

# ROS2\_benchmarking 测试报告

## 1. 说明

此测试框对 ROS2 的通信特性快速且自动地在多个轴上进行评估，并且包含 ROS1 和 ROS2 的对比。

原地址: [https://github.com/piappl/ros2\\_benchmarking](https://github.com/piappl/ros2_benchmarking)

针对 ros2 Dashing 版本我们做了一些修改：

地址: [http://10.10.81.27:7990/projects/TEST/repos/ros2\\_benchmarking/browse](http://10.10.81.27:7990/projects/TEST/repos/ros2_benchmarking/browse)

模拟环境: **单机 IPC**

首先模拟本机上 docker 端口到端口下的流量限制(limit), 延迟(delay), 丢包(loss), 重复(duplication), 损坏(corruption), 乱序(reorder)现象, 然后在此基础上, 针对每一种情况再从 firstReceived, latency(延时), lostPackets(丢包), throughput(吞吐量) 这四个轴进行评估。

然后我们分别选择 ros1, ros2:fastrtps, ros2:opensplice, ros2:connext 作为测试环境。

我们将结果分为两组对比：

1. ros1 和 ros2:fastrtps(因为 fastrtps 为 ros2 默认的 DDS)
2. ros2:fastrtps, ros2:opensplice, ros2:connext

以下篇幅为生成的数据对比图结果。由于图片太多，我们先给出结论：

## 2. 结论

**第一部分 ros1 vs ros2:fastrtps**

### 1. 模拟流量限制: 从 100 – 1000kbit

	结论
firstReceived	无影响
latency	ros1 延时略低于 ros2
lostPackets	无影响
throughput	差别不大

## 2. 模拟延迟：从 0 – 1000ms

	结论
firstReceived	ros2 略优
latency	ros2 略优
lostPackets	ros2 无丢包情况， ros1 出现丢包情况
throughput	ros2 平稳， ros1 出现吞吐量下降情况

## 3. 模拟丢包：从 0 – 100%

	结论
firstReceived	ros2 优于 ros1
latency	ros2 优于 ros1
lostPackets	ros2 优于 ros1
throughput	ros2 优于 ros1

## 4. 模拟重复：从 0 – 100%

	结论
firstReceived	无影响
latency	ros2 优于 ros1
lostPackets	无影响
throughput	差别不大

## 5. 模拟损坏：从 0 – 100%

	结论
firstReceived	高损坏率 ros2 依然能收到，优于 ros1
latency	ros2 优于 ros1
lostPackets	ros2 优于 ros1
throughput	ros2 优于 ros1

## 6. 模拟乱序：从 0 – 100%

	结论
firstReceived	差别不大
latency	差别不大
lostPackets	无影响
throughput	差别不大

结论 1：

ros2(fastrtps)在 "性能" 上要优于 ros1.

## 第二部分 ros2:fastrtps vs ros2:opensplice vs ros2:connext

结论 2：

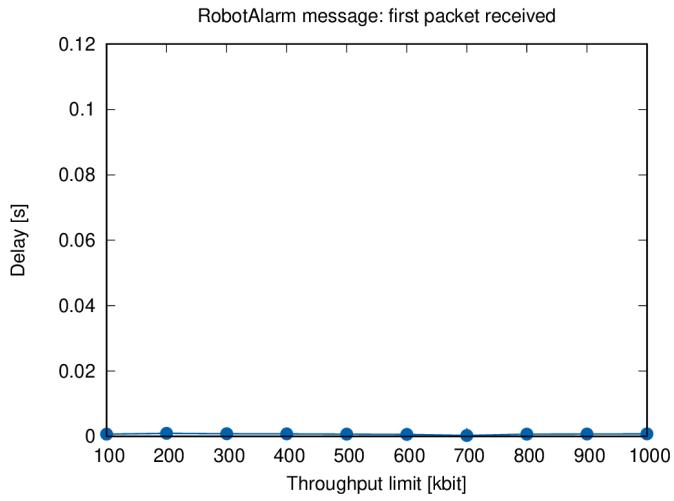
我们仅根据生成的图标结果确实难以给出一个一致性的结论，到底谁优谁劣。还需要针对不同需求，选择不同的 DDS.

针对 ROS2 and different DDS/RTPS vendors 官网有一个比较完整的说明：

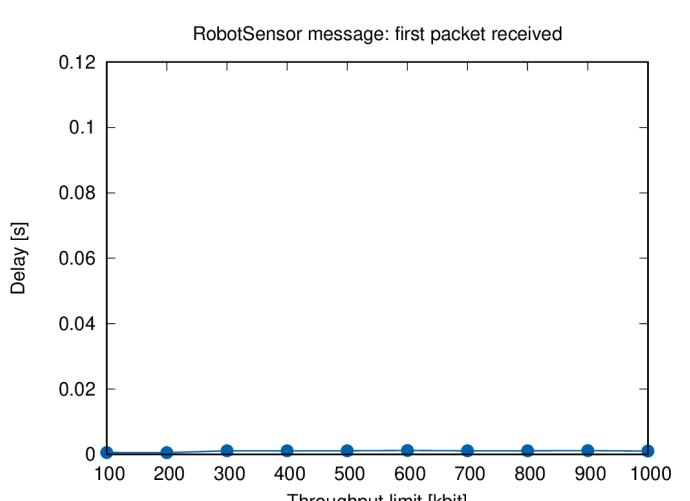
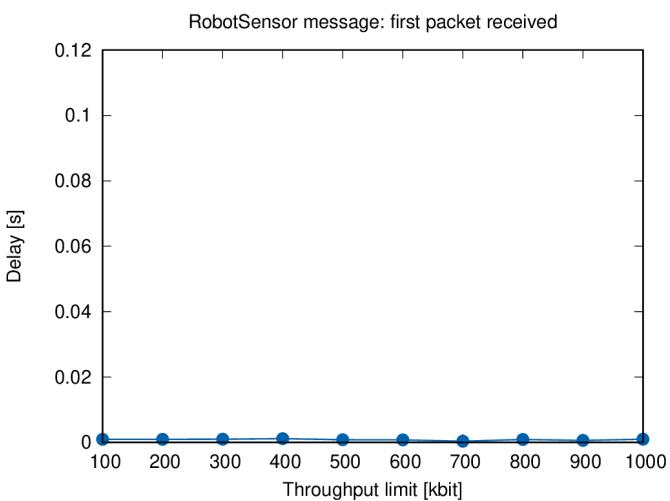
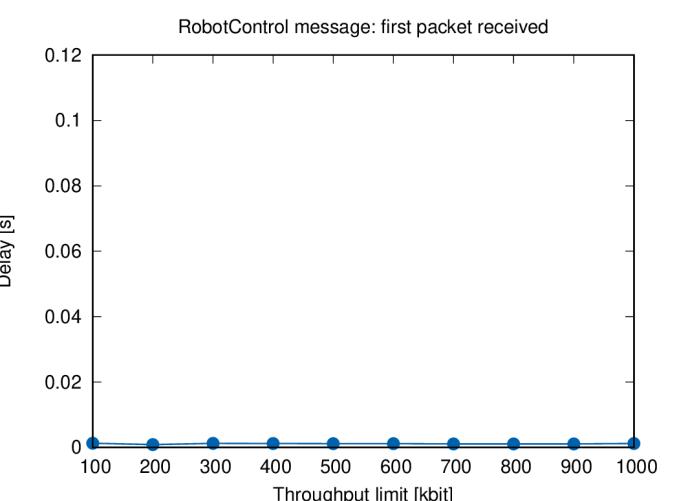
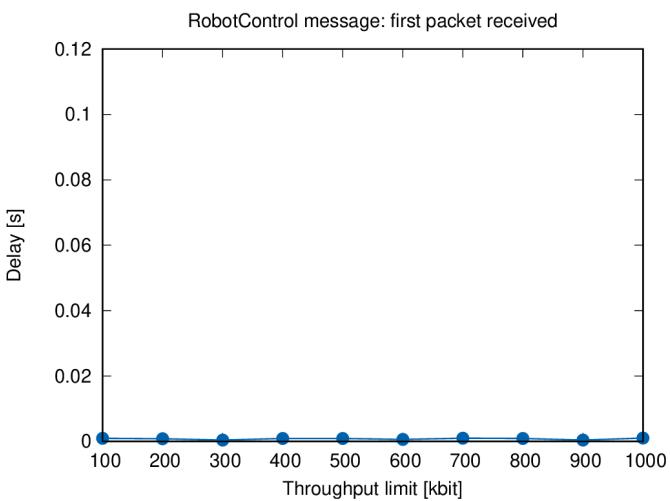
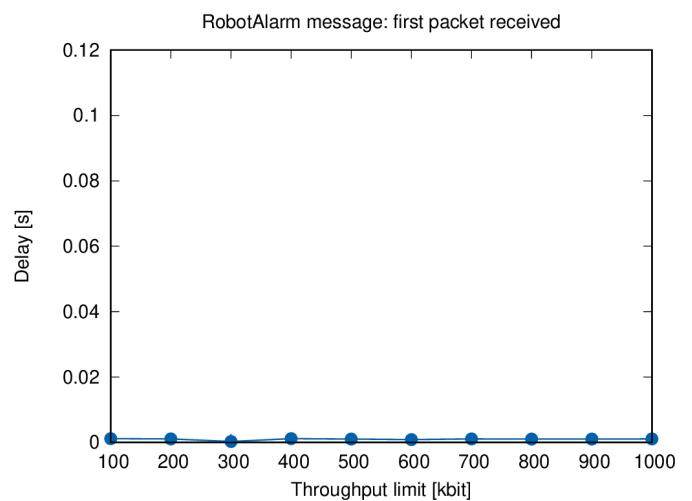
<https://index.ros.org/doc/ros2/Concepts/DDS-and-ROS-middleware-implementations/>

"ROS2 支持多种不同的 DDS/RTPS 实现，这是由于当它选择一个 vendor 来使用时，并不是完全一致的 (one size fits all)。当选择一个中间件实现时，有多种因素需要考虑，比如说后备考虑如 license，或者技术考虑如平台兼容性或计算规模。vendors 可能专注于不同的需求而提供多于一种的 DDS 或 RTPS 实现，比如说，RTI 就有多种不同的 Connex 实现，它们的目的各不相同，比如有些是针对特定处理器平台的，有些是满足不同应用的安全性验证的"

## 1. Limit: first-received ros1

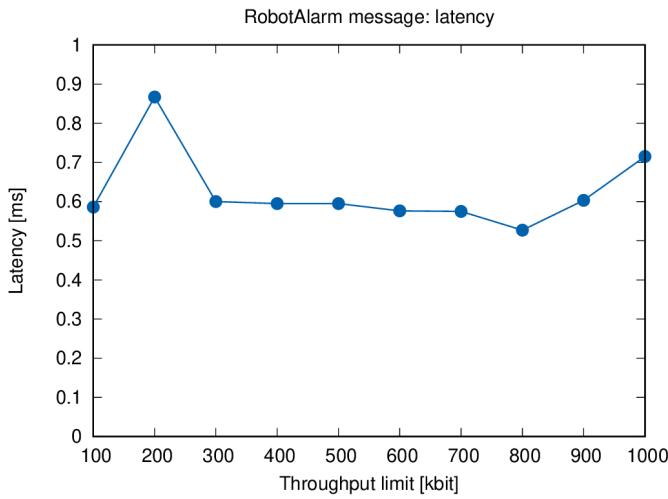


ros2:fastrtps

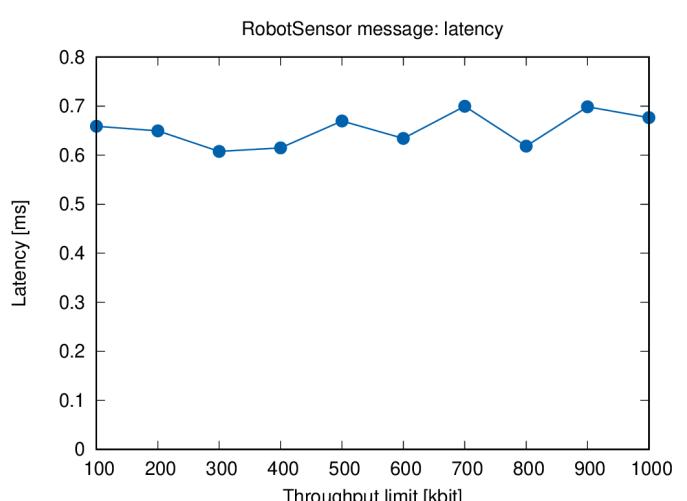
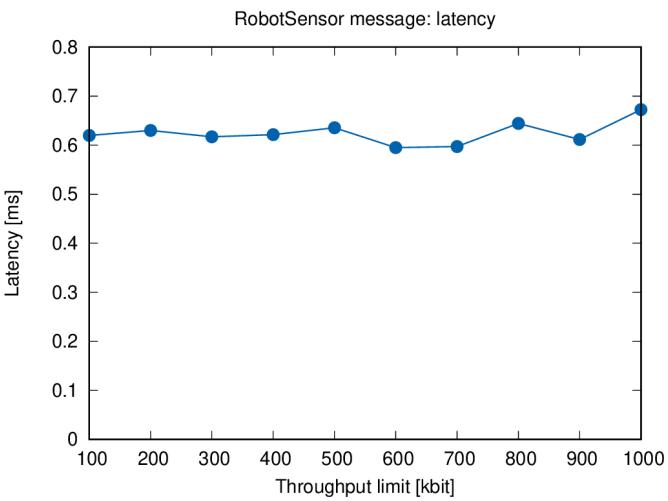
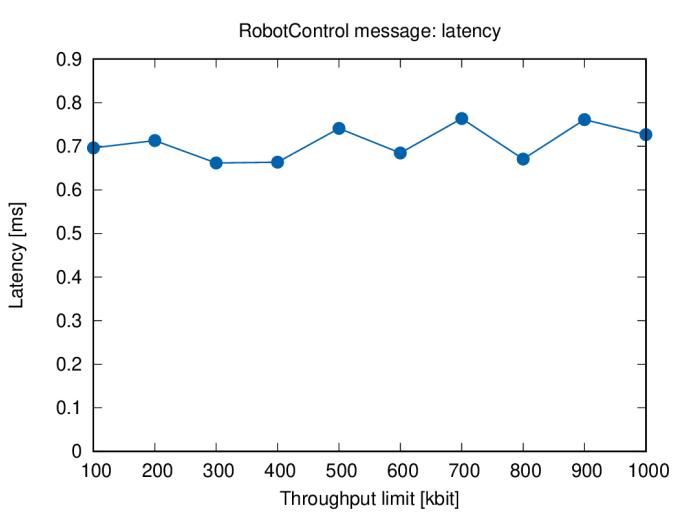
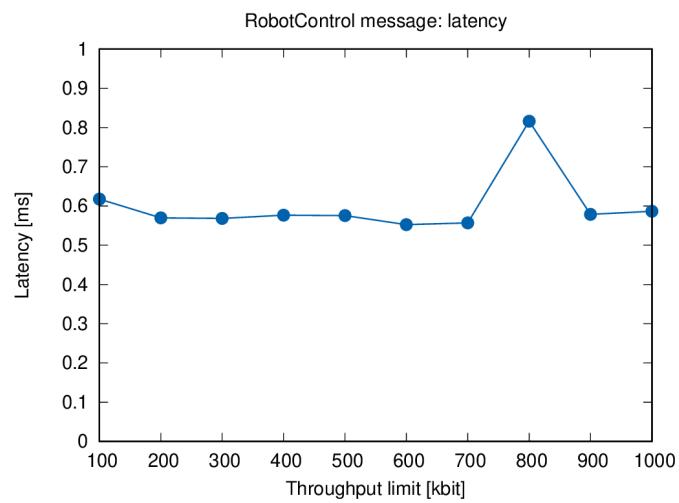
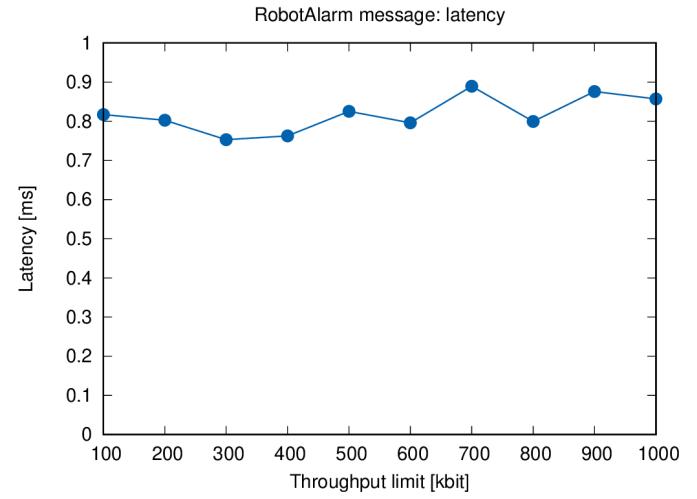


## 1. Limit: latency

ros1

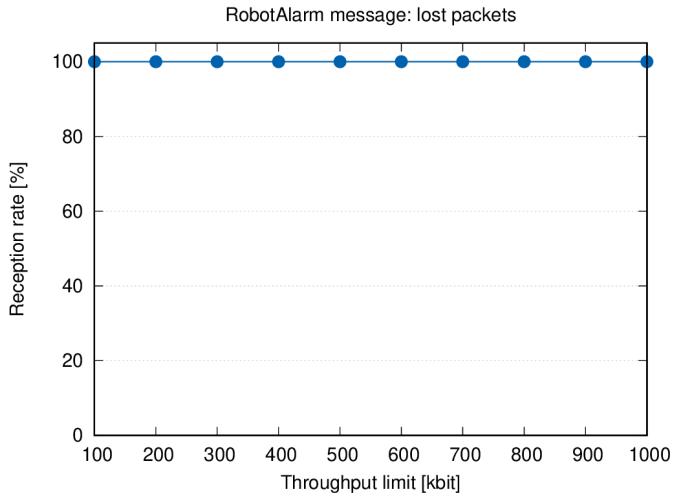


ros2:fastrtps

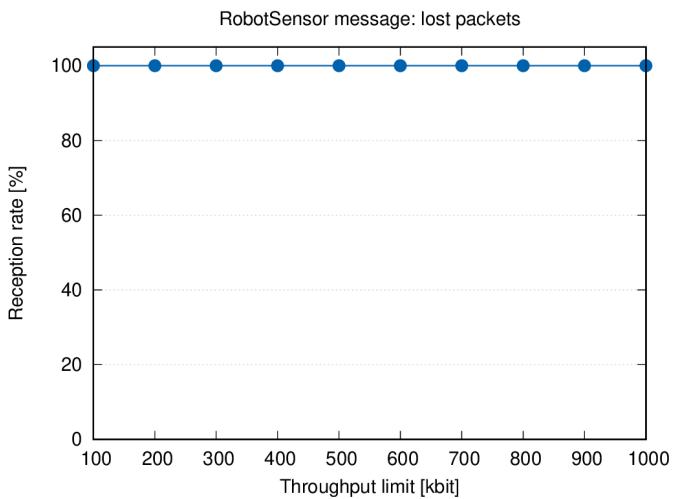
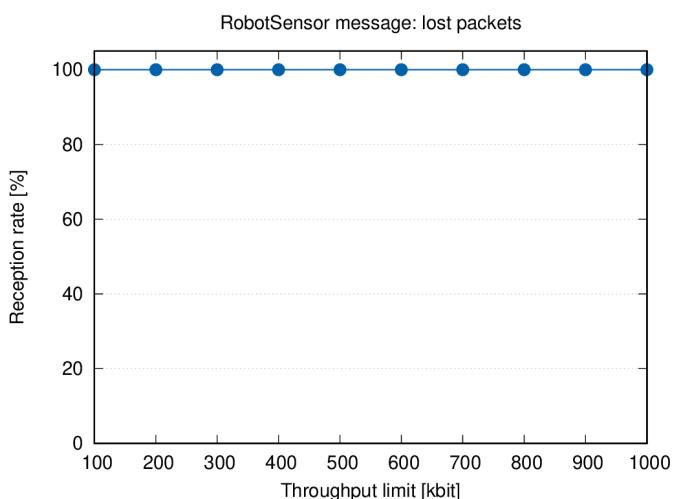
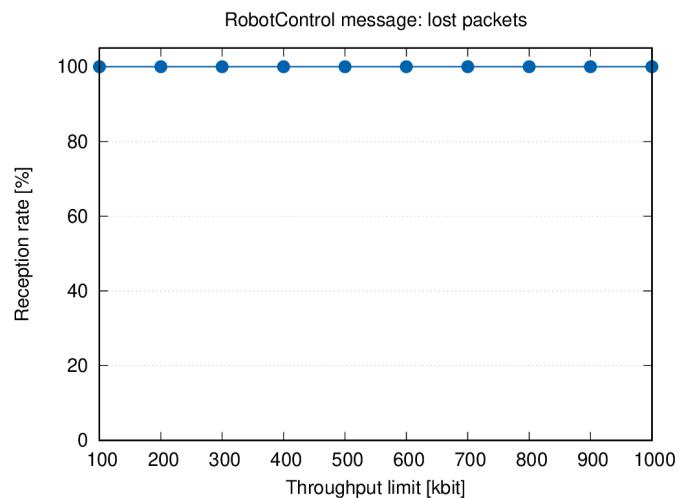
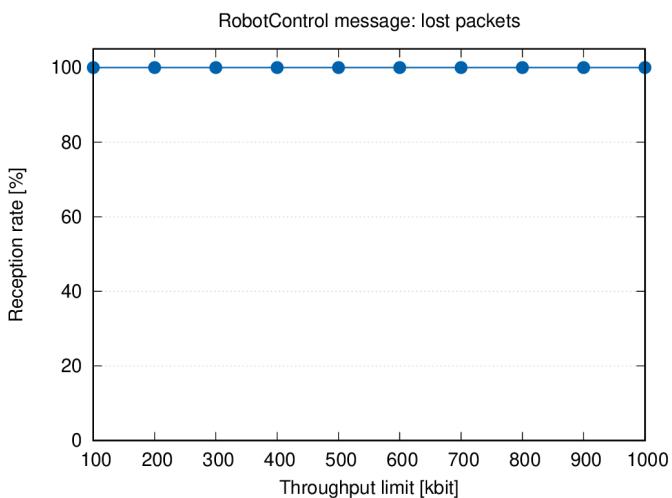
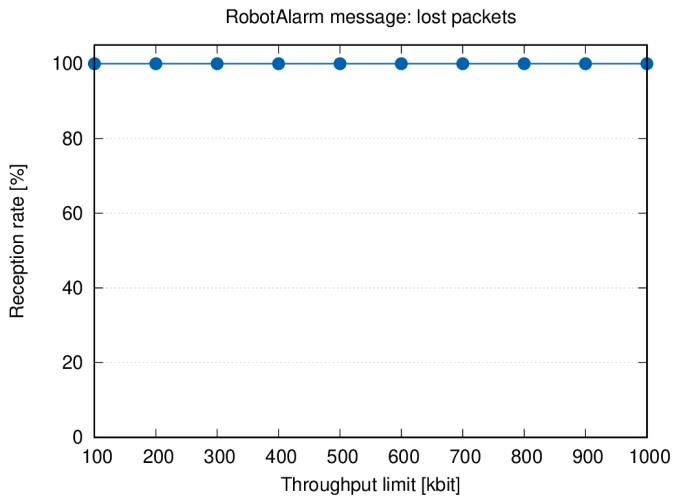


