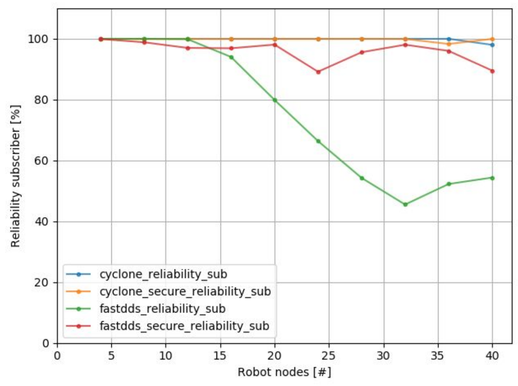
WireGuard VPN tunnel between Cloud and Manager which can be impaired to simulate Wi-Fi networks. All AMR traffic is routed through this tunnel.

Secondary sites or customers deployed on other testbeds connecting to the Cloud node on the main setup.

Using Docker Swarm we can deploy upto 40 AMRs on the VirtualWall2 testbed.

## RESULTS

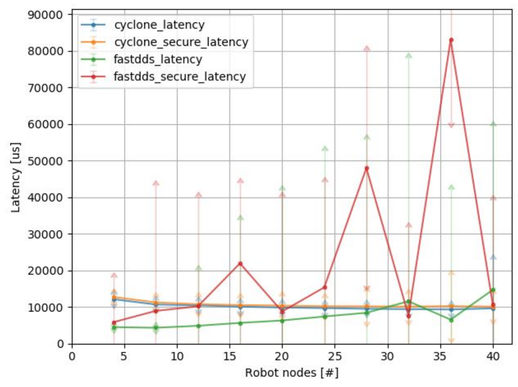



Performance metrics of RMW middlewares while scaling the number of connected AMRs.

Reusable sQoS tool to test next ROS2 versions, different QoS settings and new RMWs.

Performance degrades when many ROS2 nodes are used.

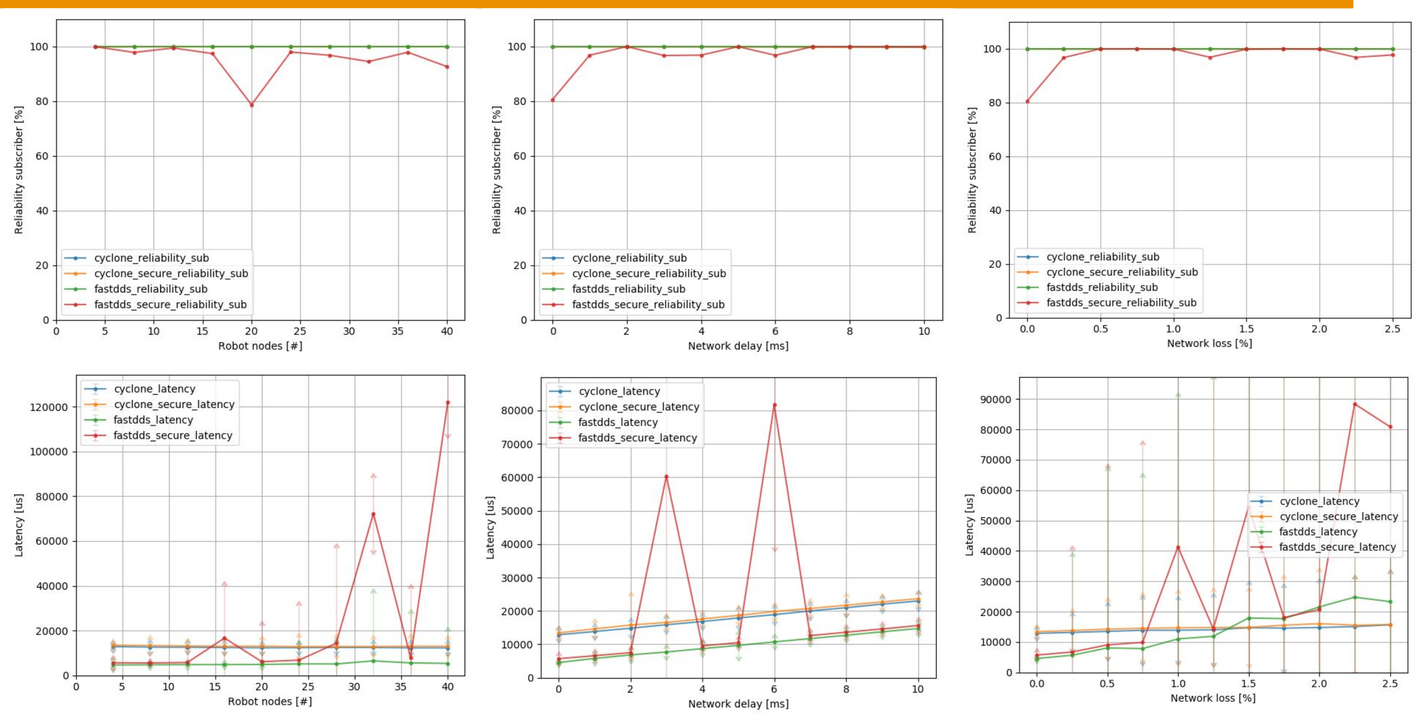



## MORE RESULTS

Graphs represent ROS2 subscribers for a subset (only heartbeat and diagnostics) of the AMRs ROS2 nodes: reliability (top row); and average latency (bottom row). First column shows the results when scaling AMRs on a normal network while the second and third column show the results for 4 AMRs on impaired networks (loss/delay).

Default ROS2 RMW implementation, Fast DDS, shows unreliable behavior with high variance in the latency. Overhead of ROS2 security is neglectable.

Limiting the number of ROS2 nodes greatly affected the reliability of our connected AMRs and we can further increase the amount of connected AMRs.




## CONCLUSIONS

Validated scalability of the ROS2/VPN RaaS framework.

Resolved many challenges regarding distributed ROS2 setup over a WireGuard VPN tunnel. Created setup documentation to configure and setup a production ready environment.

Further validated ROS2 security features and gained more insights in ROS2 optimizations.

Discovered limitations of the default ROS2 RMW implementation which resulted in switching to Cyclond DDS.

System Quality of Service tool enables us to benchmark new configurations and changes to the platform, giving us a clear baseline metric to compare against.

## POST MORTEM

RaaS-o-ROS2 needs further optimization to limit meta traffic between ROS2 nodes, and optimize the on robot topics and services with shared memory to speed-up the performance.

Fed4Fire+ testbeds offer easy-to-setup and unrestricted test environments for scaling testing. It helped us gain the knowhow to setup a distributed ROS2 environment.

RaaS-o-ROS2 project has accelerated the development of the scalability of our RaaS platform.

Ready for the next major step in going to production: making the Omnit SAFE.