## 1. Limit:lost-packets

ros1

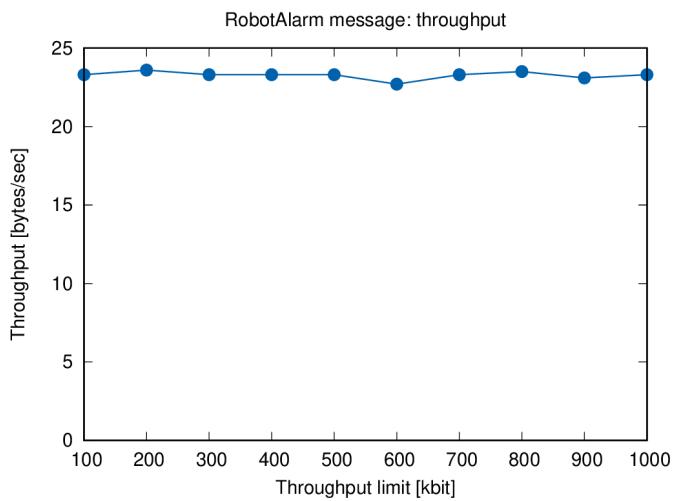


ros2:fastrtps

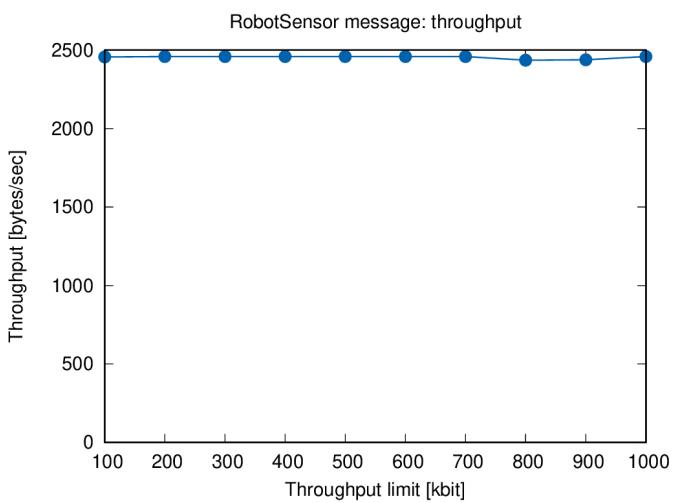
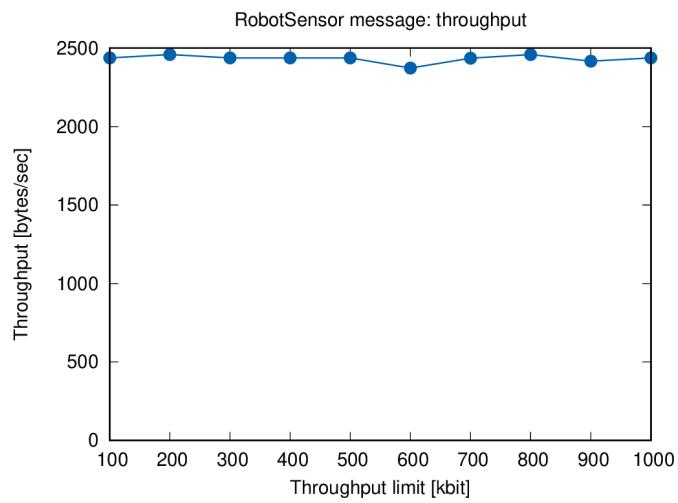
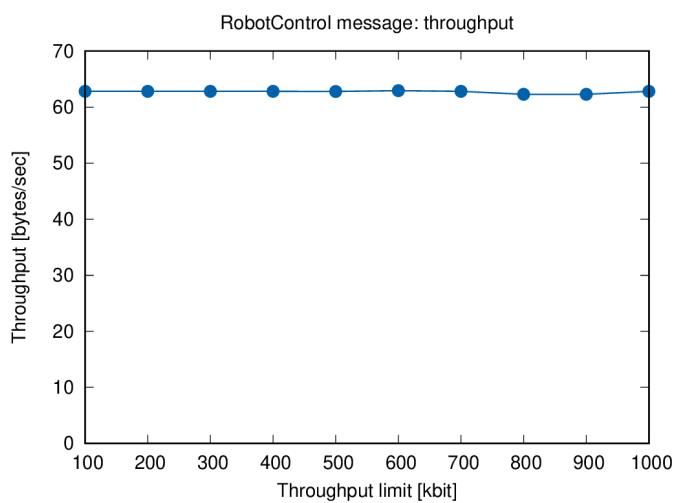
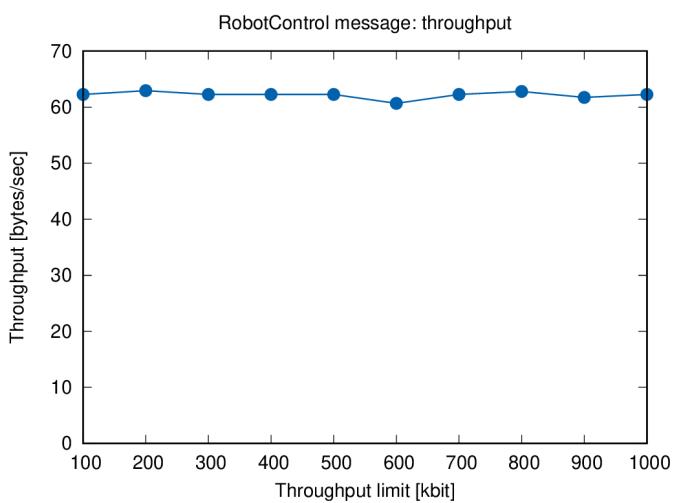
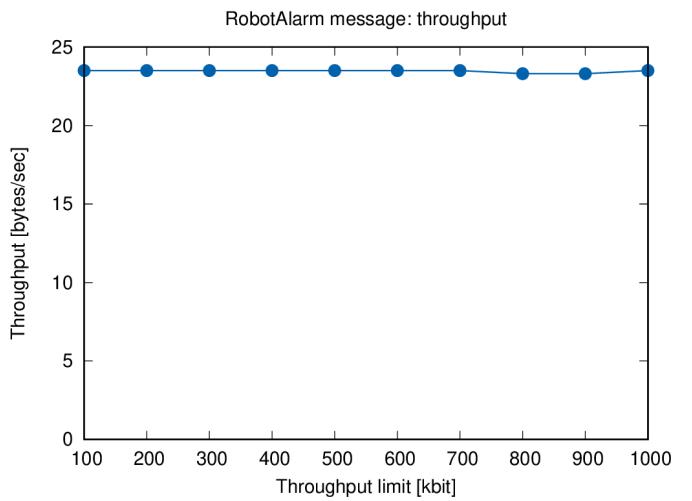


# 1.Limit:throughput

ros1

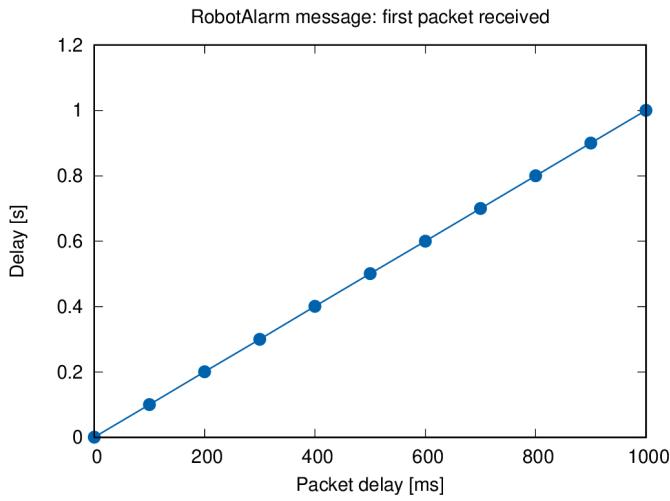


ros2:fastrtps

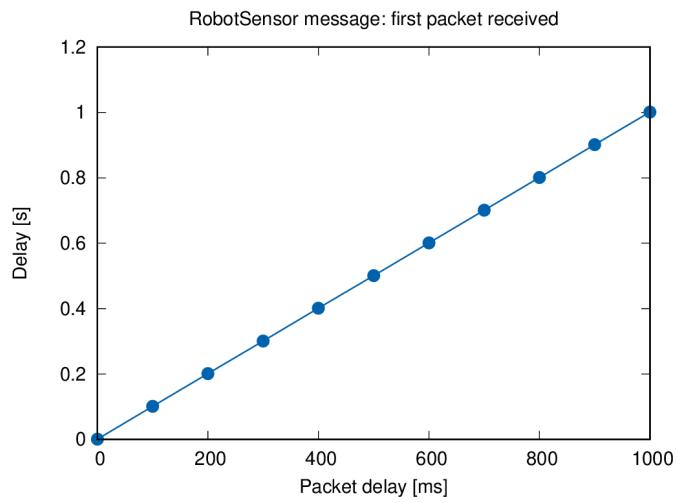
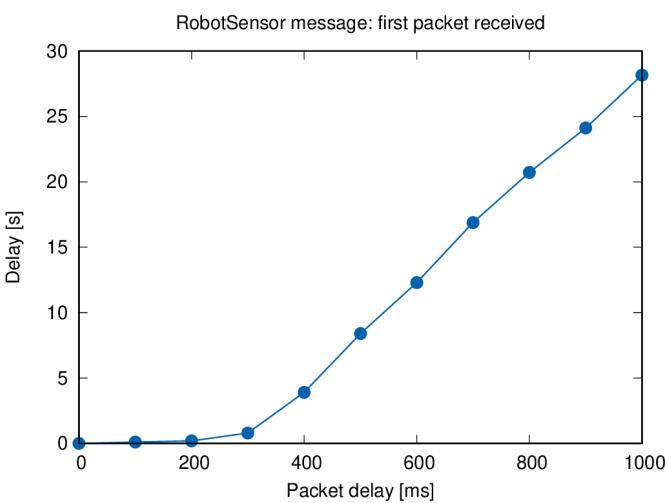
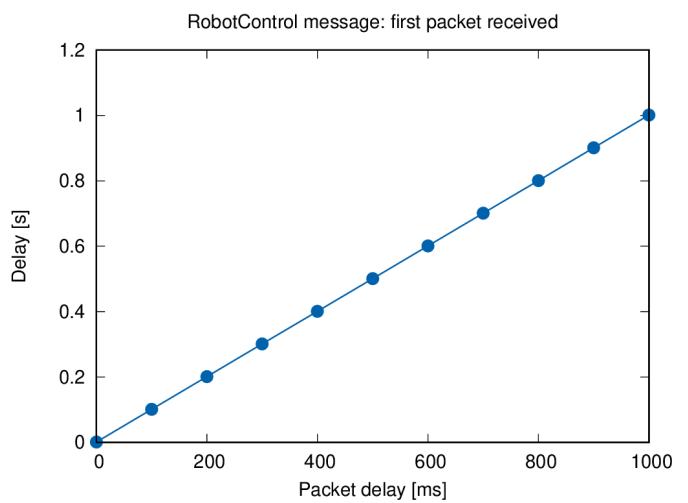
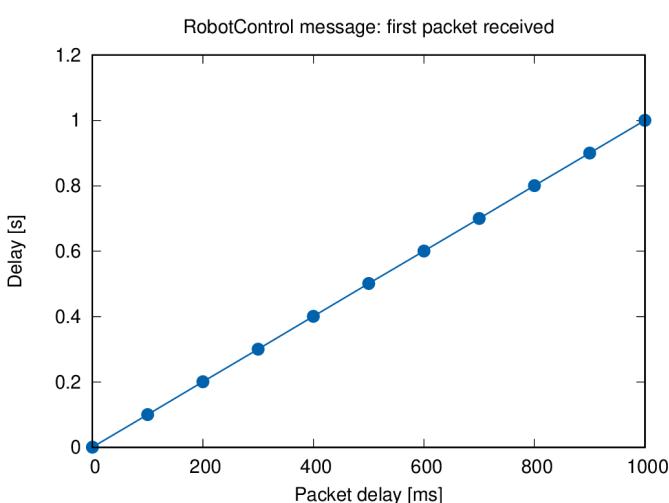
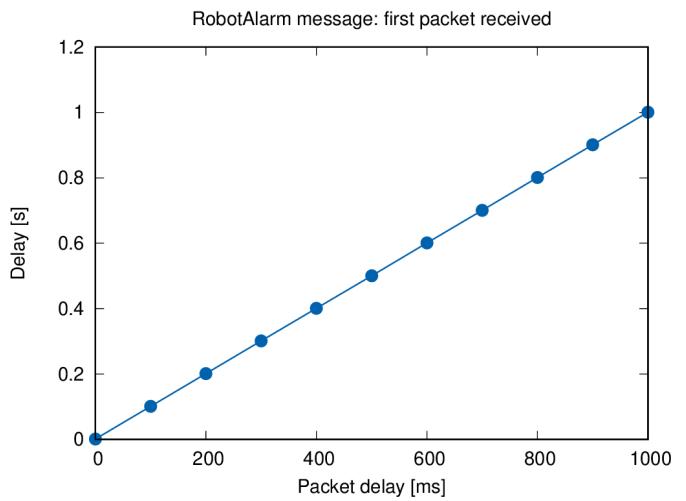


## 2. Delay:first-received

ros1

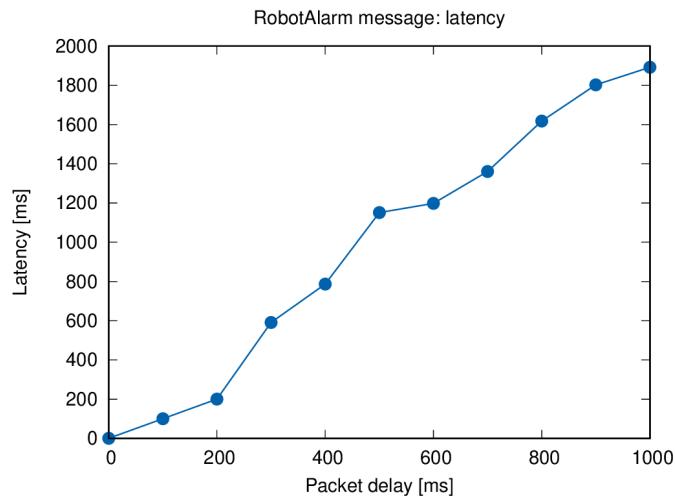


ros2:fastrtps

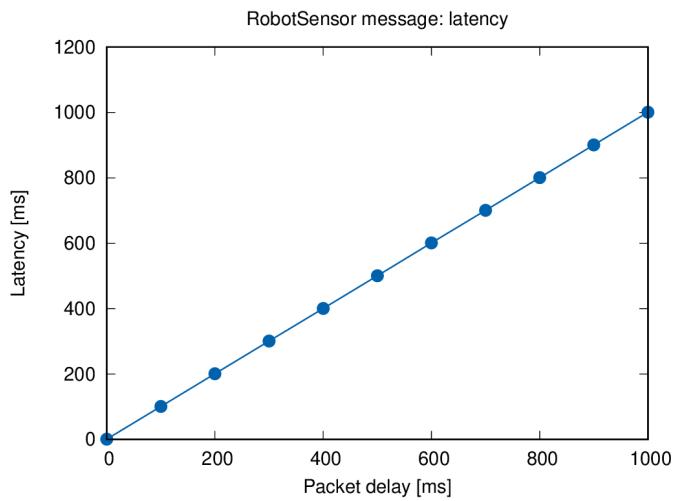
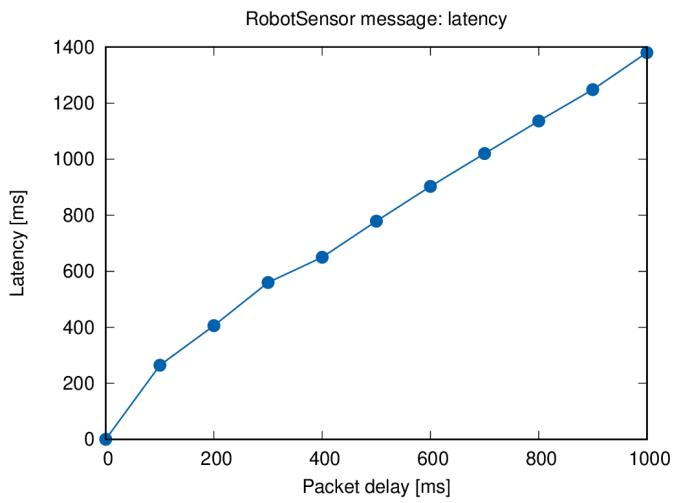
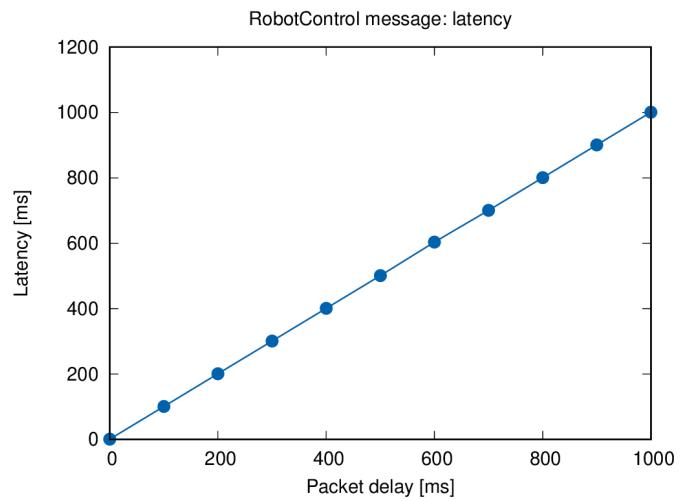
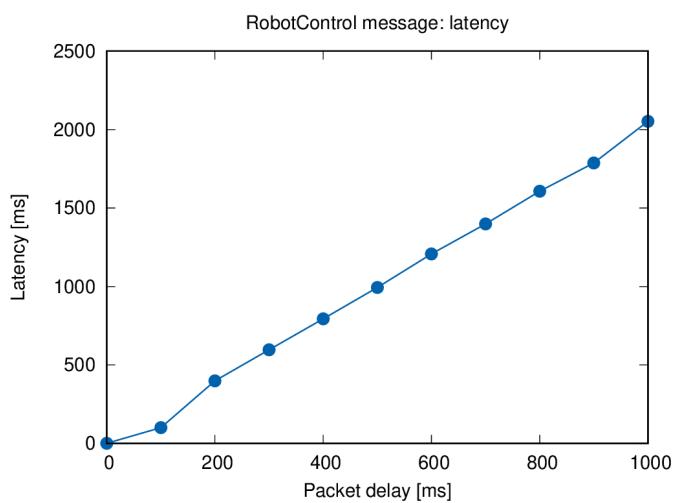
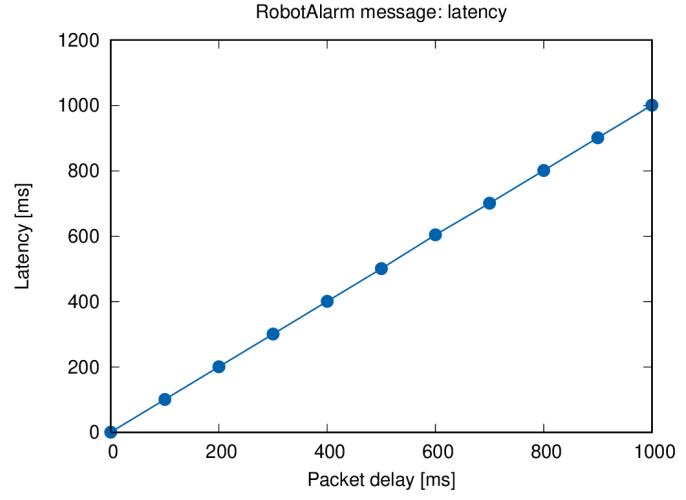


## 2. Delay: latency

ros1

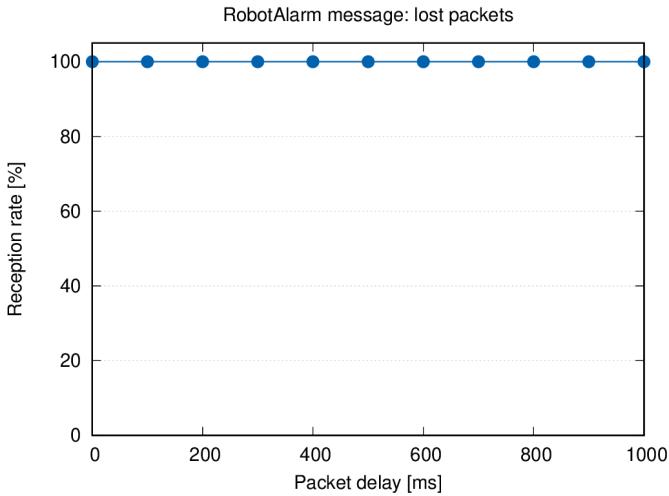


ros2:fastrtps

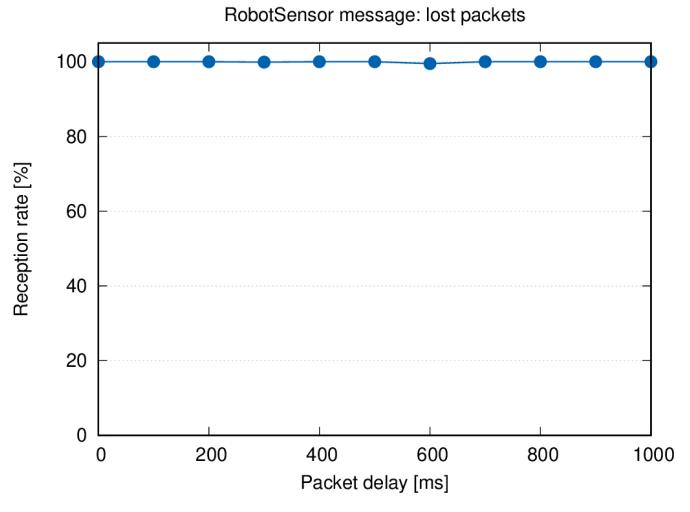
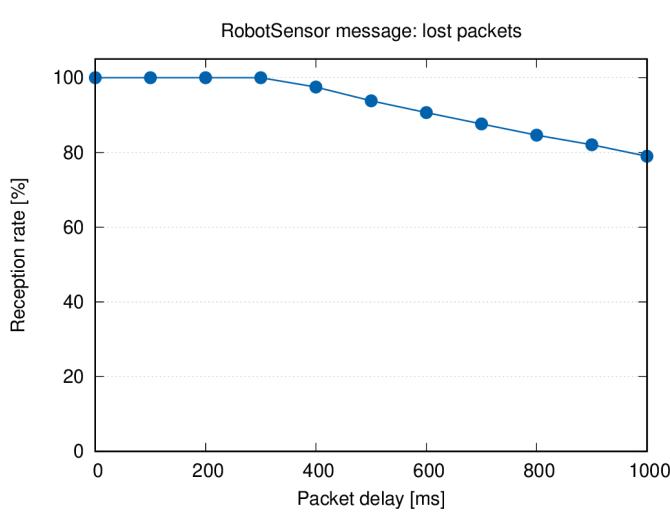
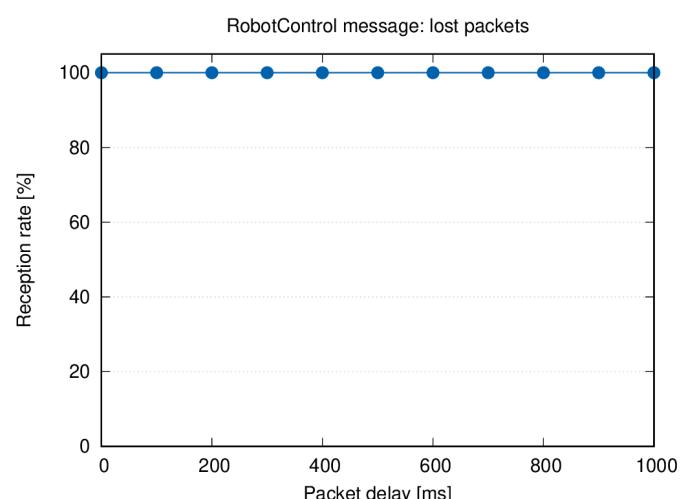
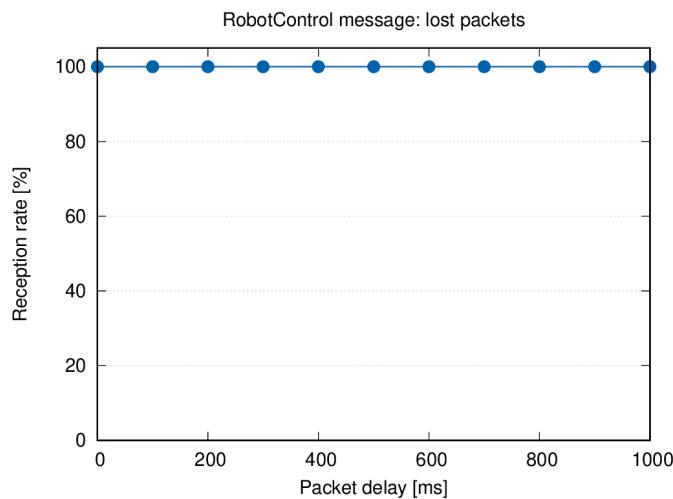
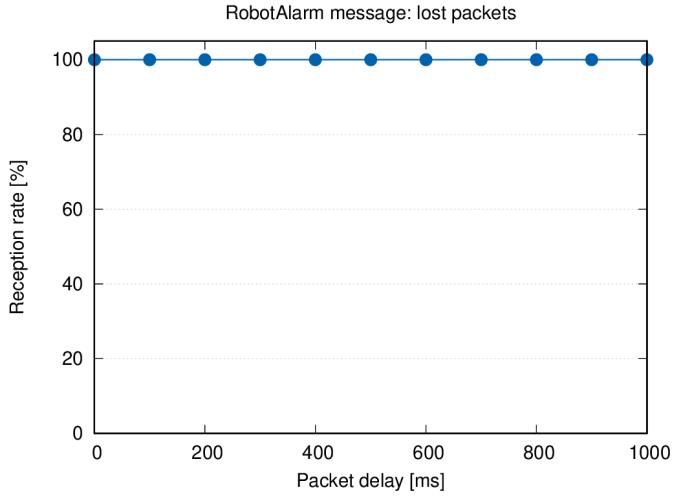


## 2. Delay: lost-packets

ros1

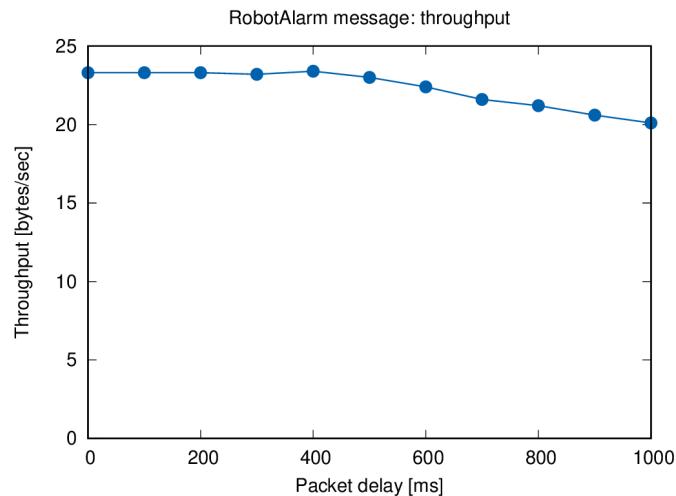


ros2:fastrtps

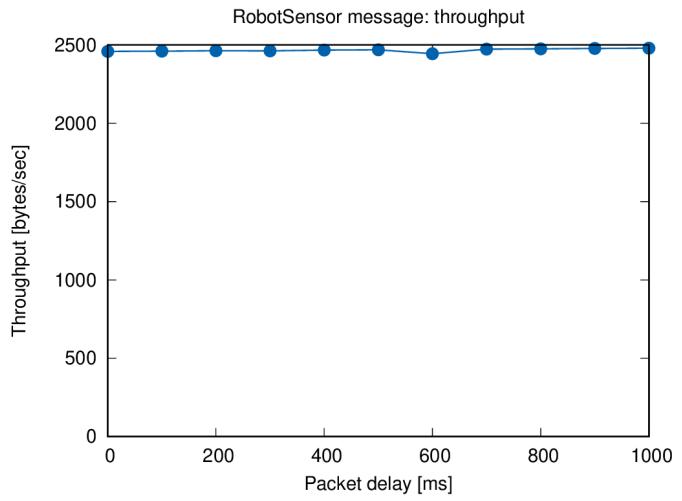
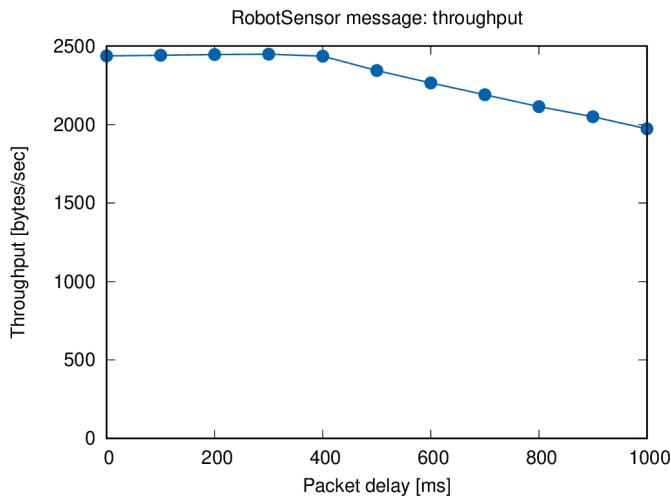
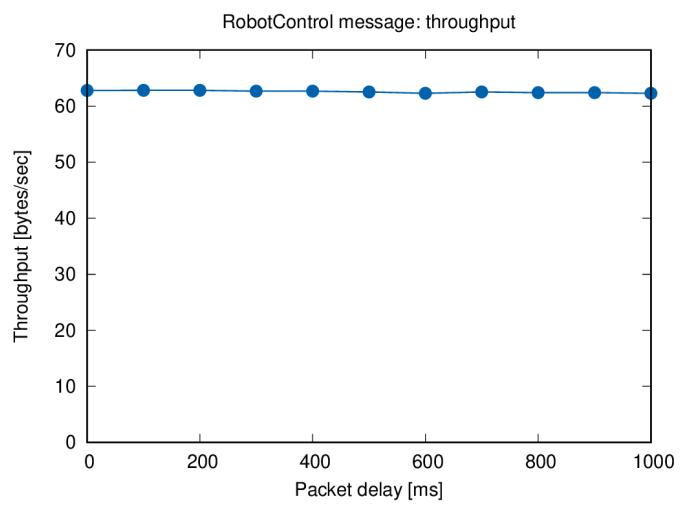
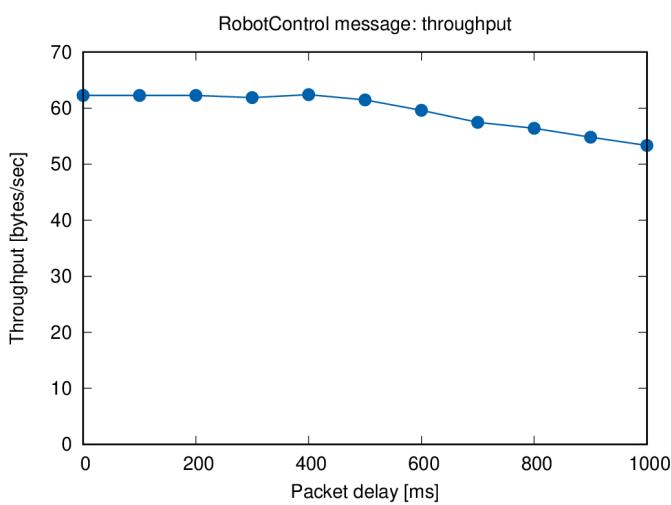
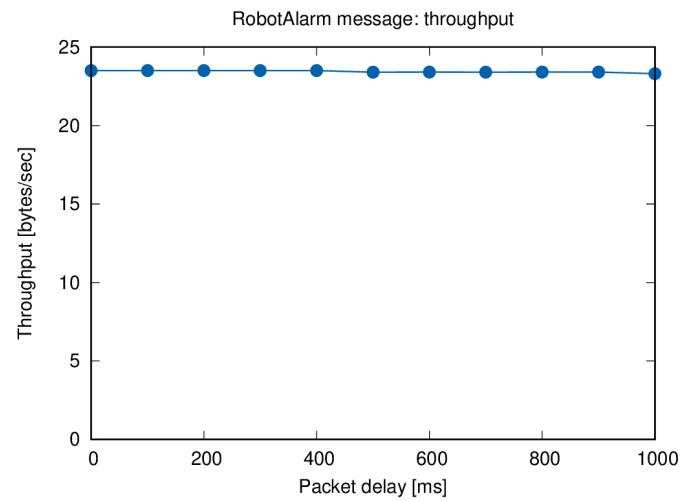


## 2. Delay: throughout

ros1



ros2:fastrtps

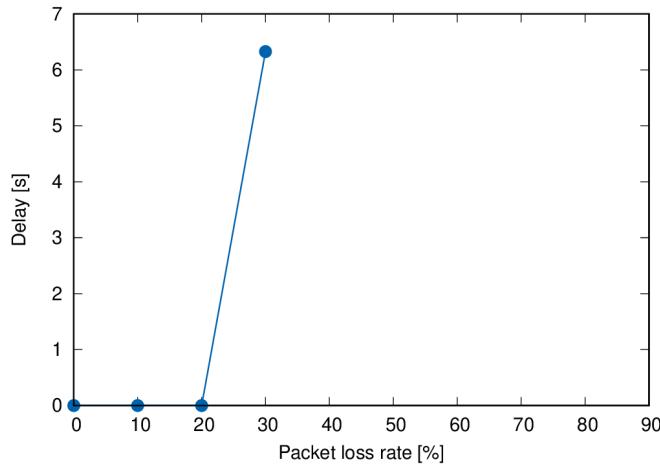


### 3. Loss:first-received

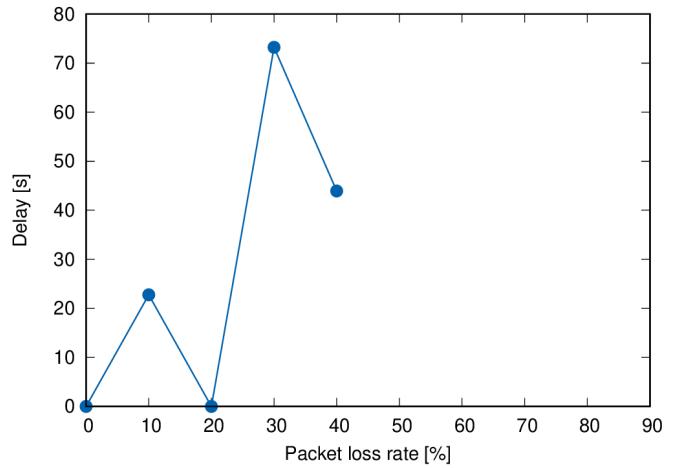
ros1

ros2:fastrtps

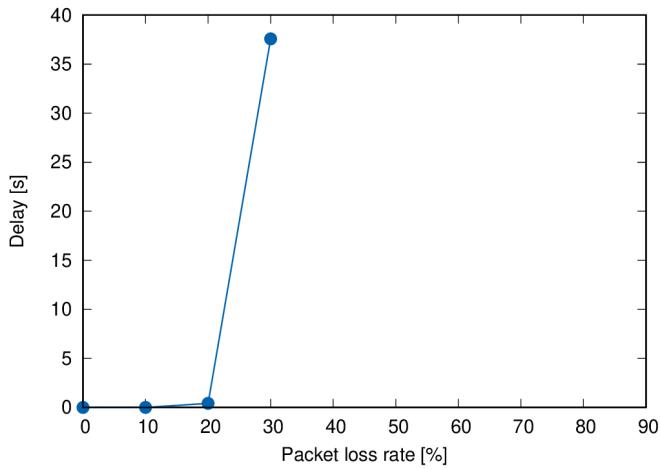
RobotAlarm message: first packet received



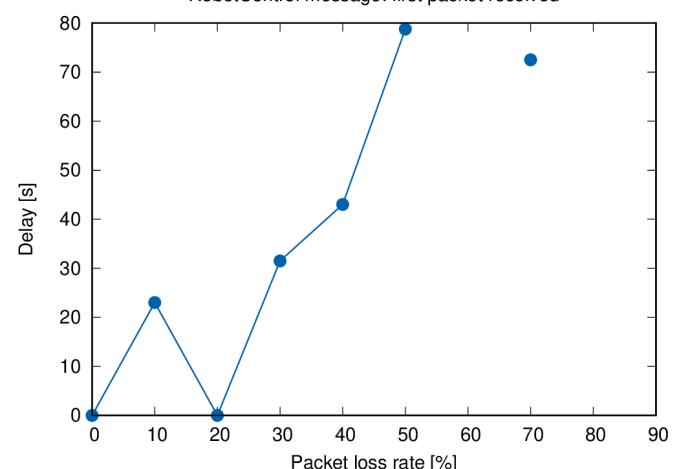
RobotAlarm message: first packet received



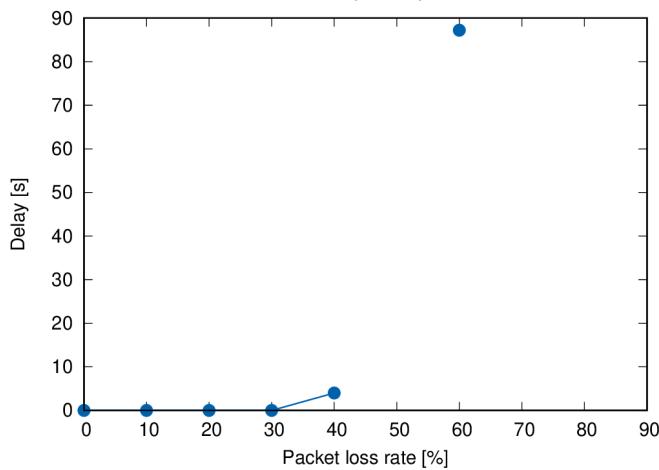
RobotControl message: first packet received



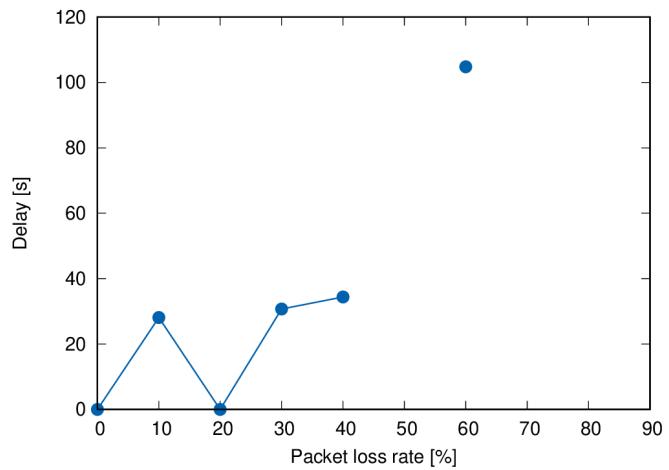
RobotControl message: first packet received



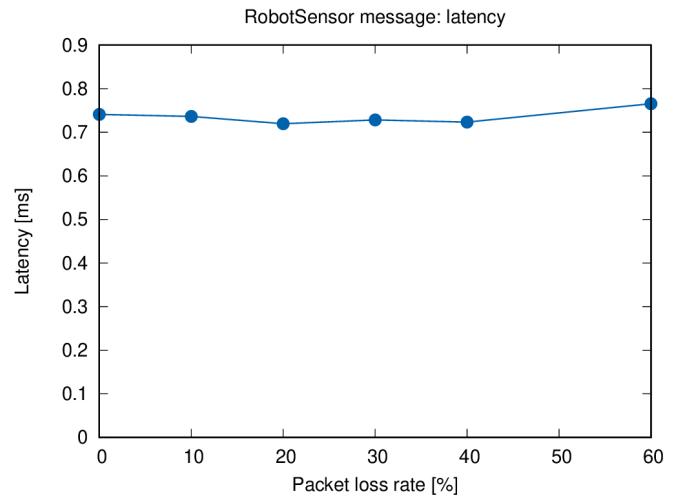
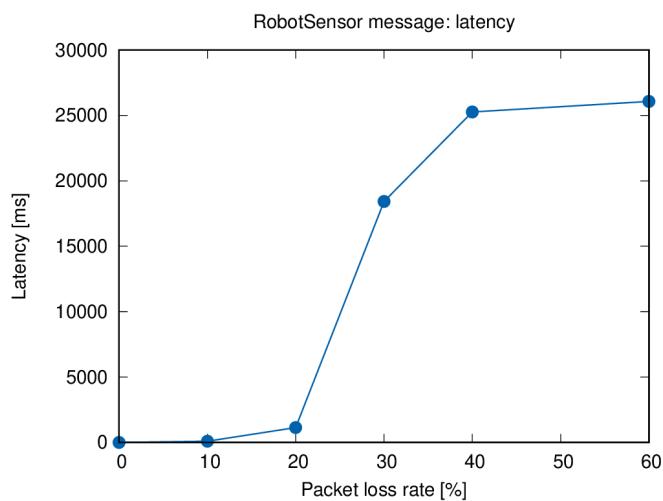
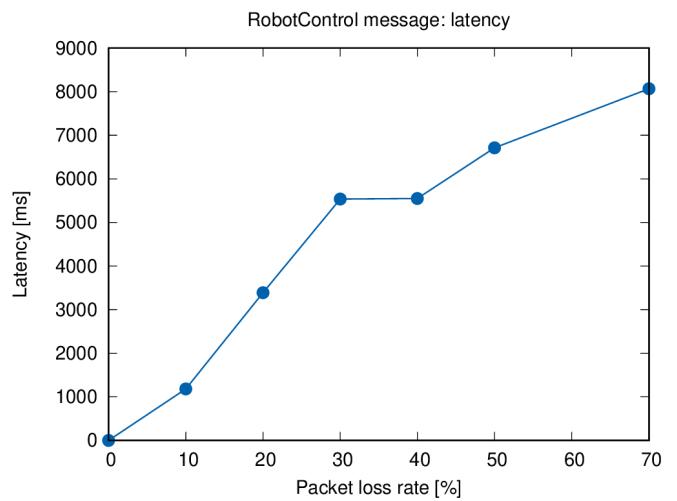
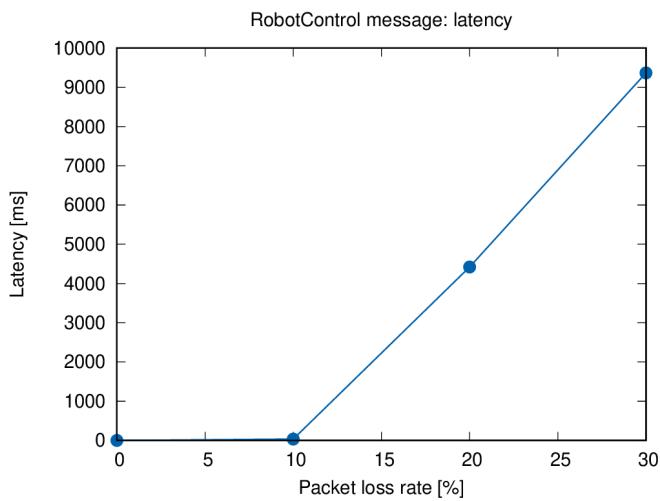
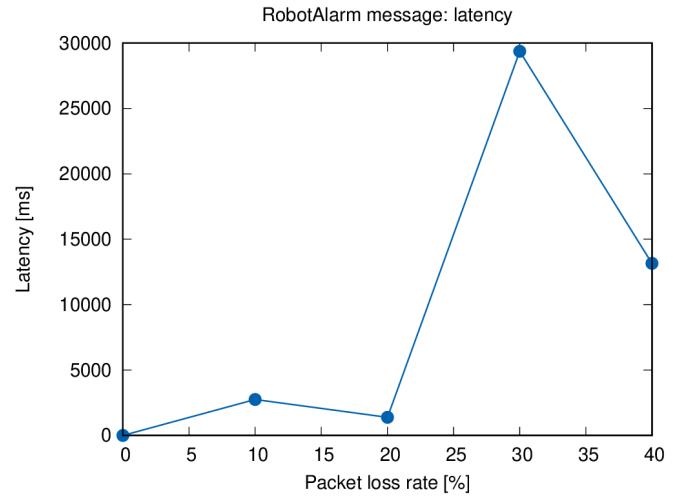
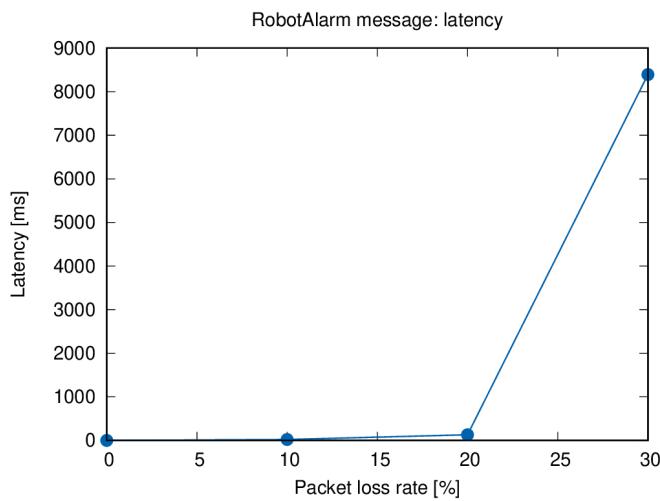
RobotSensor message: first packet received



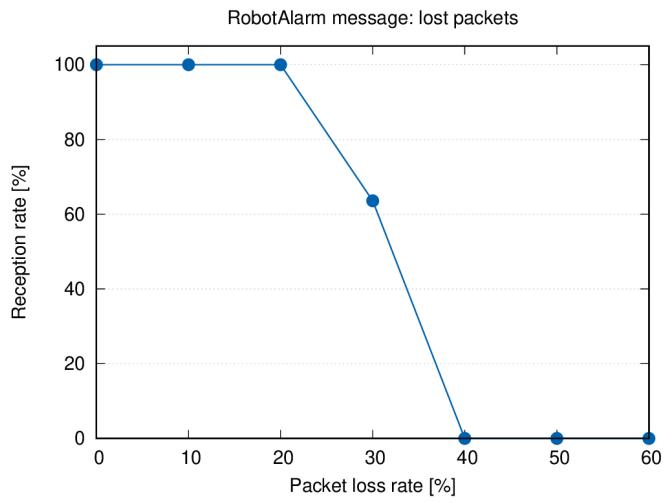
RobotSensor message: first packet received



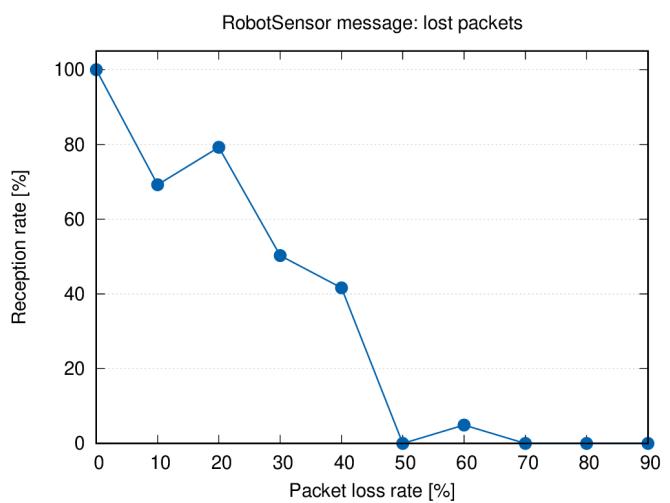
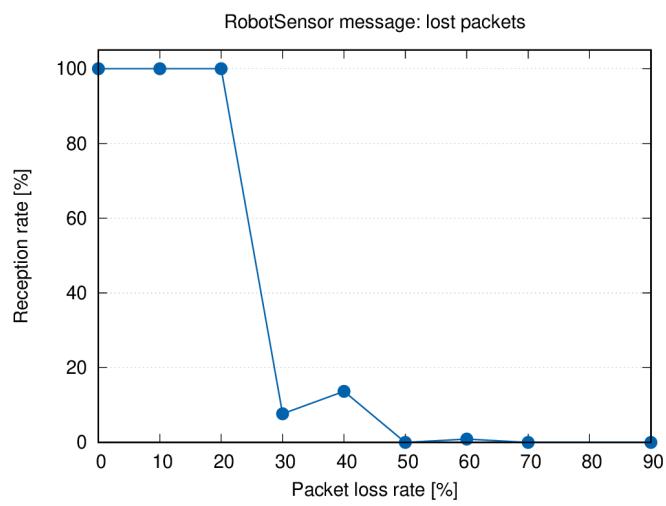
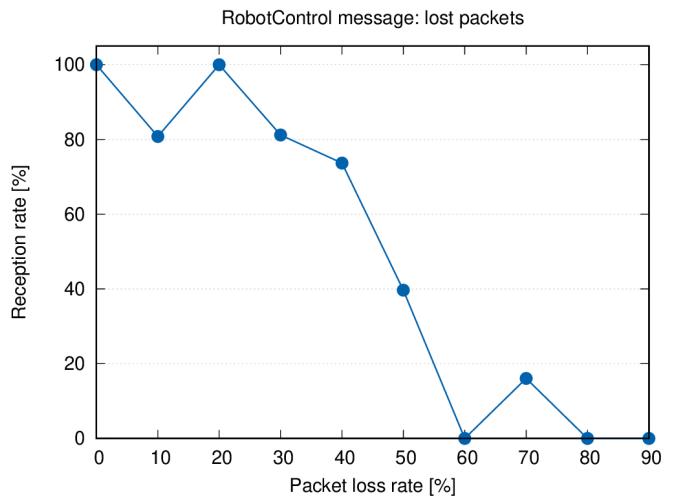
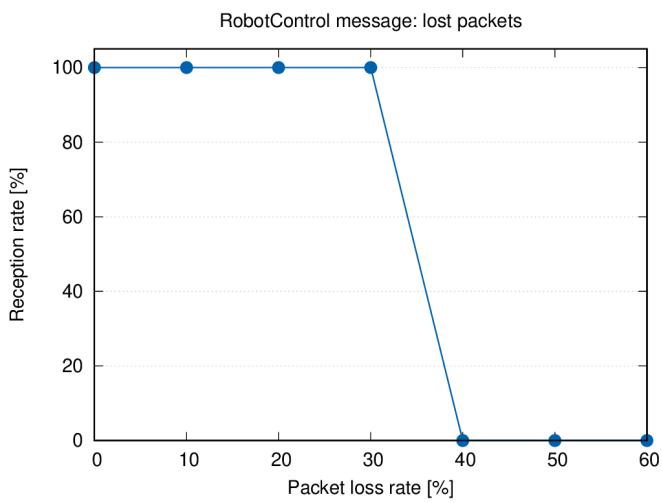
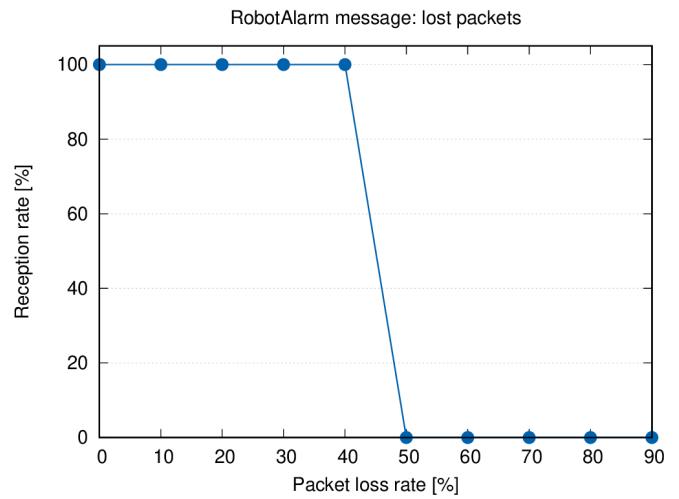
### 3. Loss: latency ros1



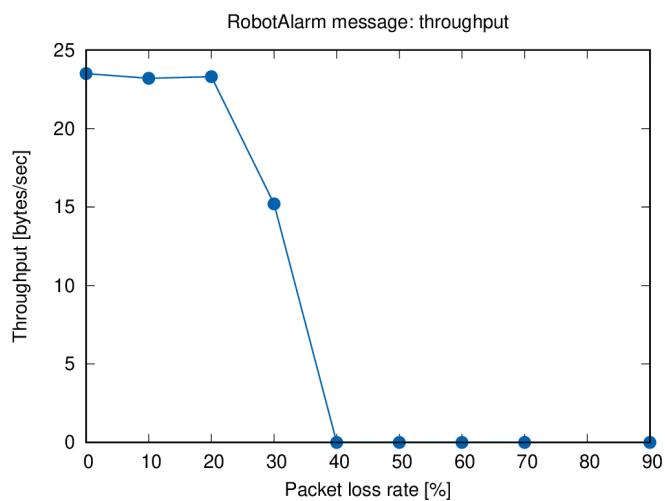
### 3. Loss: lost-packets ros1



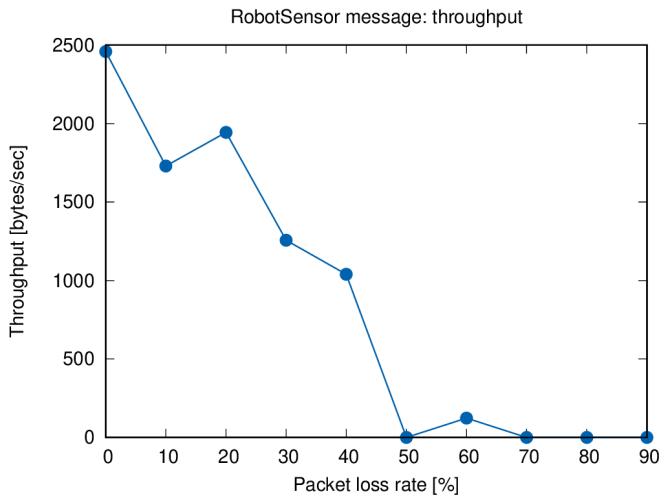
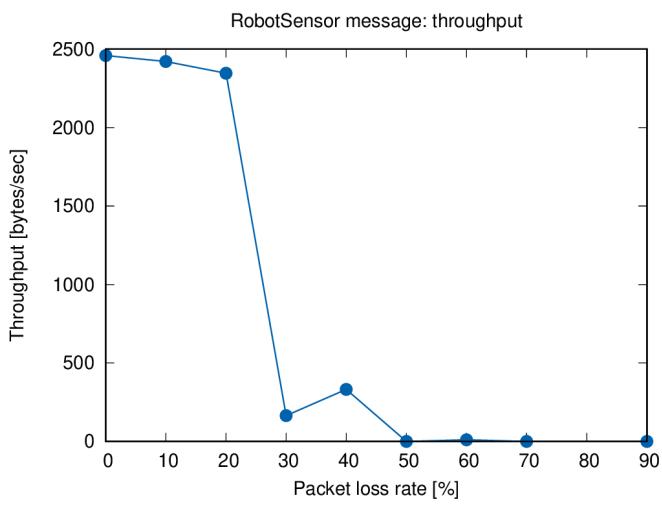
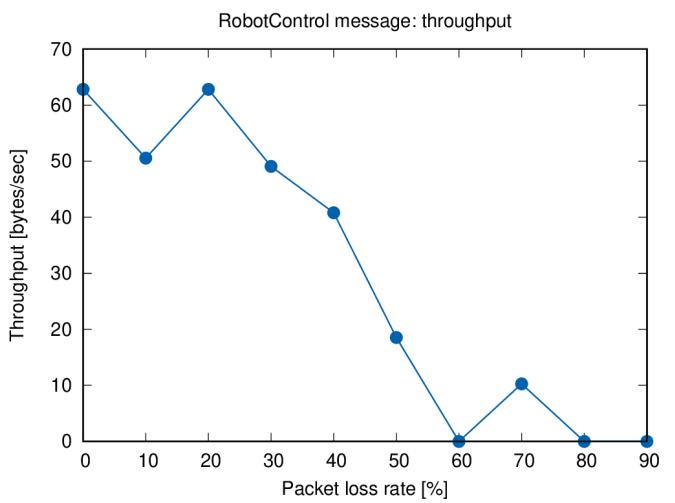
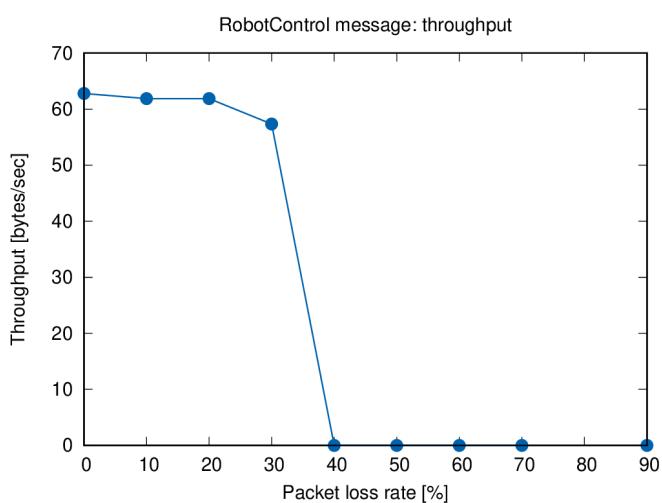
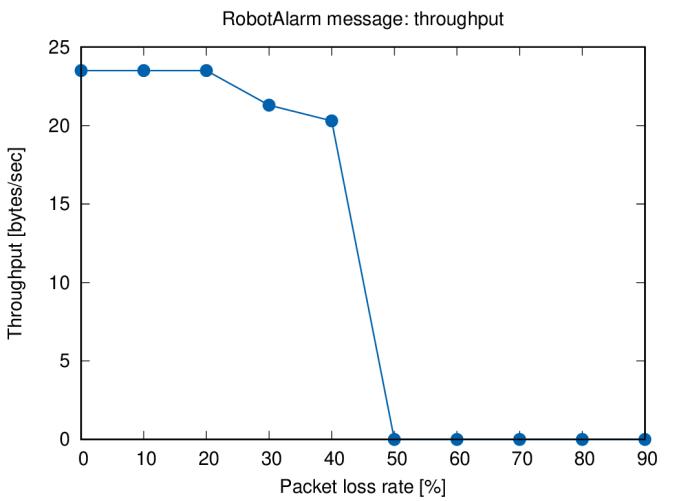
ros2:fastrtps



### 3. Loss: throughput ros1



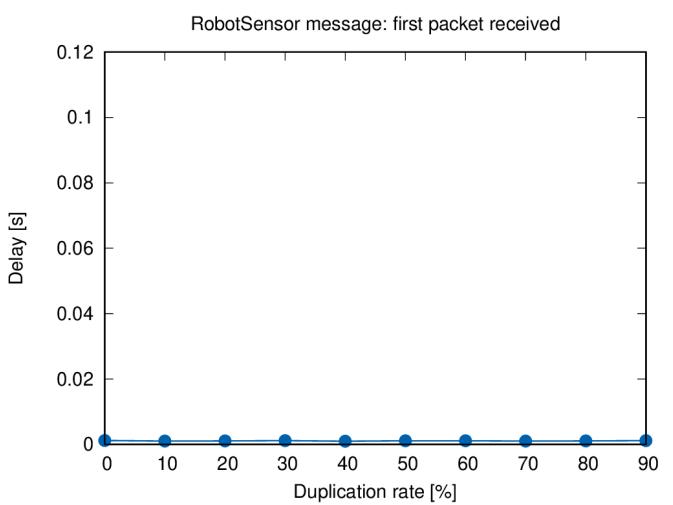
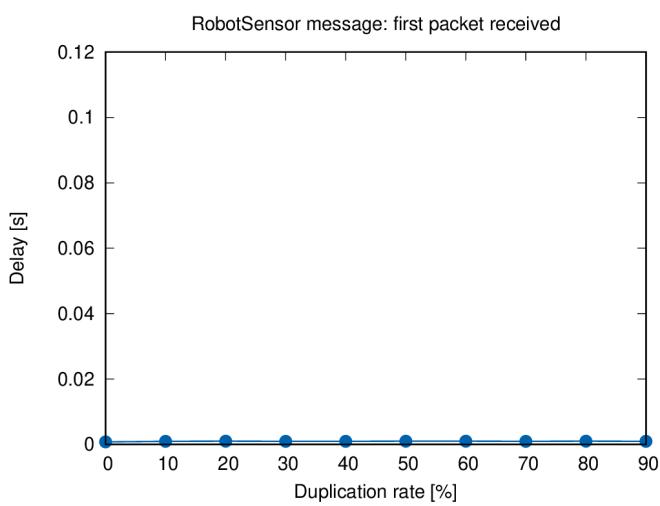
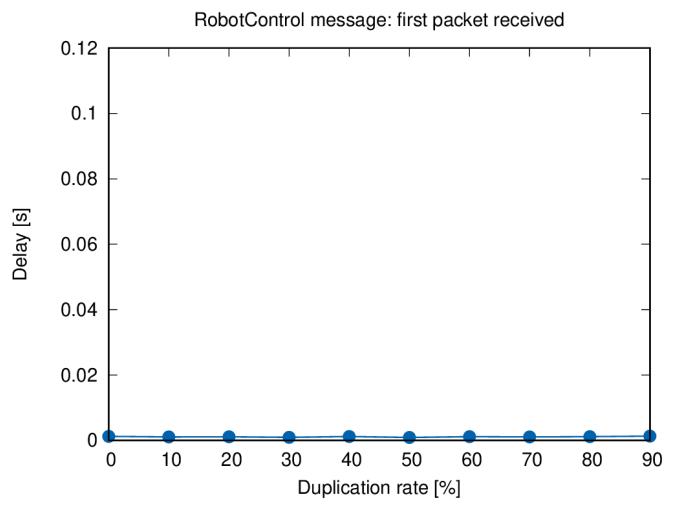
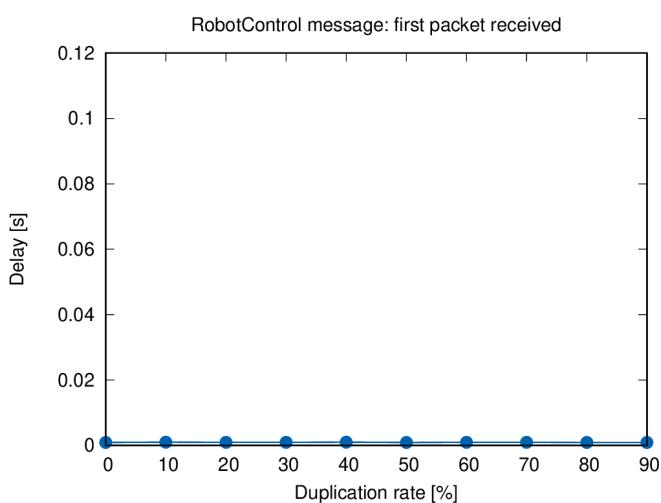
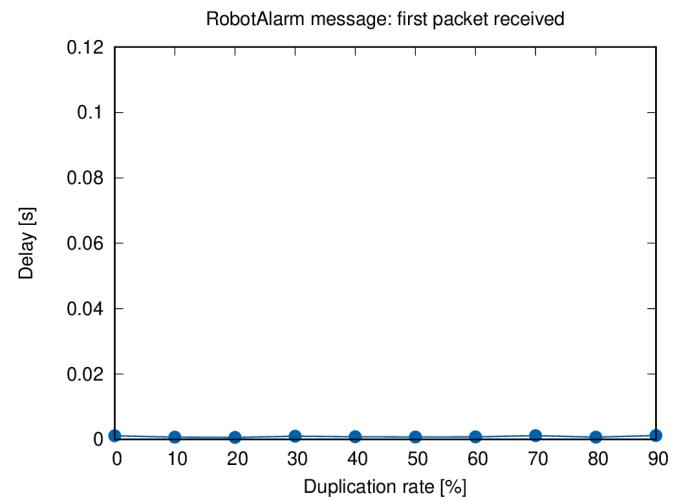
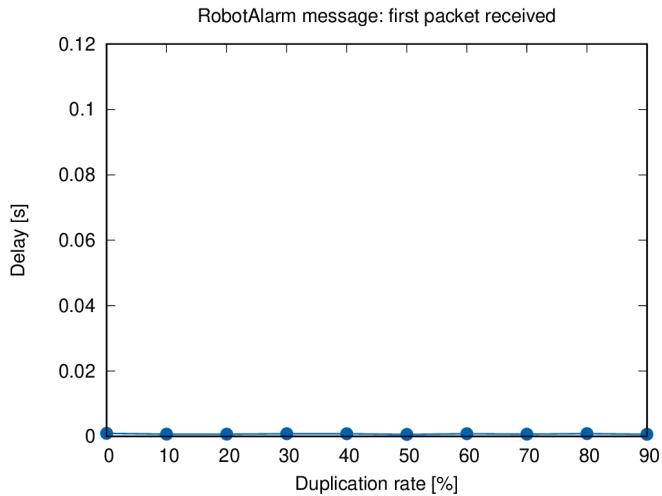
ros2:fastrtps



## 4. Duplication: first-received

ros1

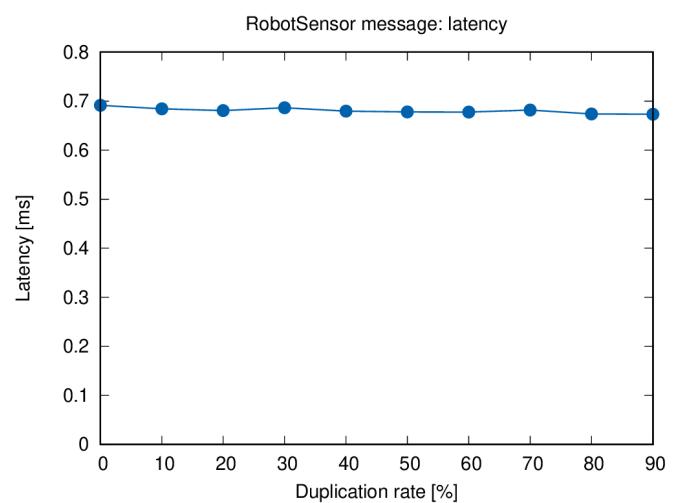
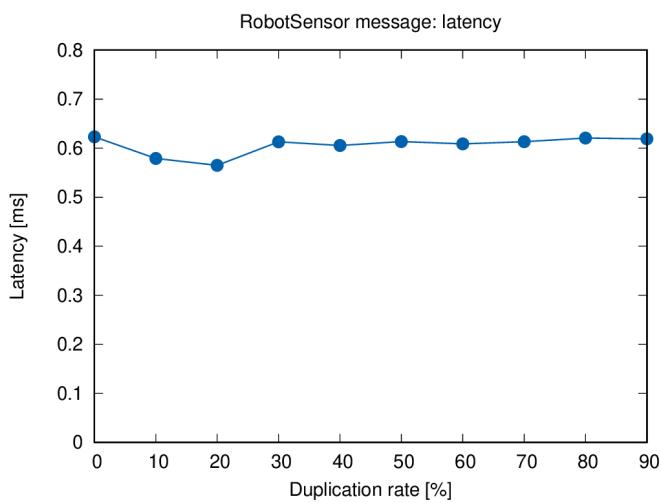
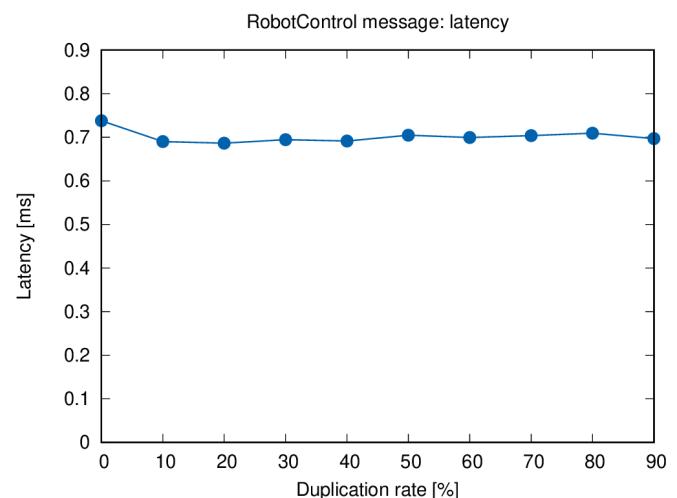
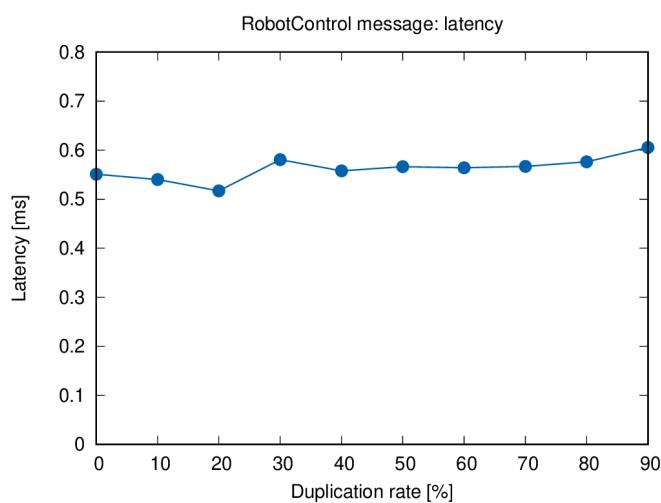
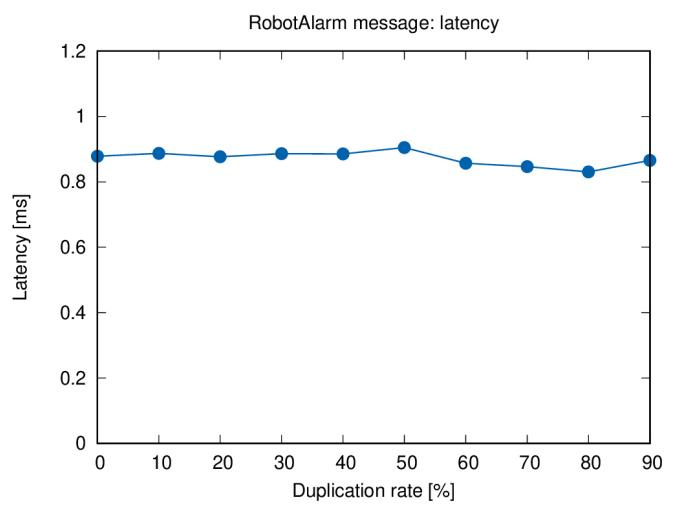
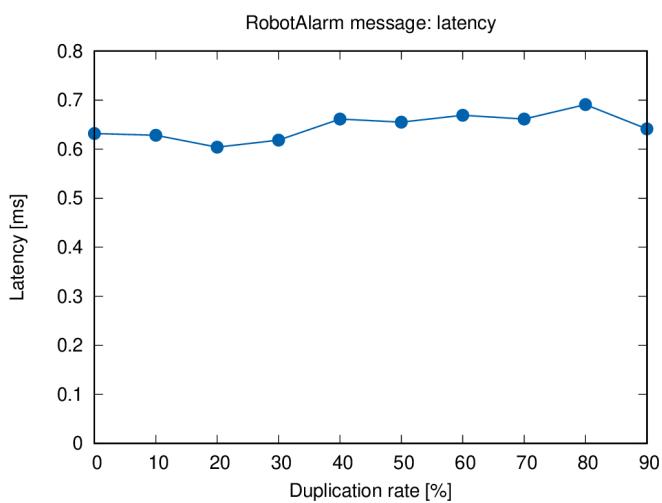
ros2:fastrtps



## 4. Duplication: latency

ros1

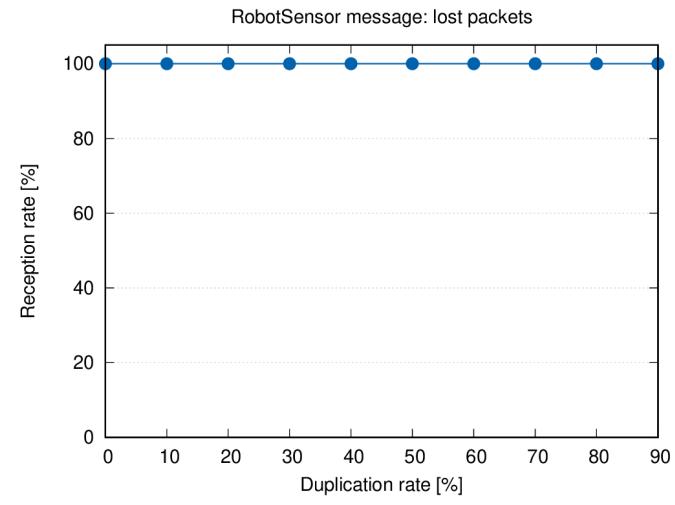
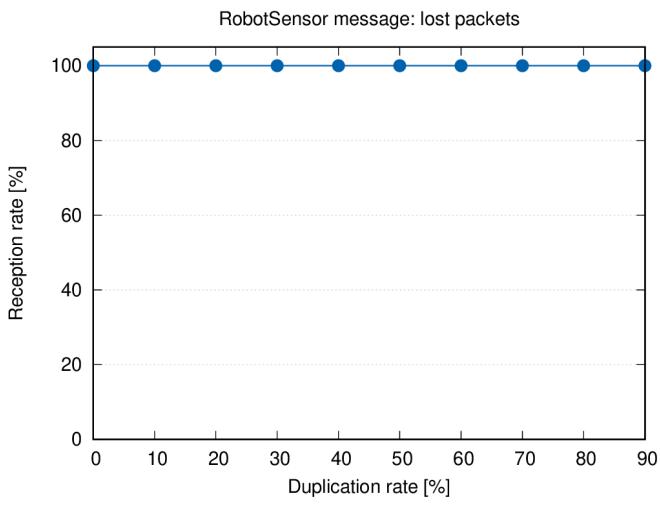
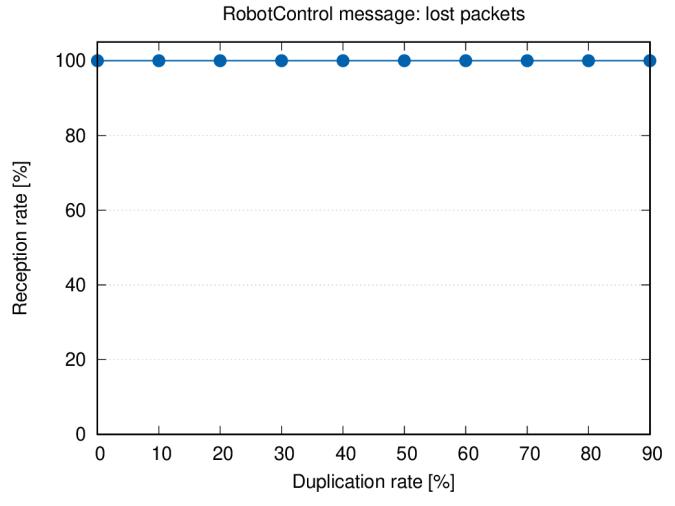
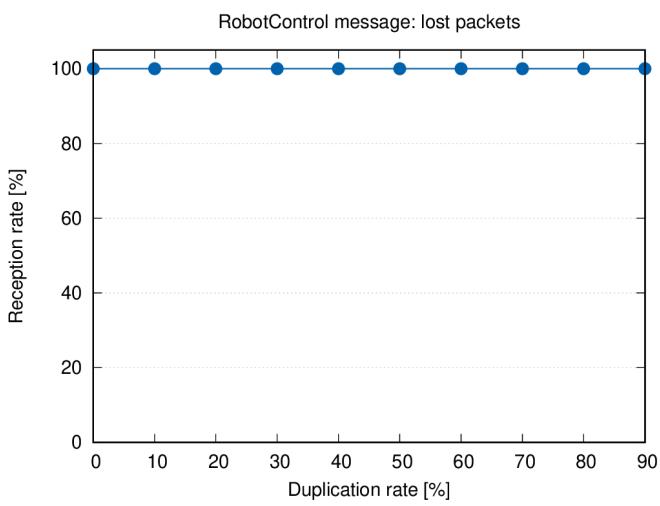
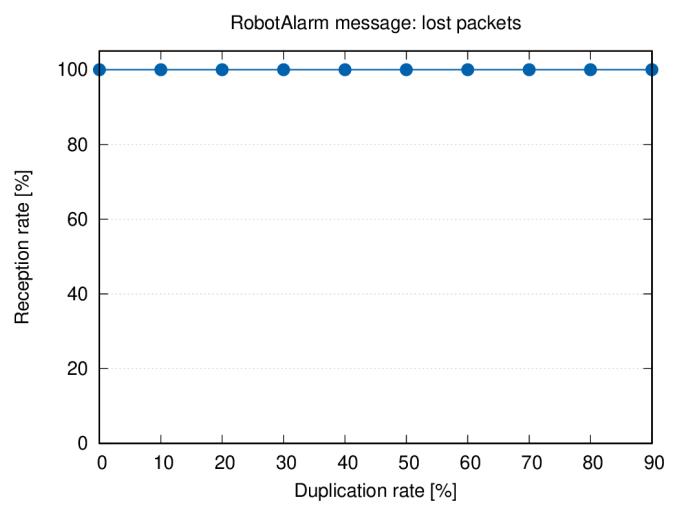
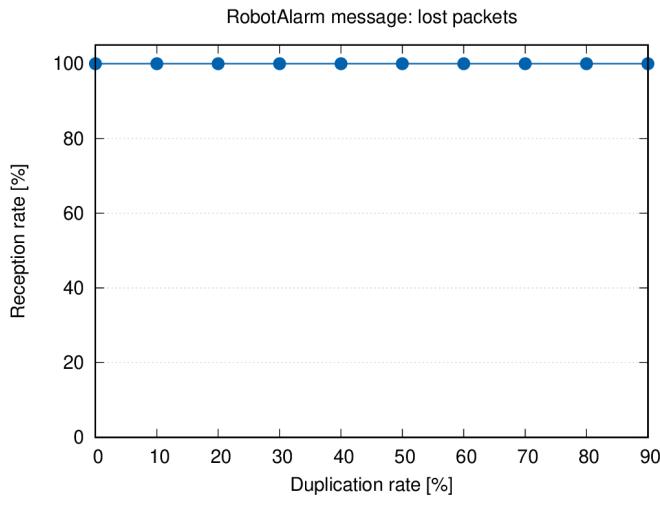
ros2:fastrtps



## 4. Duplication: lost-packets

ros1

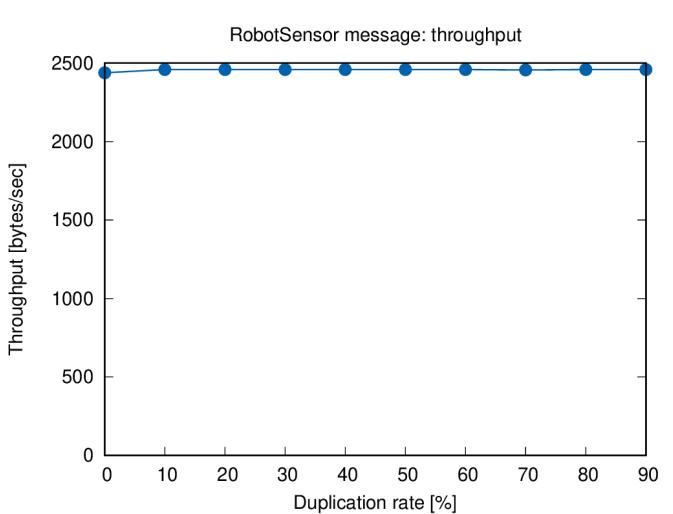
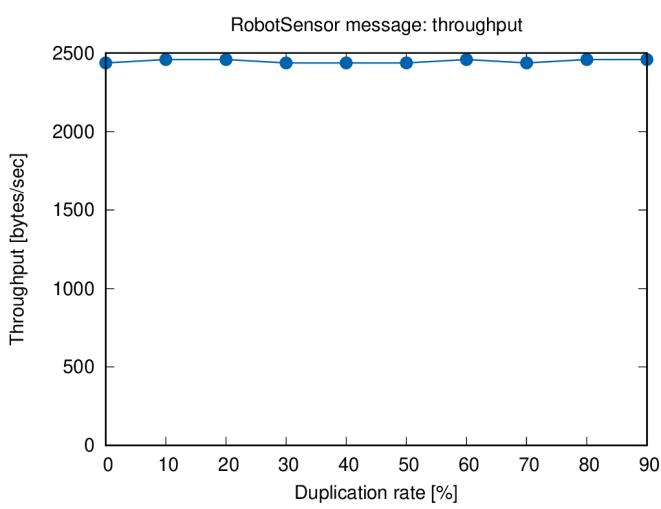
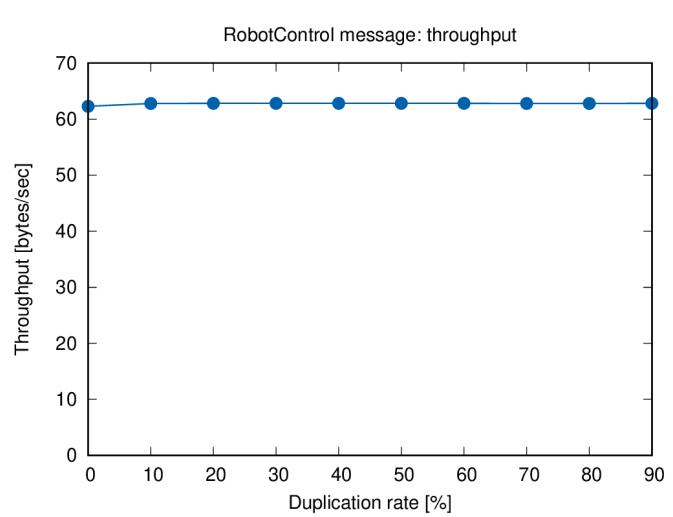
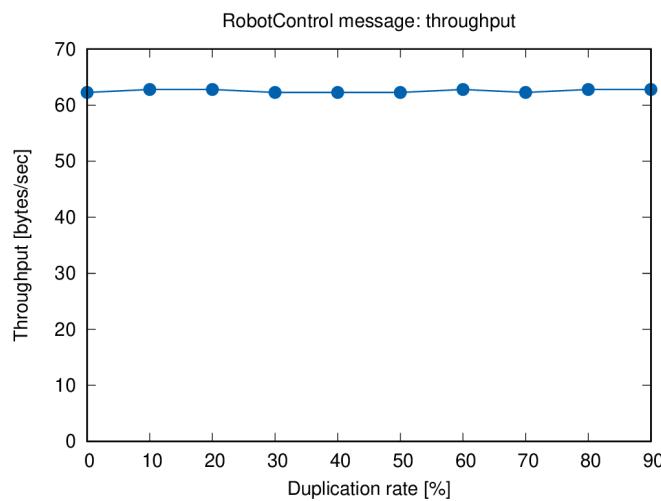
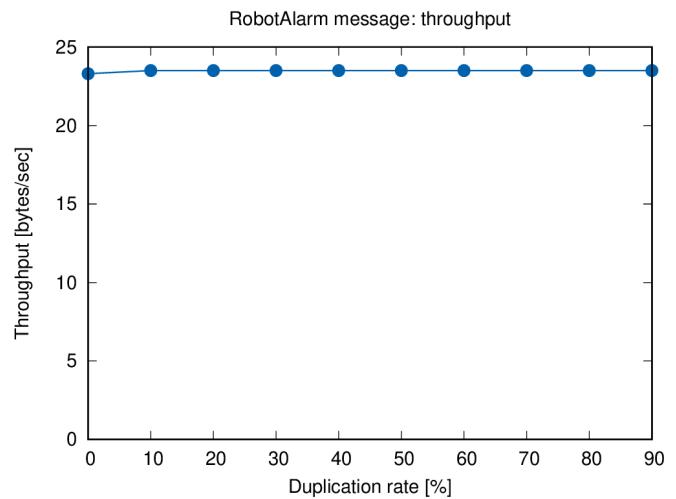
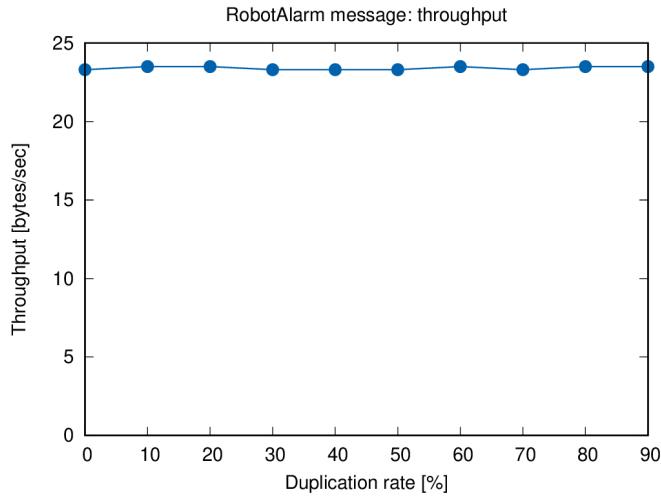
ros2:fastrtps



## 4. Duplication: throughput

ros1

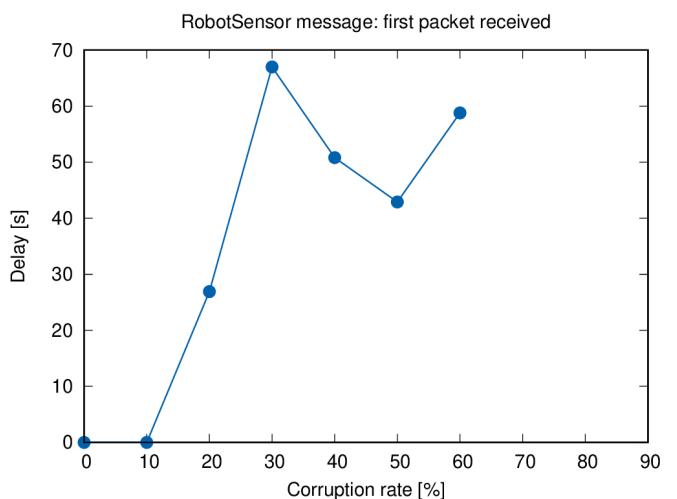
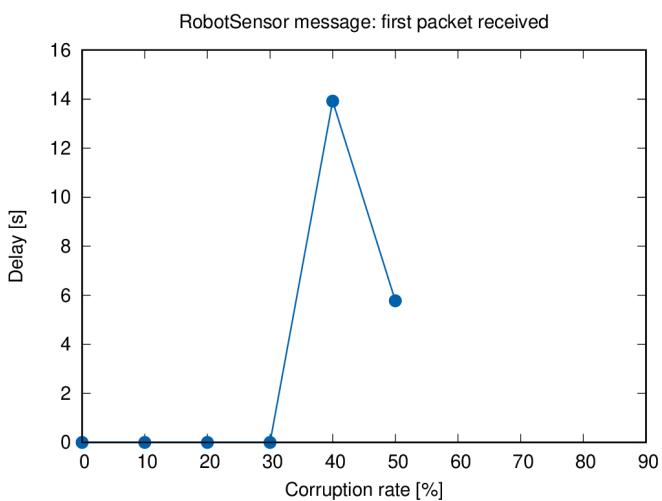
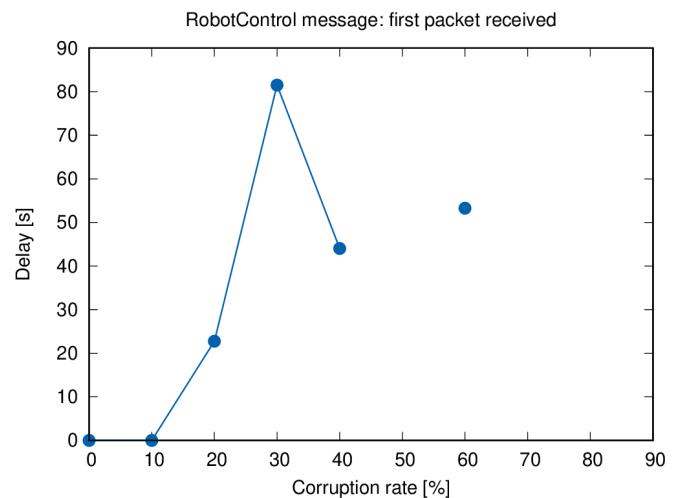
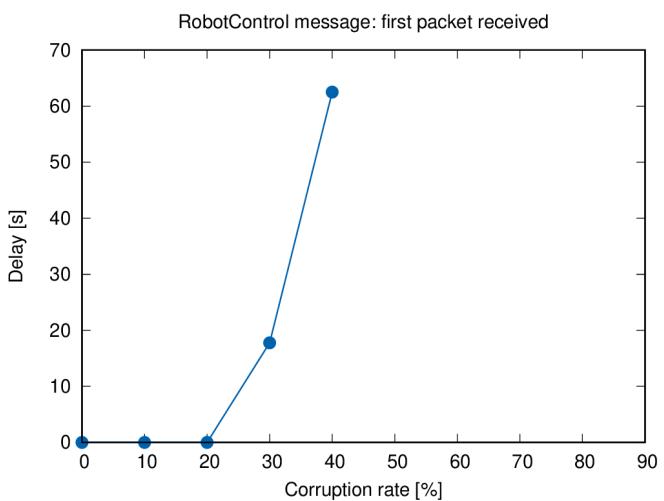
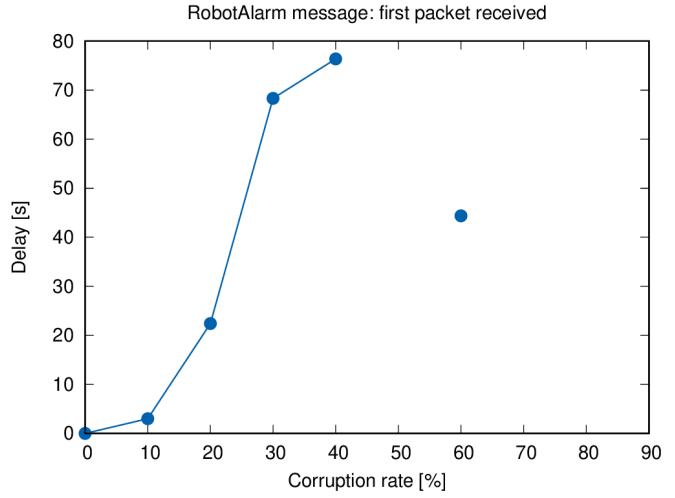
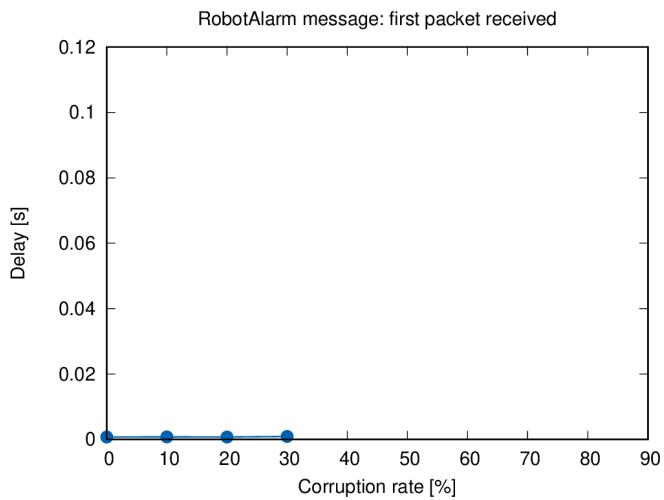
ros2:fastrtps



## 5. Corruption: first-received

ros1

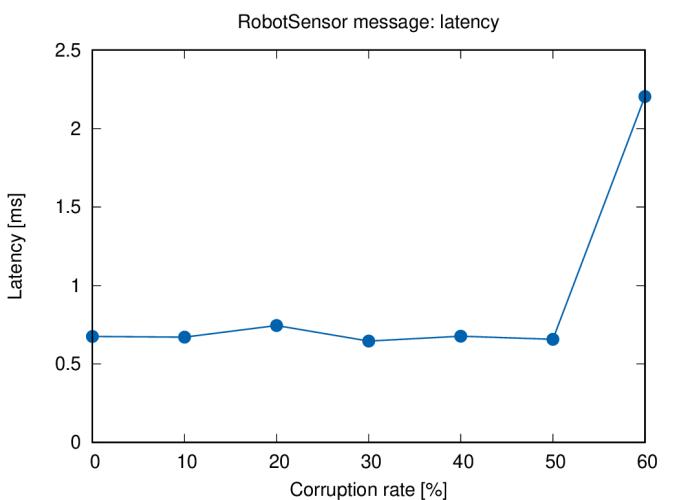
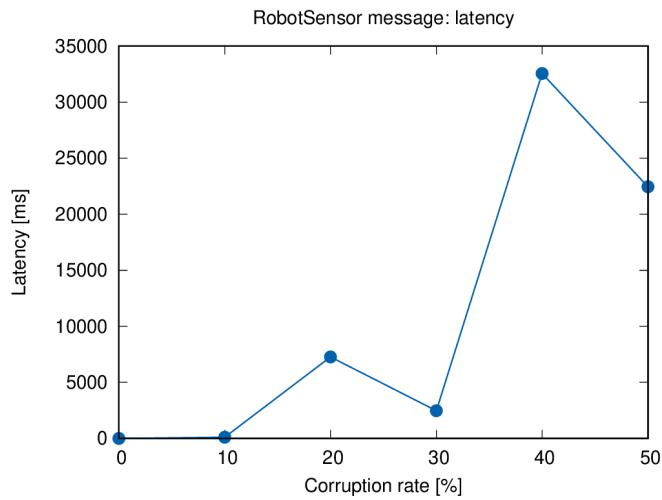
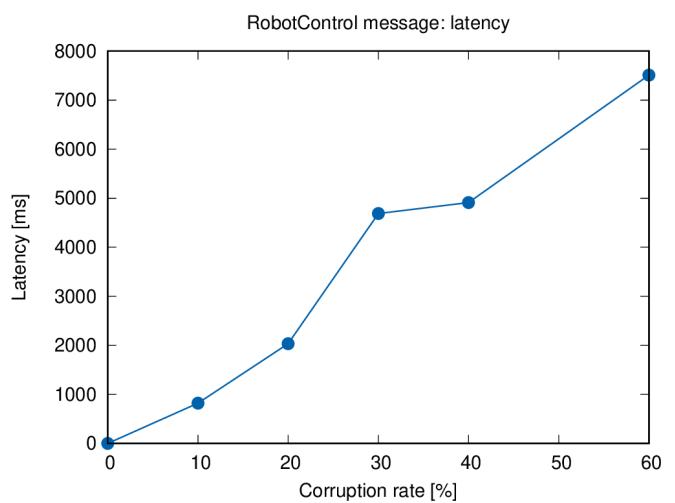
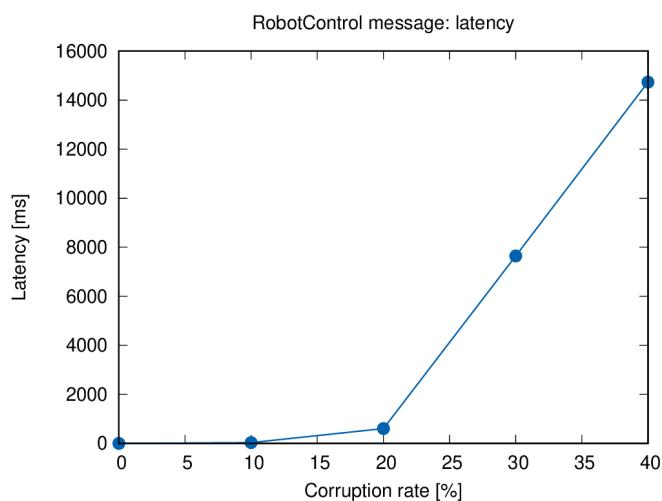
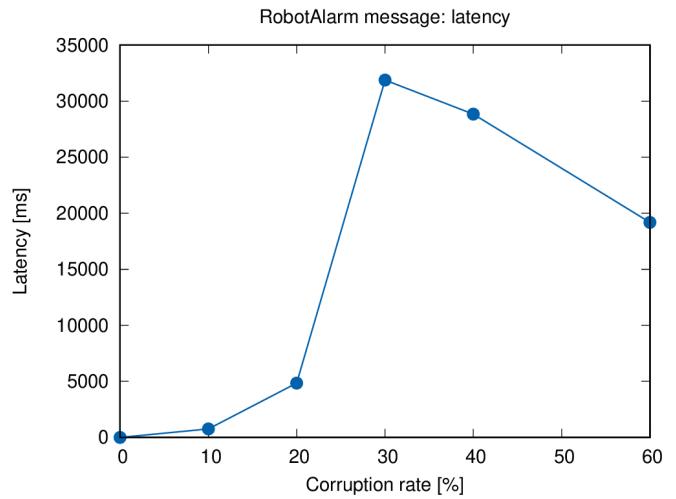
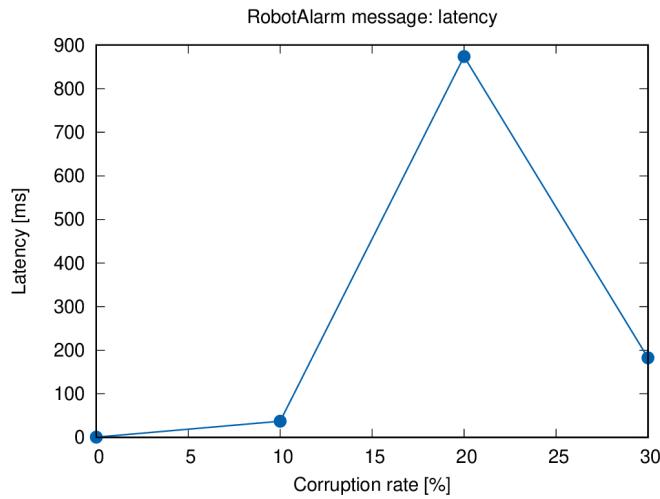
ros2:fastrtps



## 5. Corruption: latency

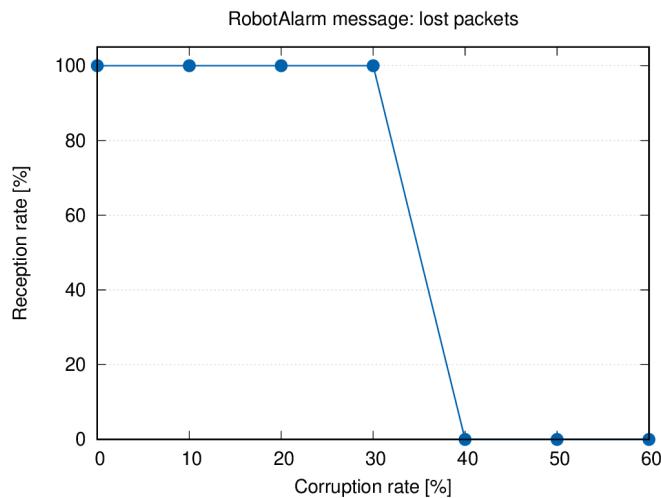
ros1

ros2:fastrtps

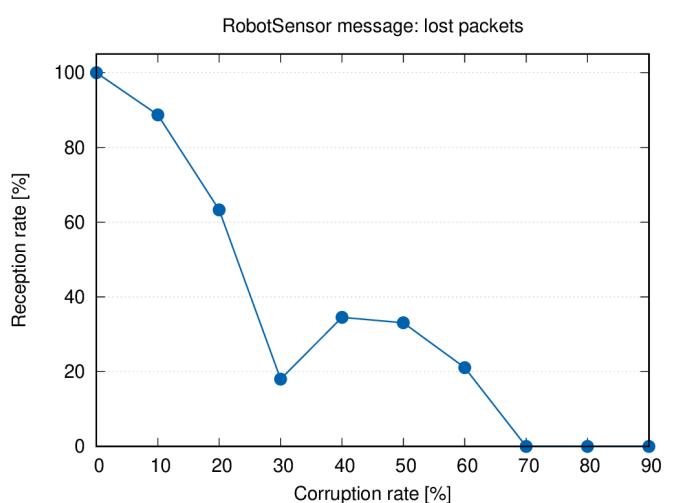
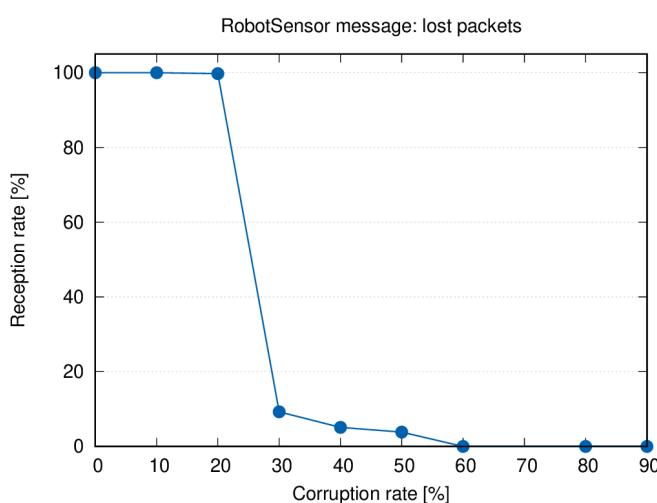
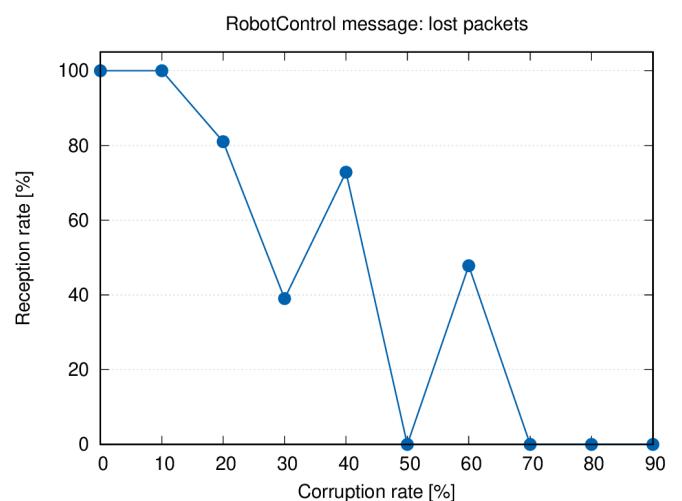
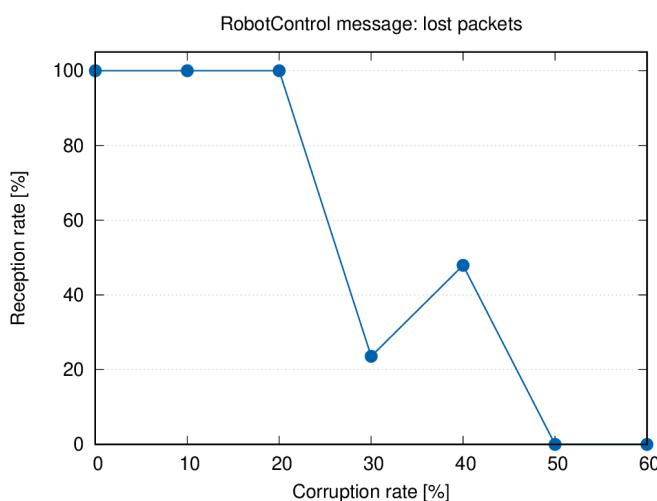
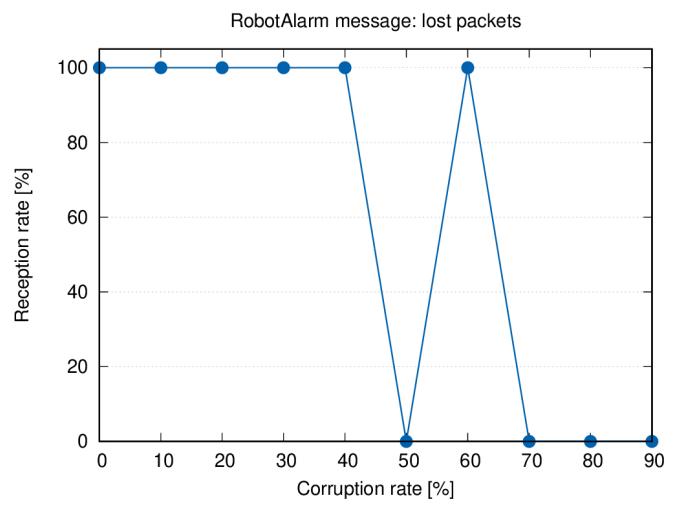


## 5. Corruption: lost-packets

ros1



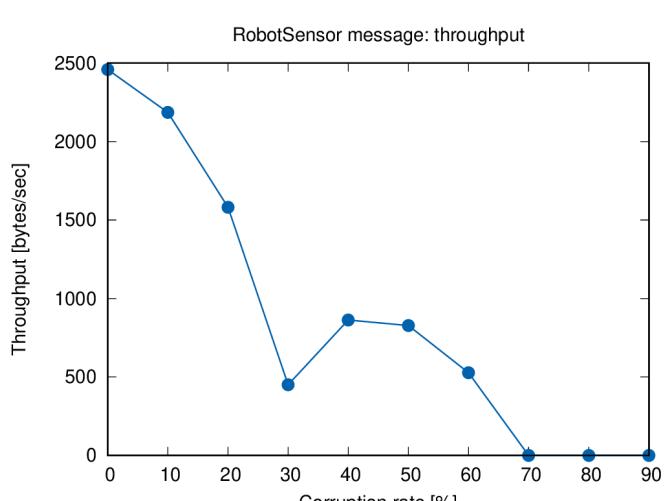
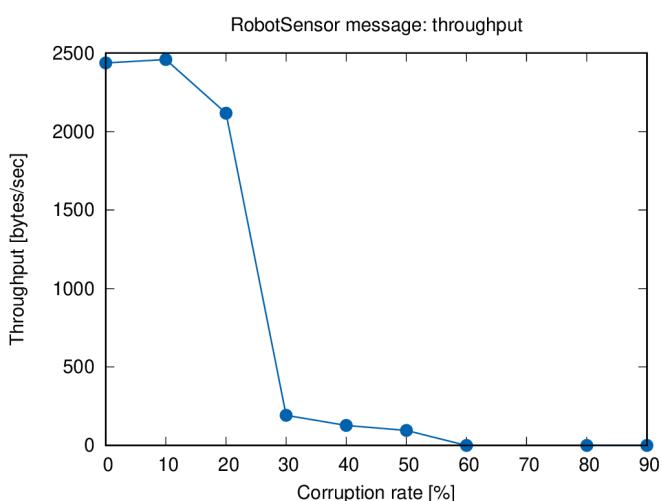
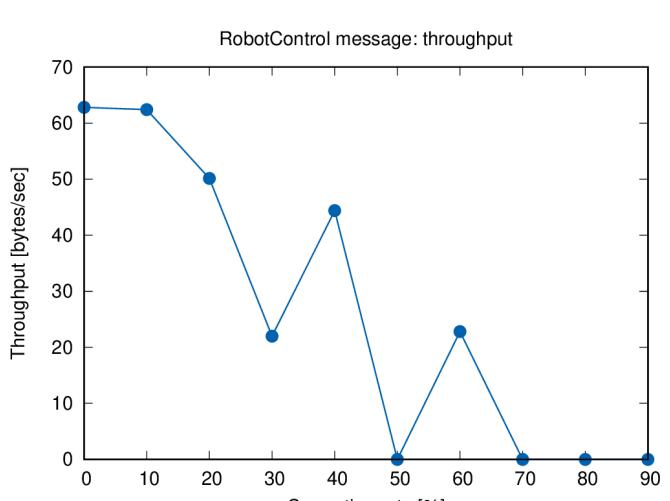
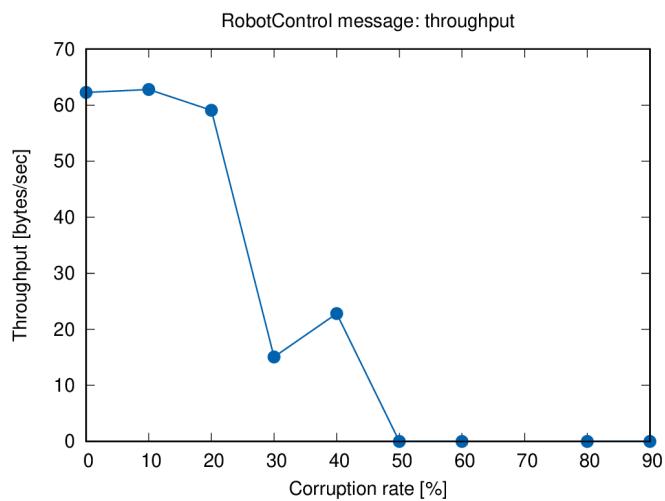
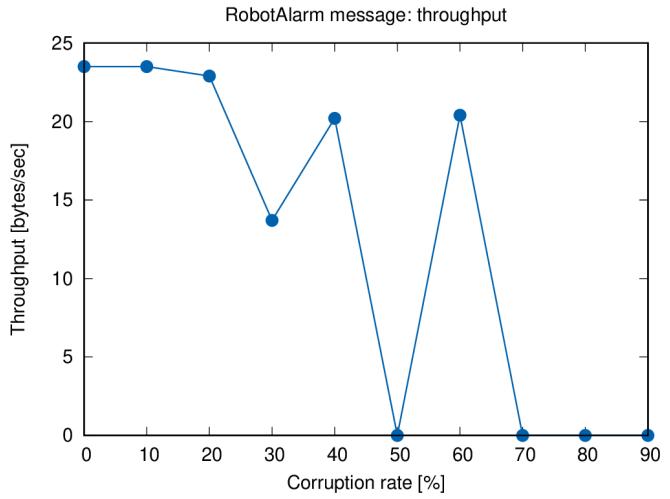
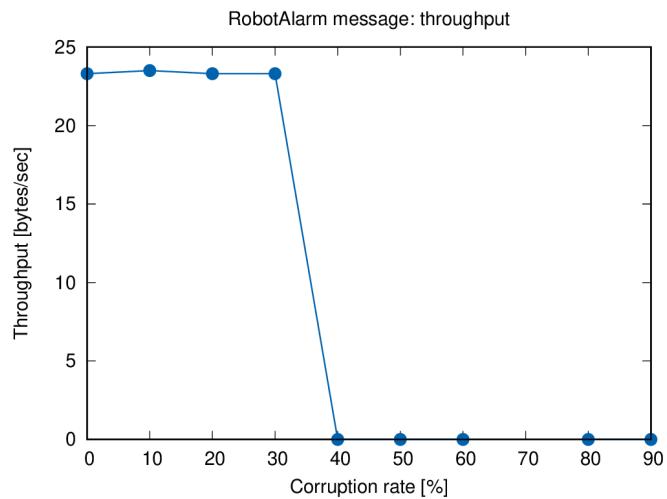
ros2:fastrtps



## 5. Corruption: throughput

ros1

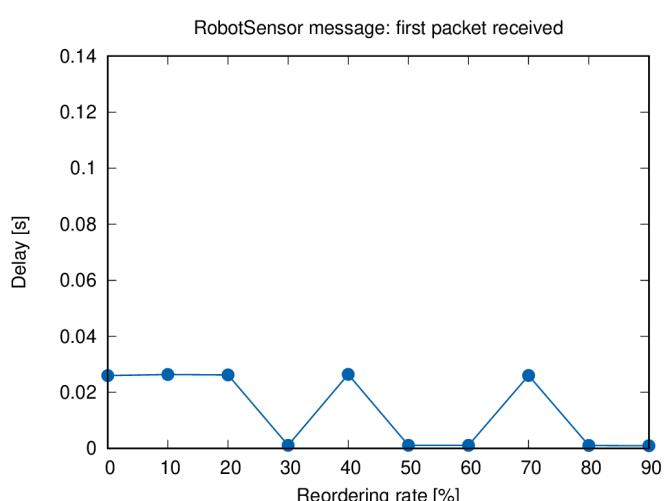
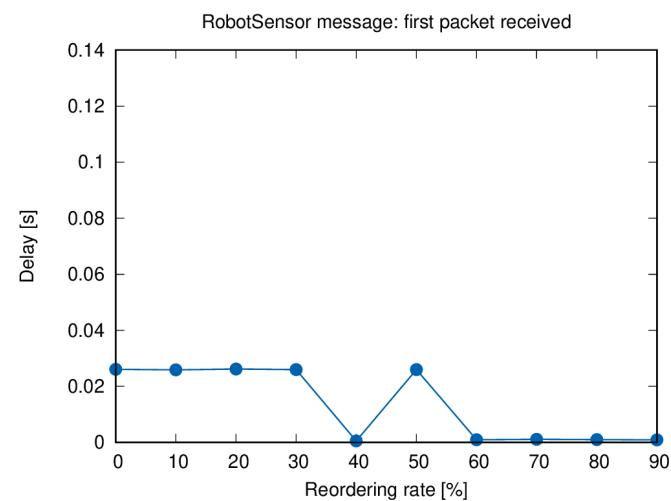
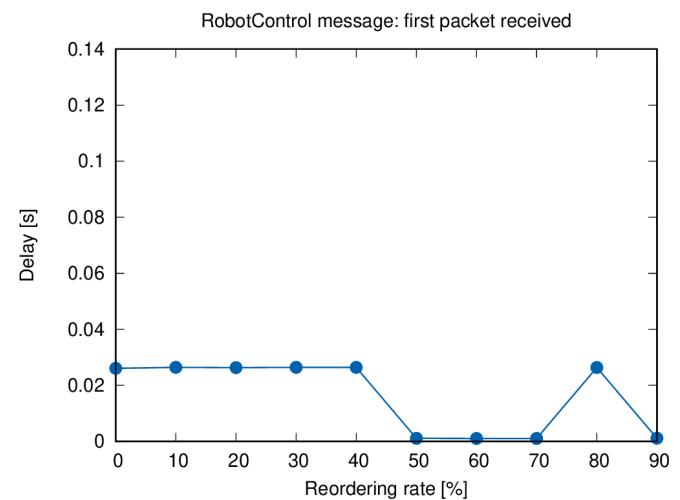
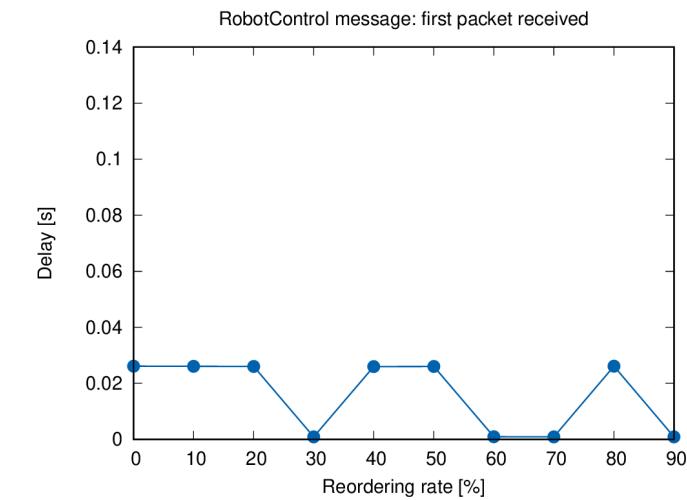
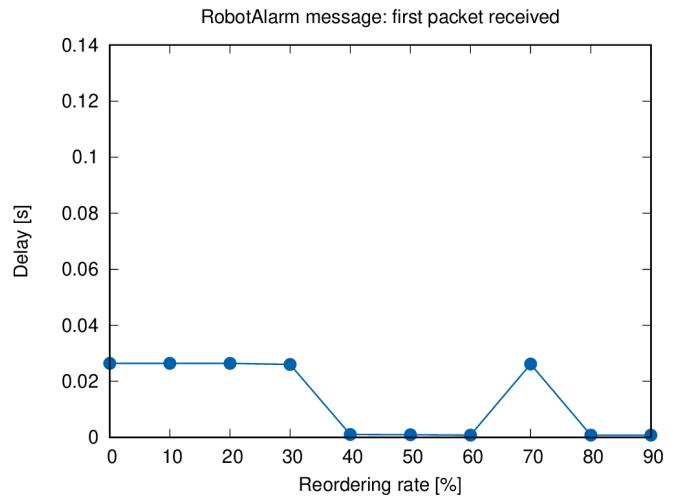
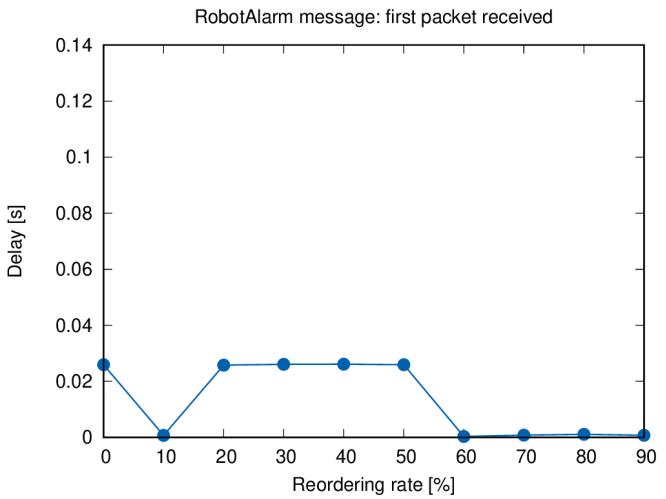
ros2:fastrtps



## 6. Reorder: first-received

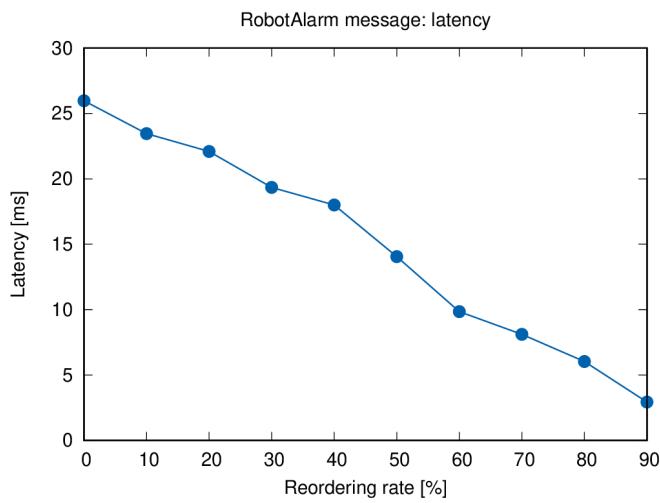
ros1

ros2:fastrtps

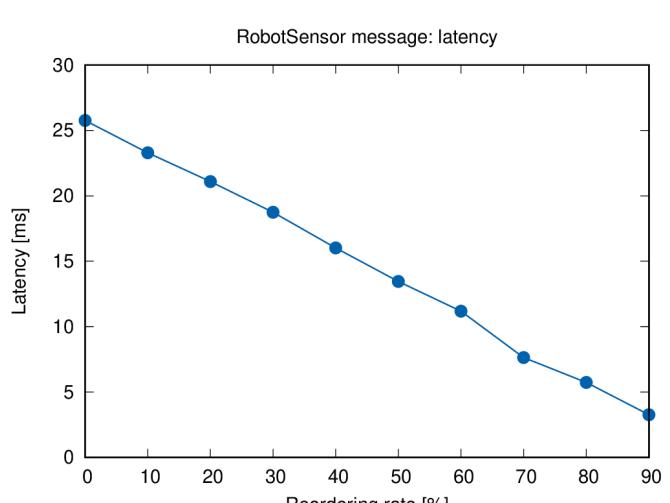
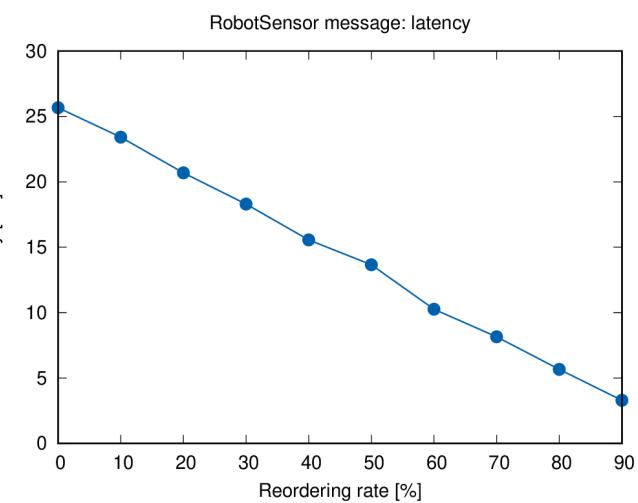
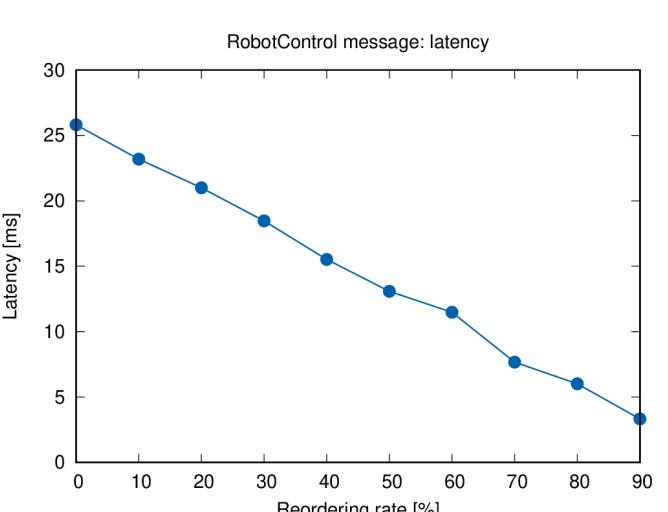
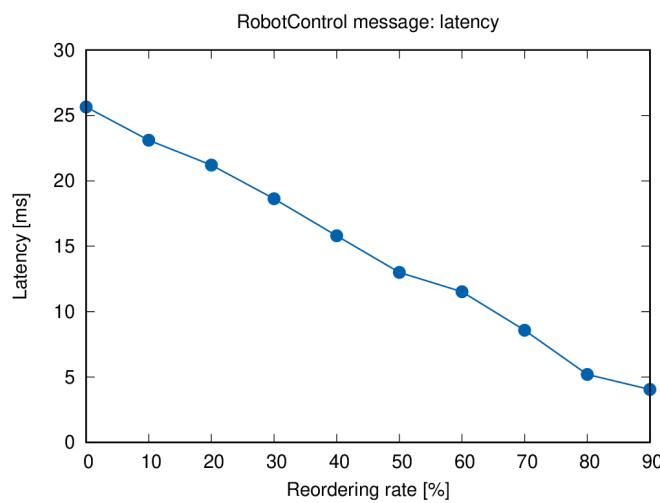
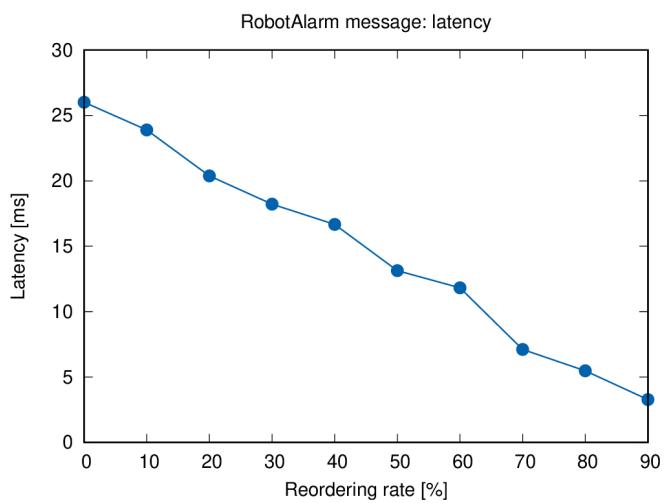


## 6. Reorder: latency

ros1



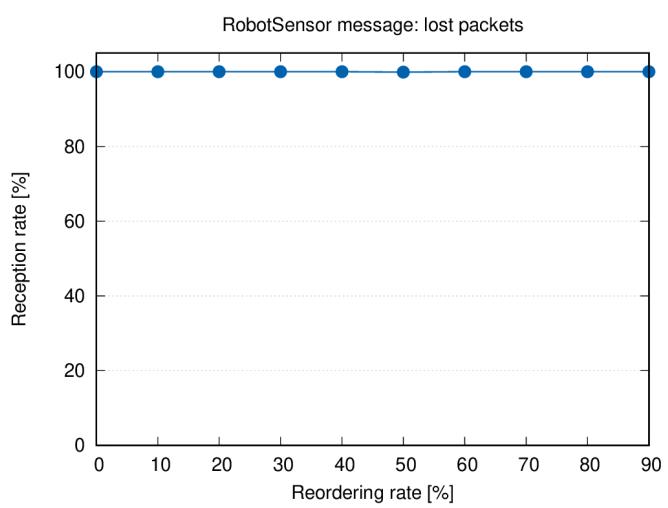
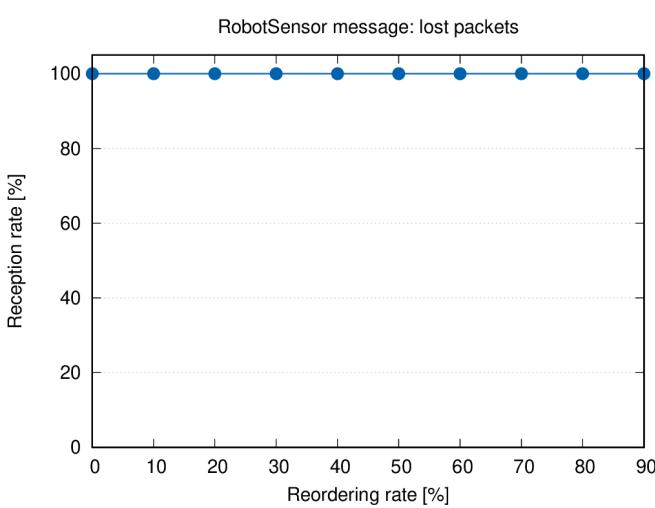
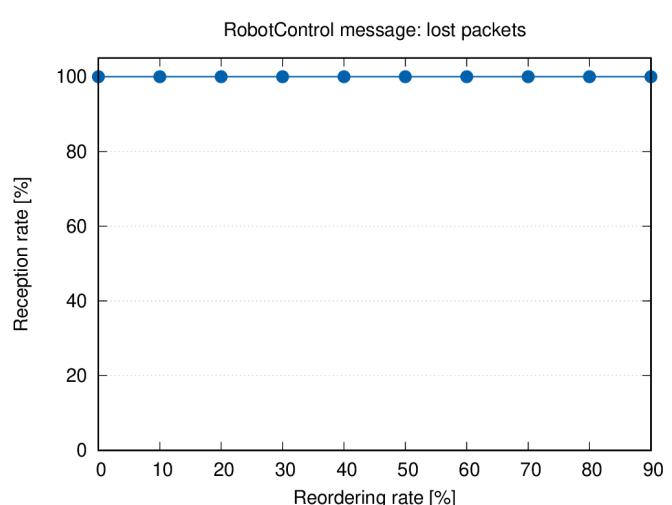
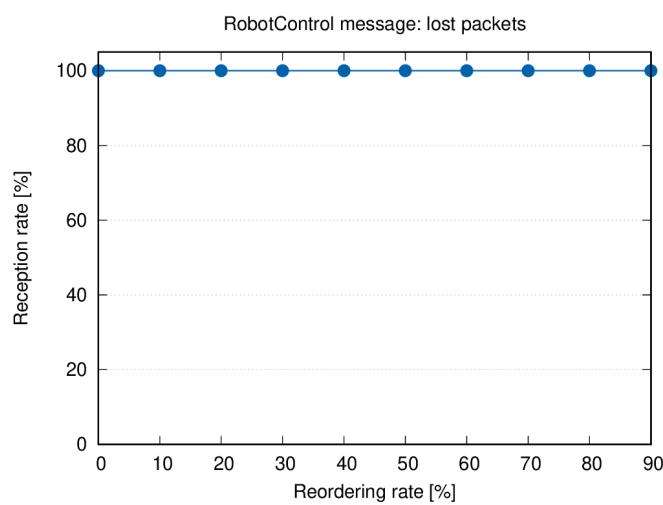
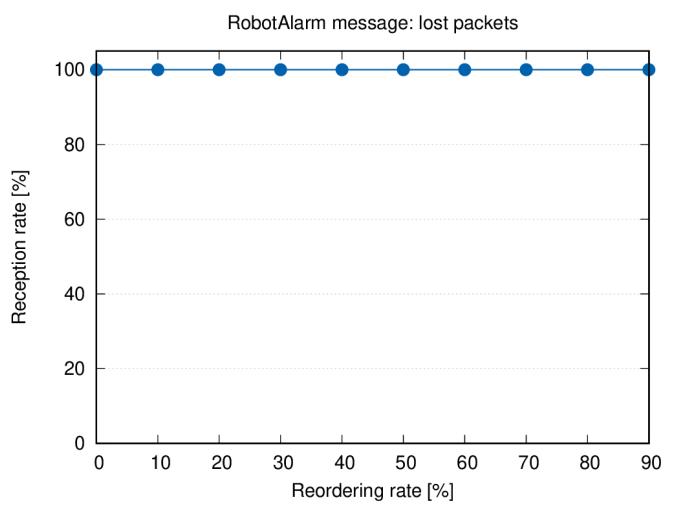
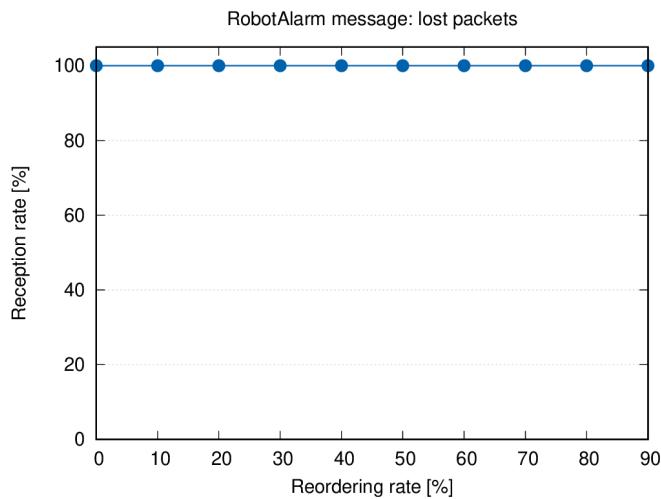
ros2:fastrtps



## 6. Reorder: lost-packets

ros1

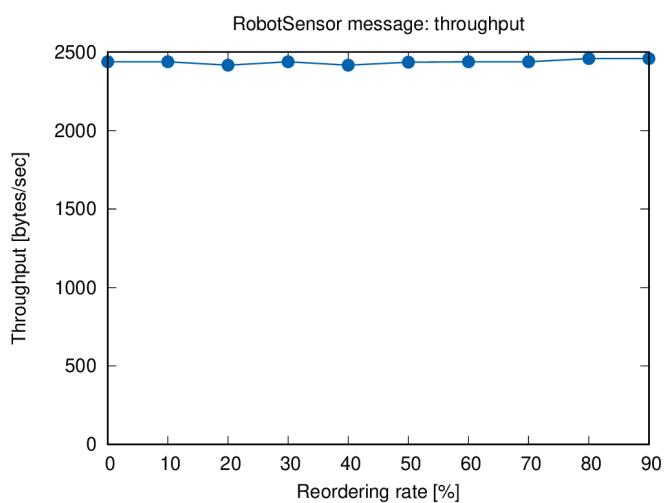
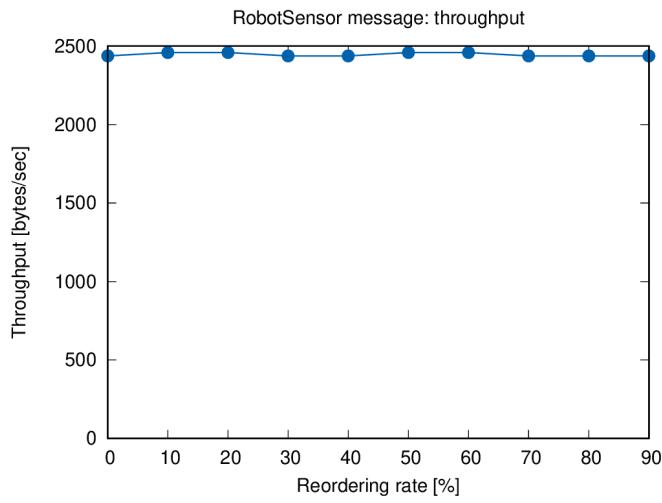
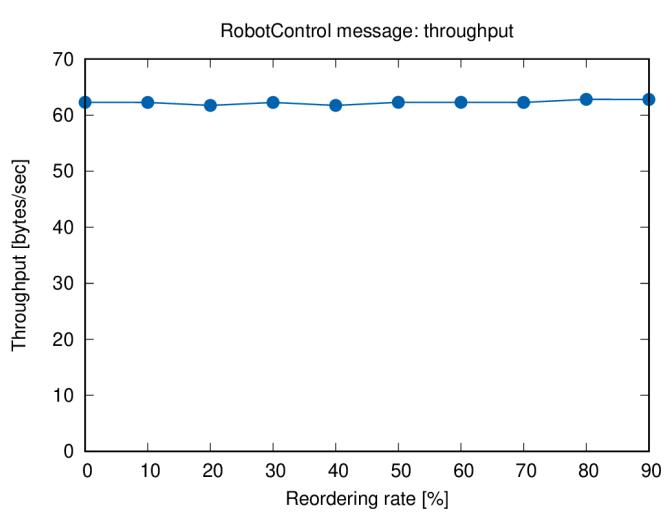
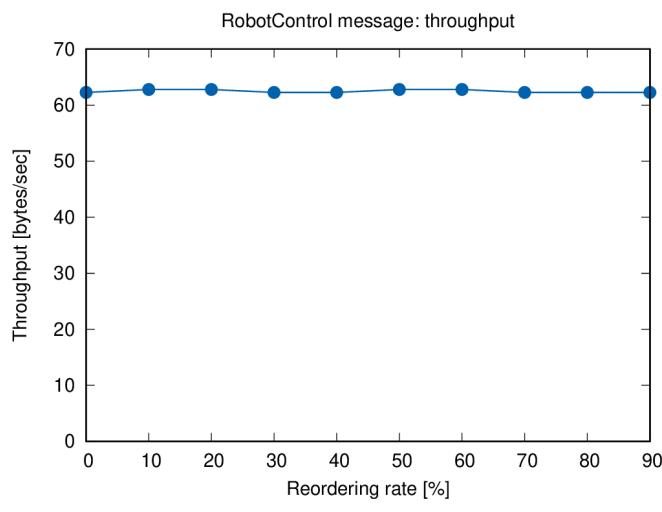
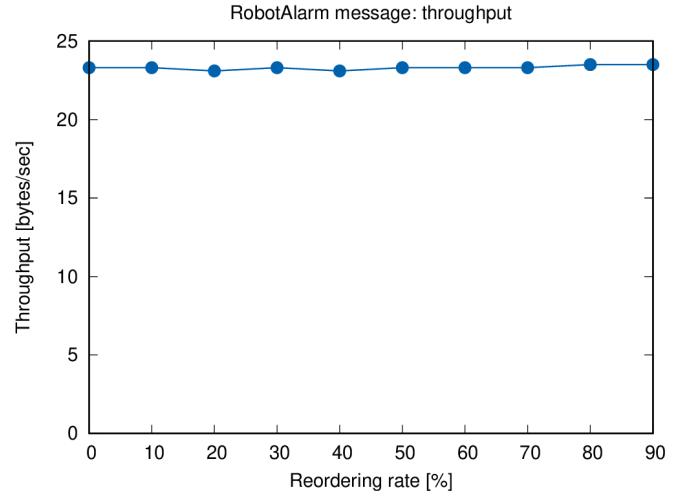
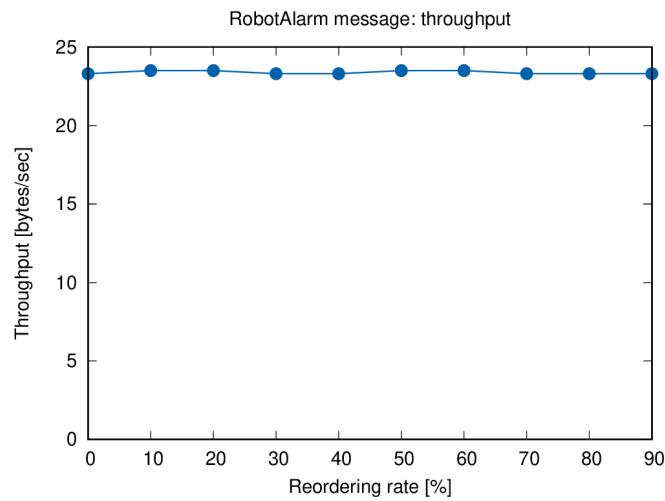
ros2:fastrtps



## 6. Reorder: throughput

ros1

ros2:fastrtps



以下篇幅为 ros2 的种 DDS 的对比:

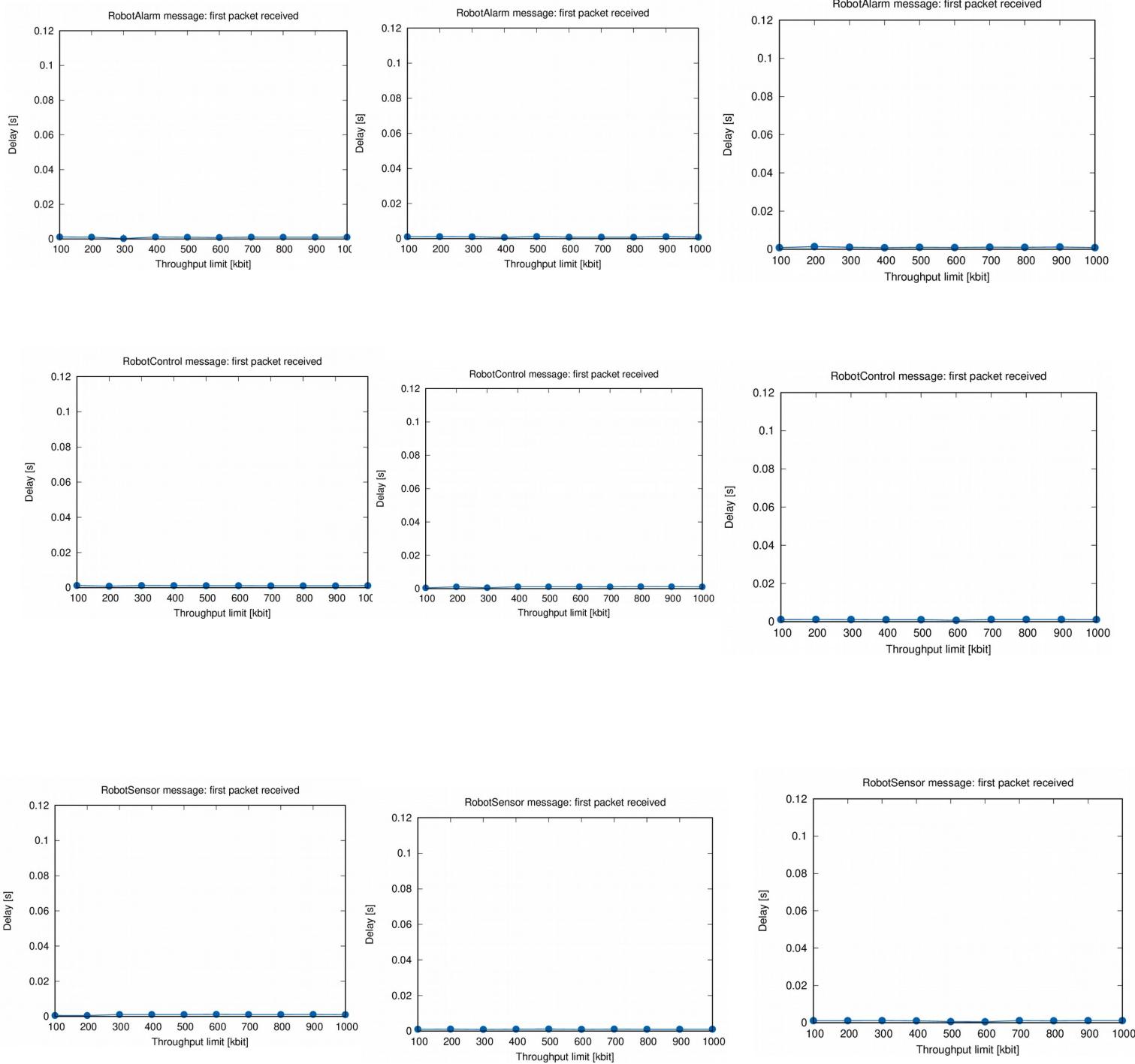
- fastrpts
- opensplice
- connext

# 1. Limit: first-received

ros2:fastrtps

ros2:opensplice

ros2:connext

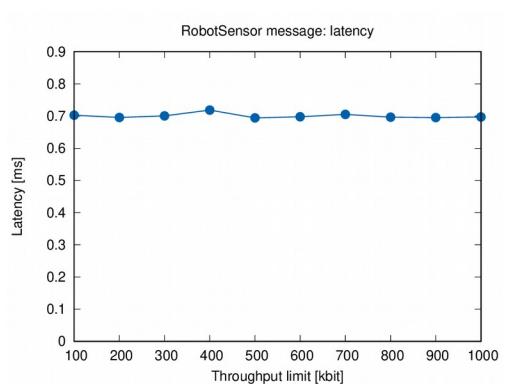
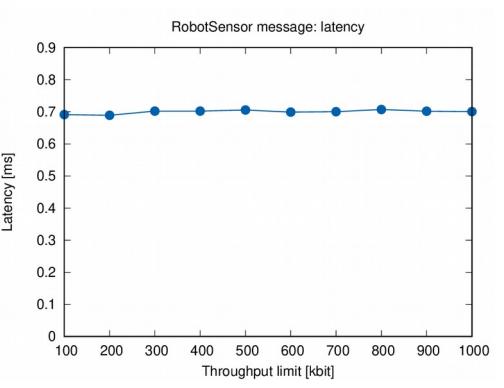
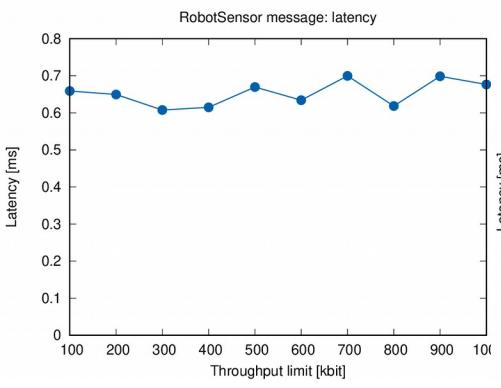
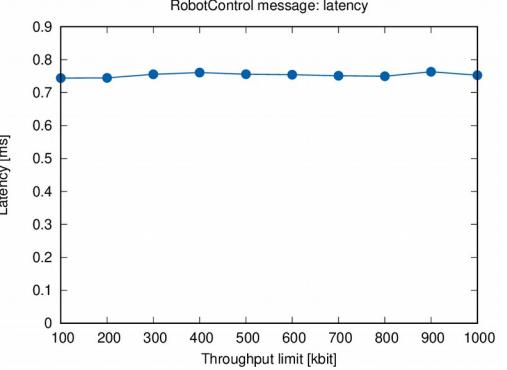
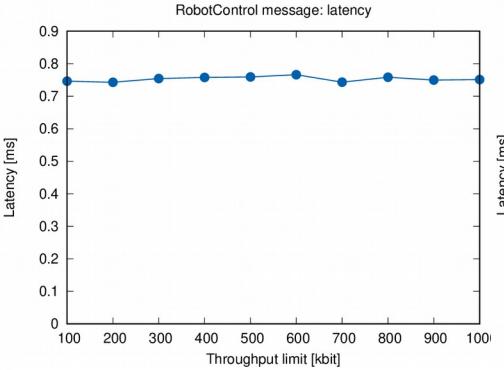
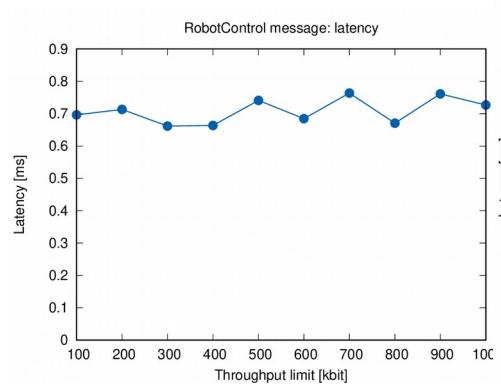
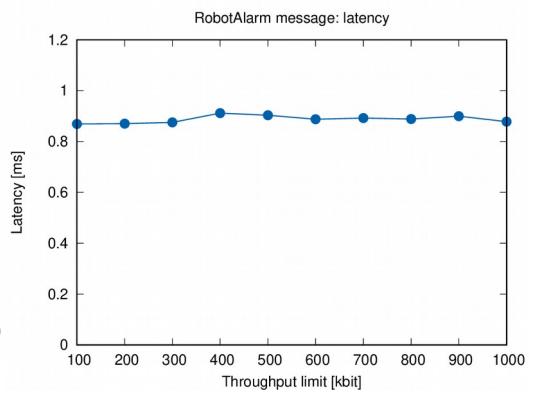
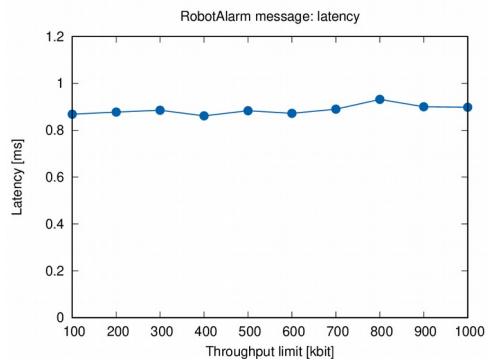
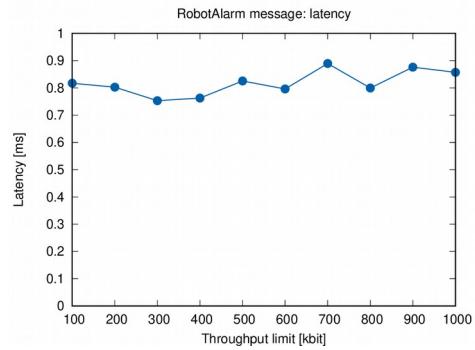


# 1. Limit:latency

ros2:fastrtps

ros2:opensplice

ros2:connext

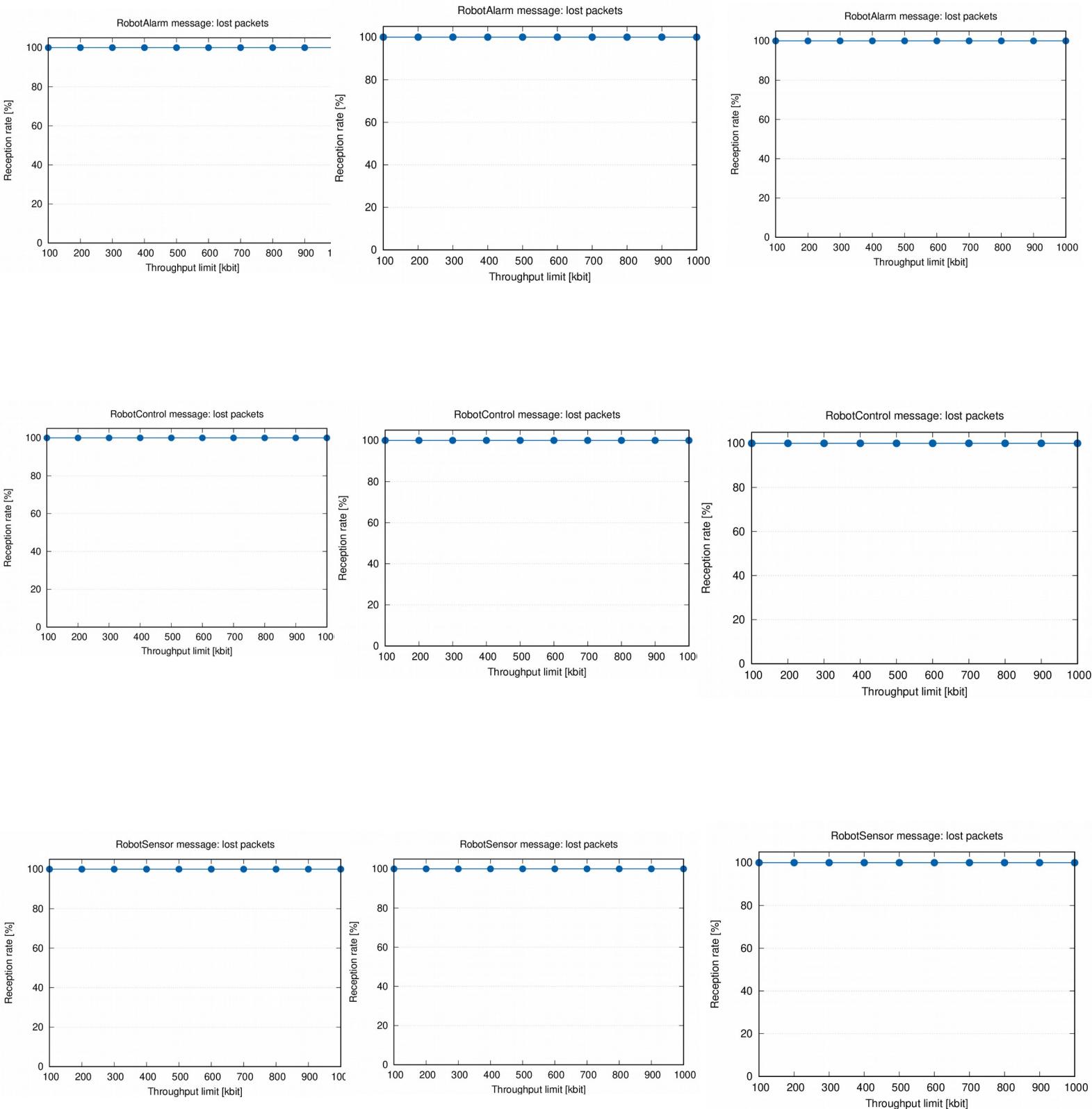


# 1. Limit:lost-packets

ros2:fastrtps

ros2:opensplice

ros2:connext

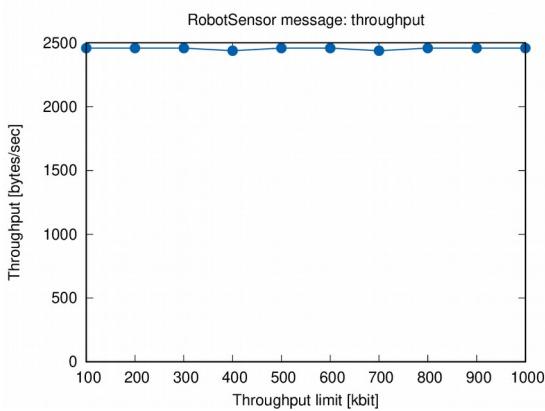
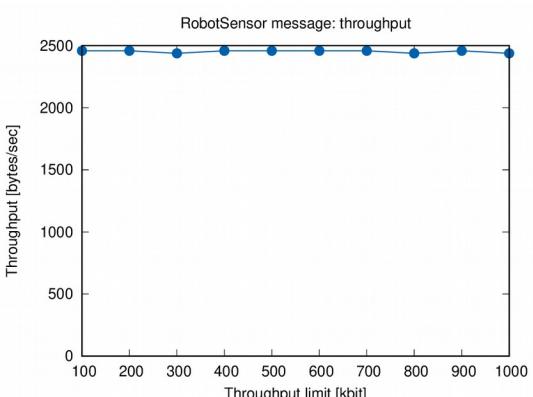
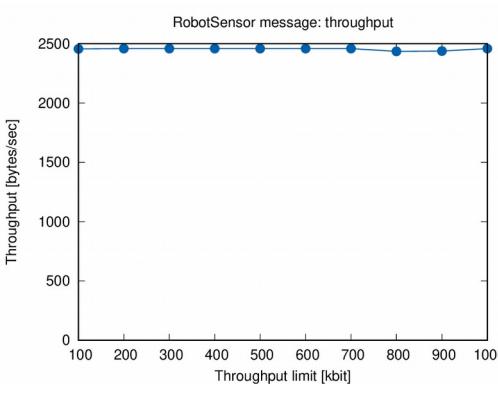
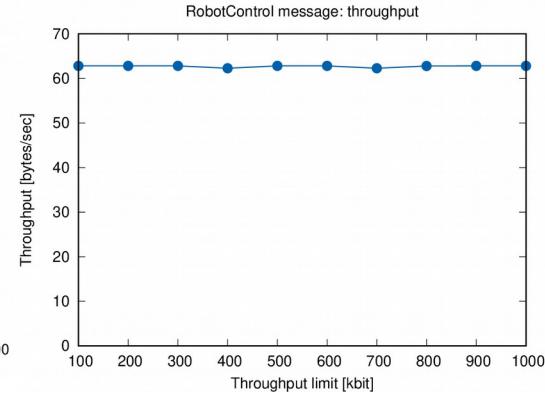
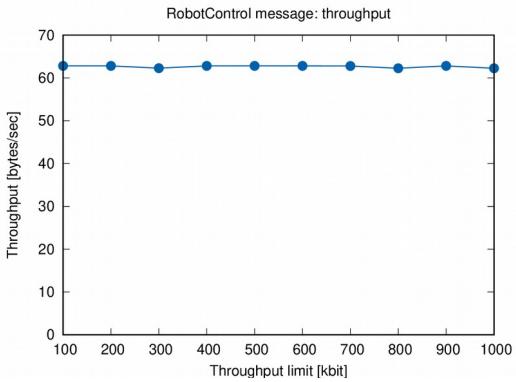
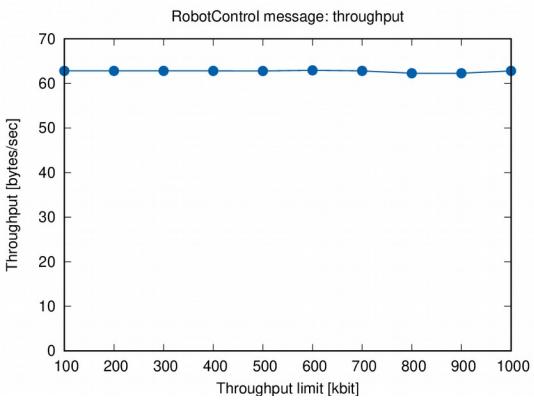
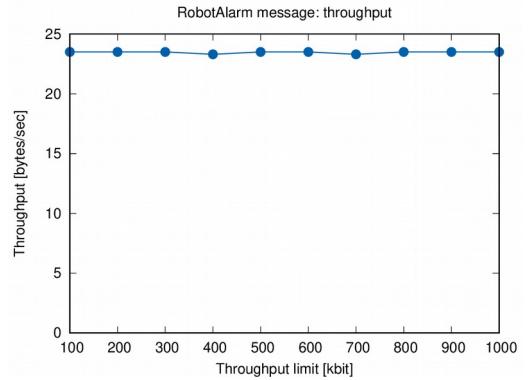
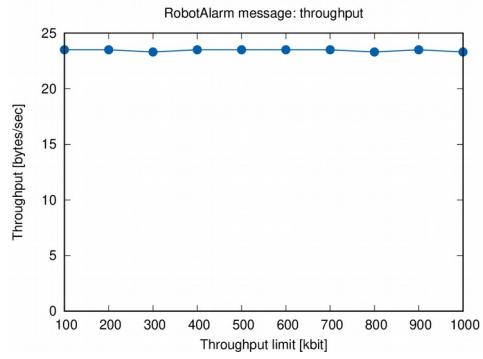
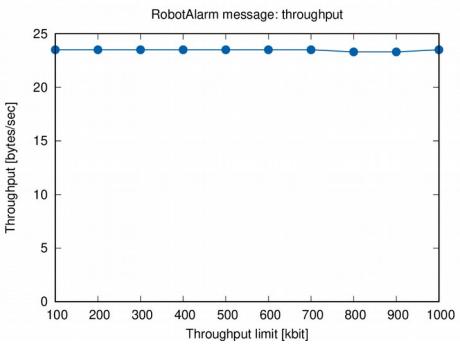


# 1. Limit:throughput

ros2:fastrtps

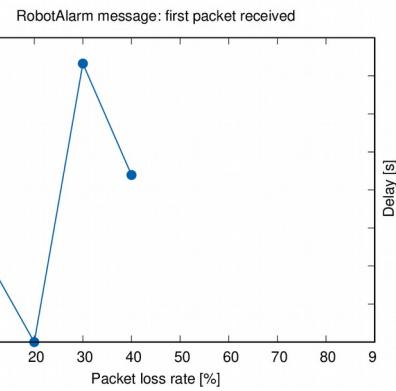
ros2:opensplice

ros2:connext

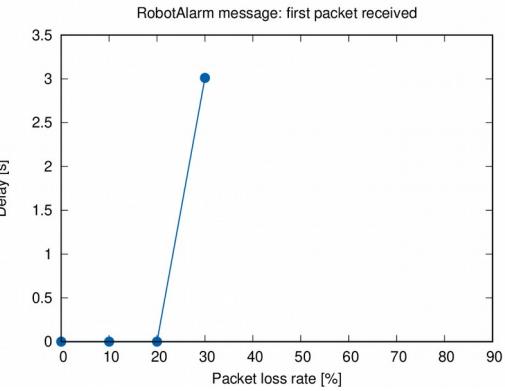


## 2. Loss: first-received ros2:fastrtps

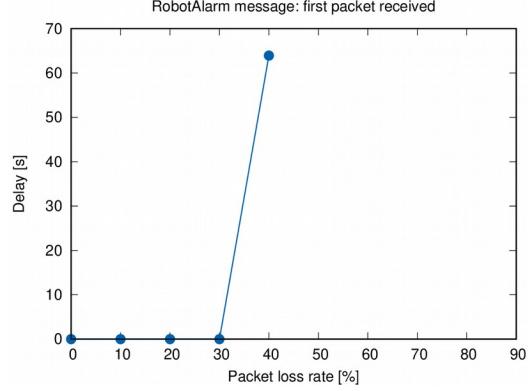
RobotAlarm message: first packet received



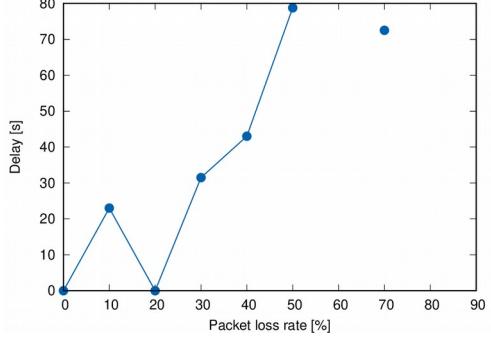
RobotControl message: first packet received



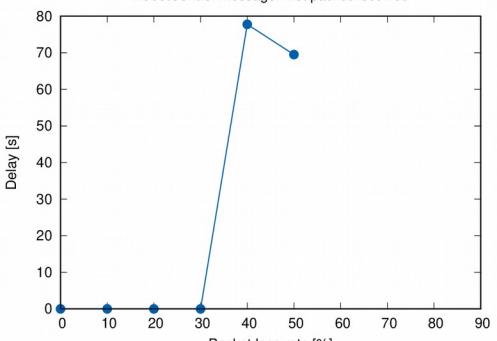
RobotSensor message: first packet received



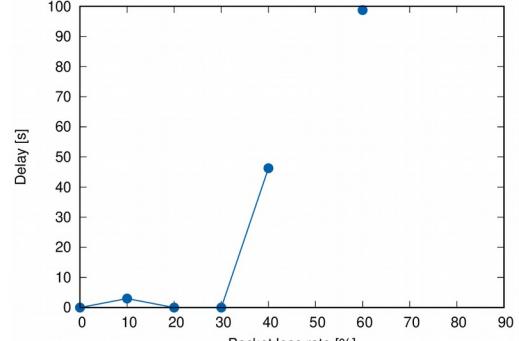
RobotControl message: first packet received



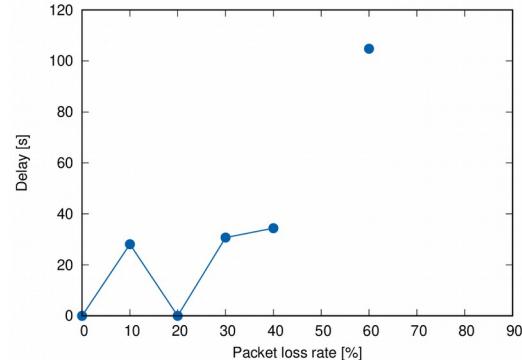
RobotControl message: first packet received



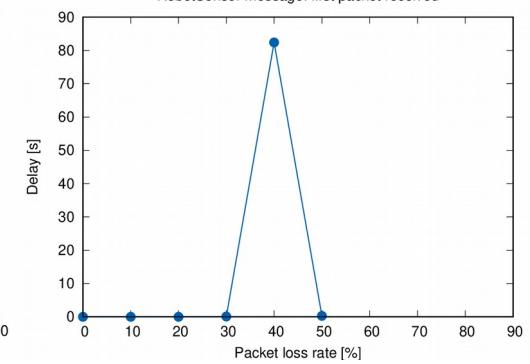
RobotControl message: first packet received



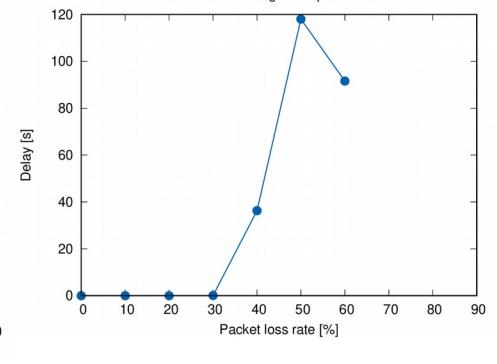
RobotSensor message: first packet received



RobotSensor message: first packet received



RobotSensor message: first packet received

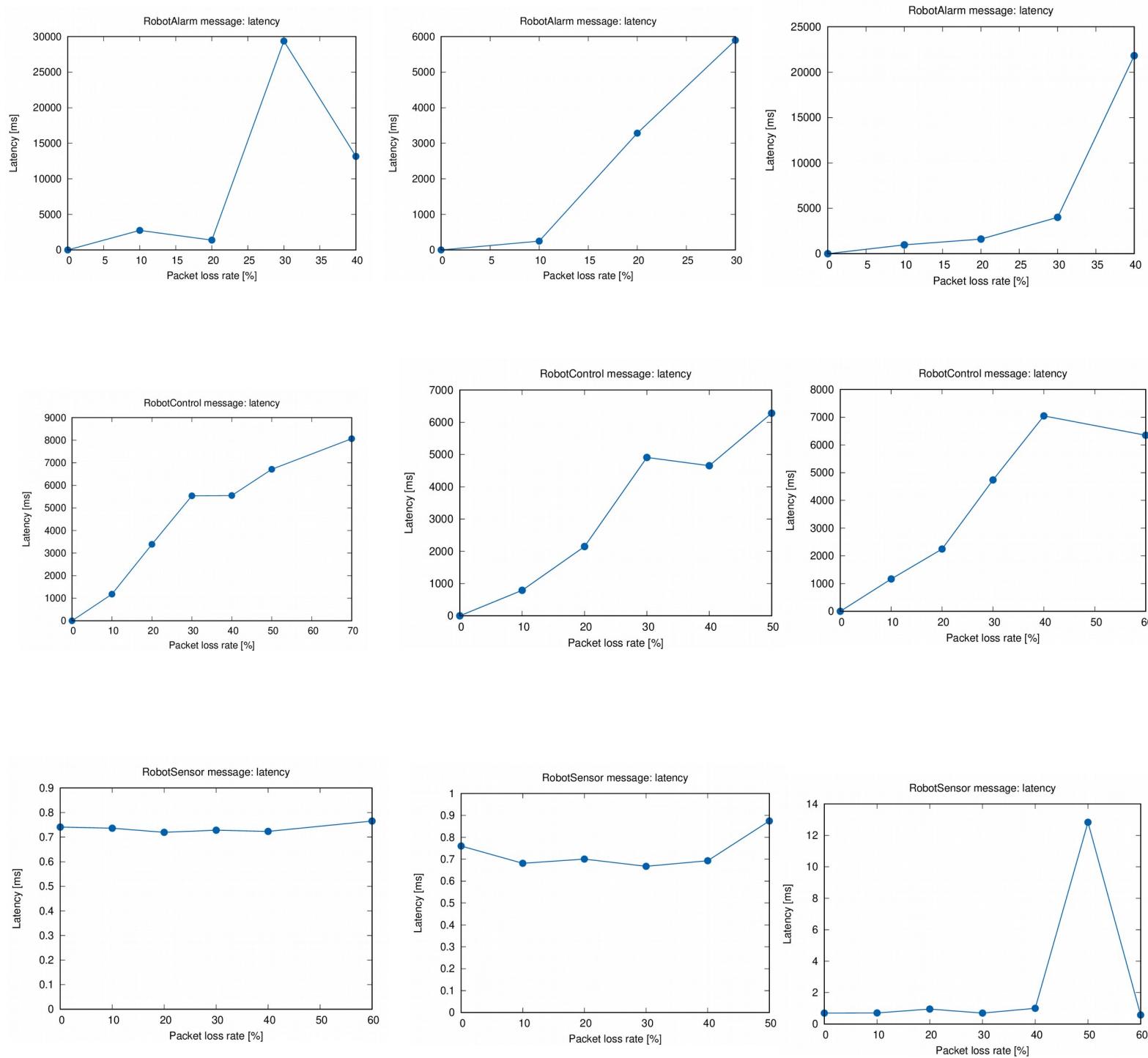


## 2. Loss: latency

ros2:fastrtps

ros2:opensplice

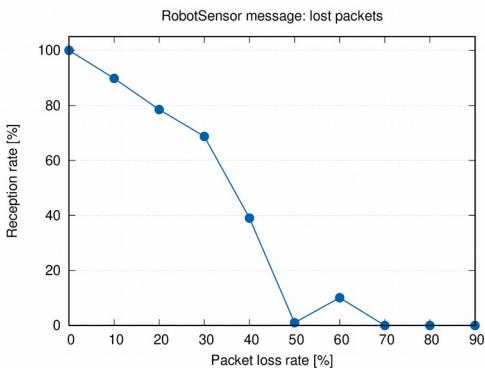
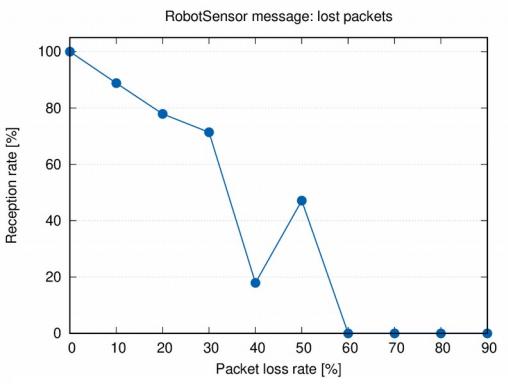
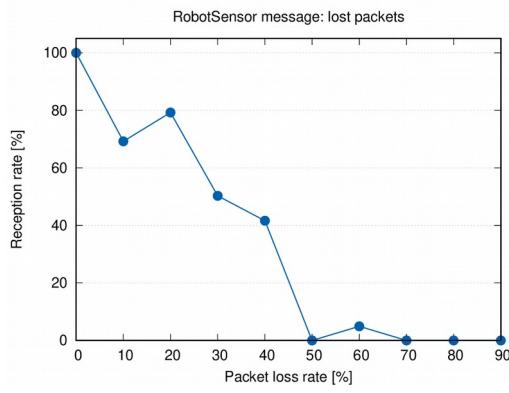
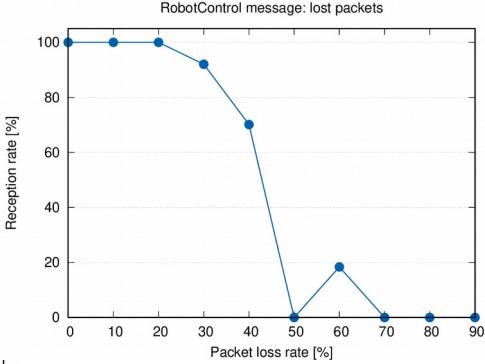
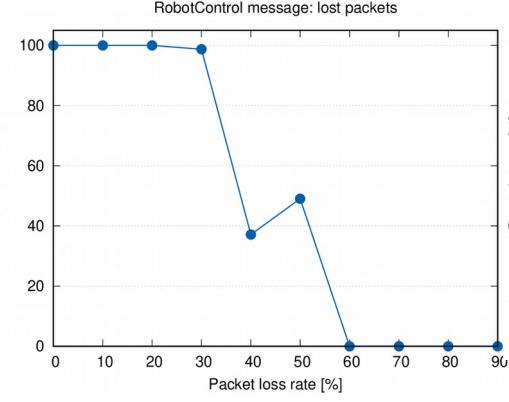
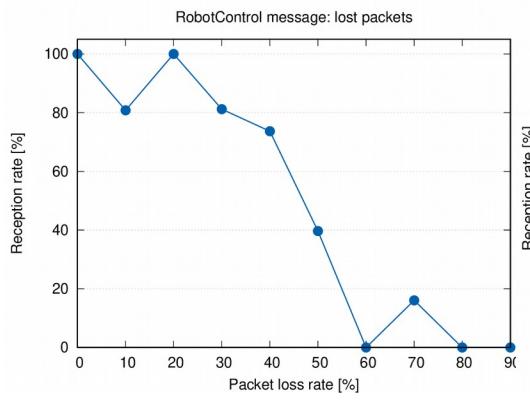
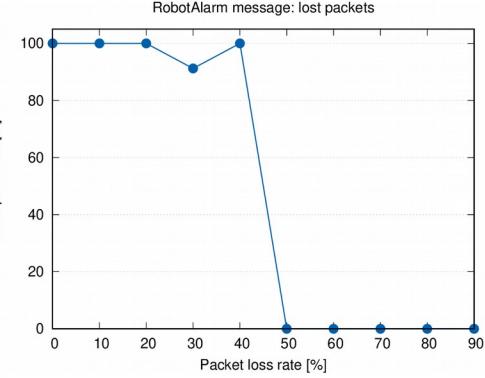
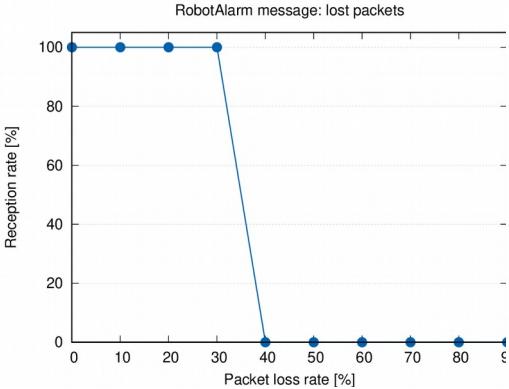
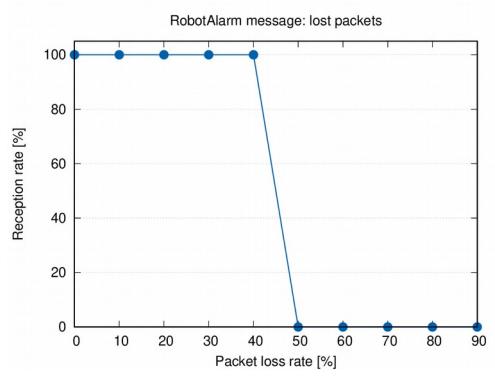
ros2:connext



## 2. Loss:lost-packets ros2:fastrtps

ros2:opensplice

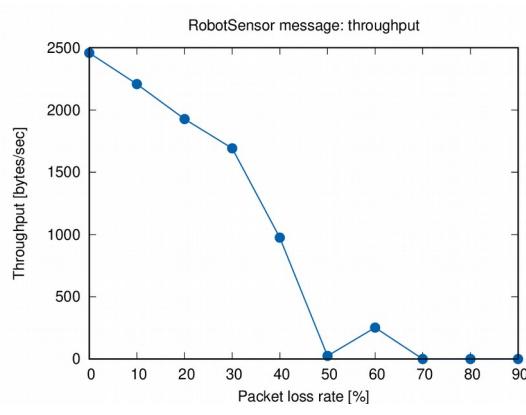
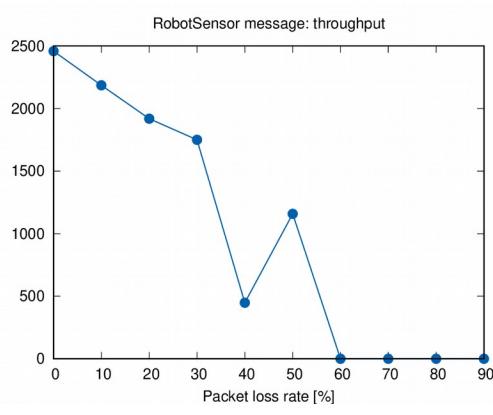
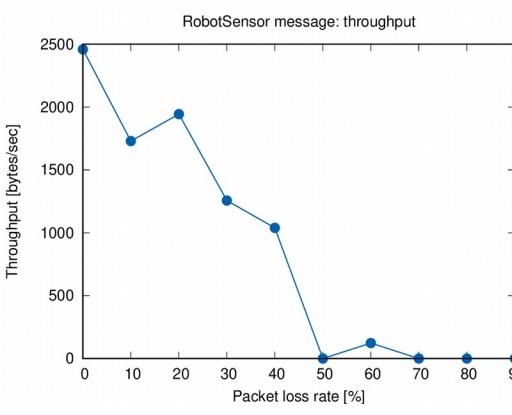
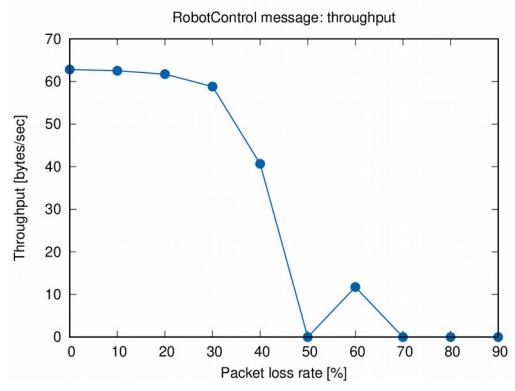
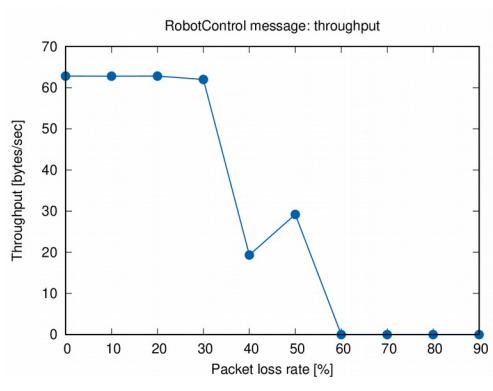
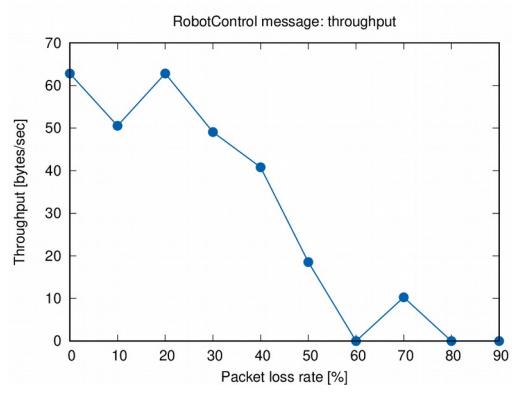
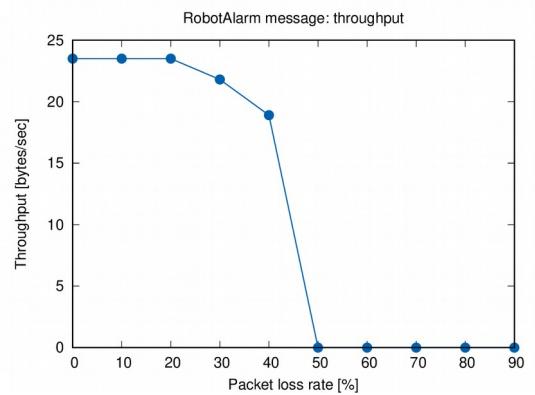
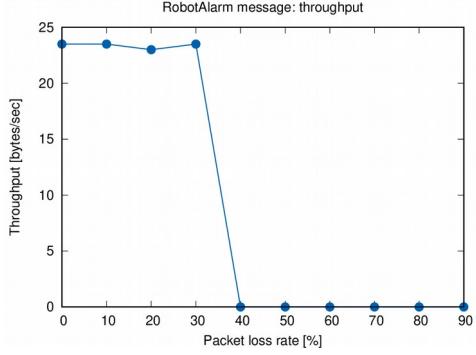
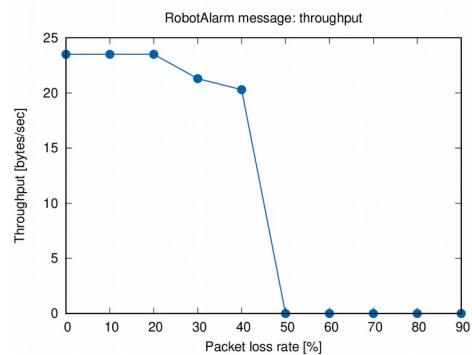
ros2:connext



## 2. Loss:throughput ros2:fastrtps

ros2:opensplice

ros2:connect

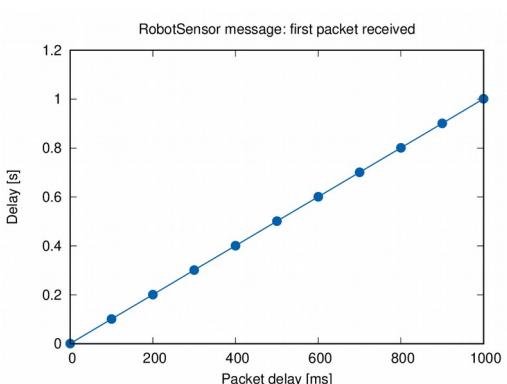
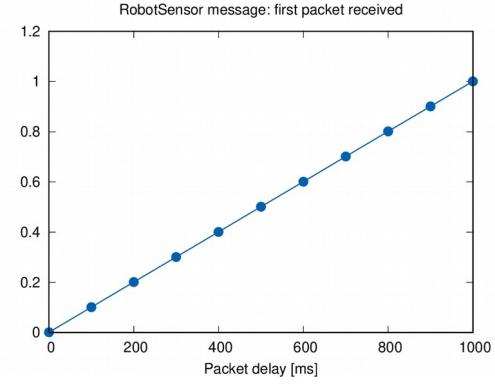
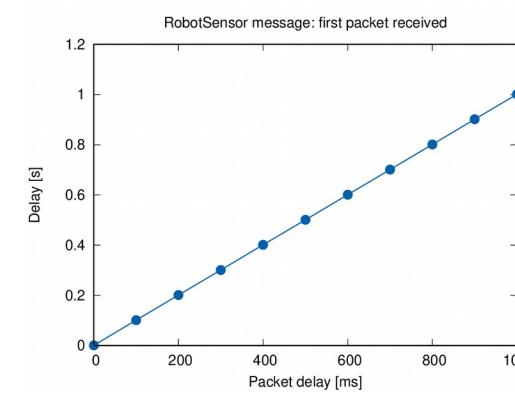
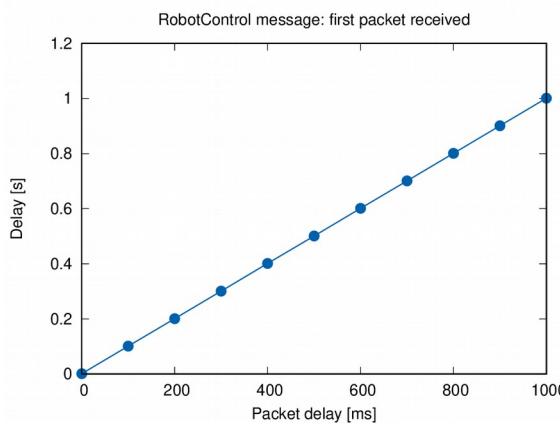
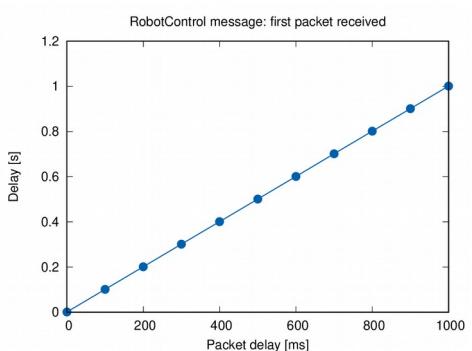
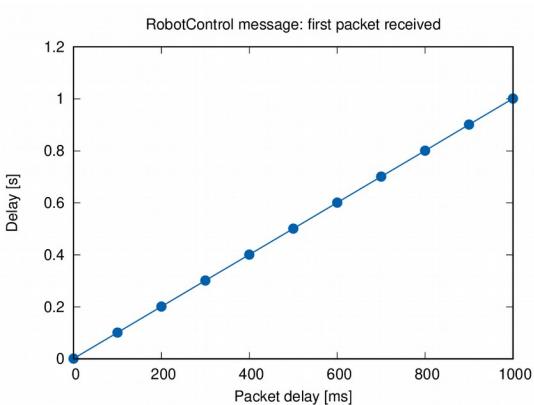
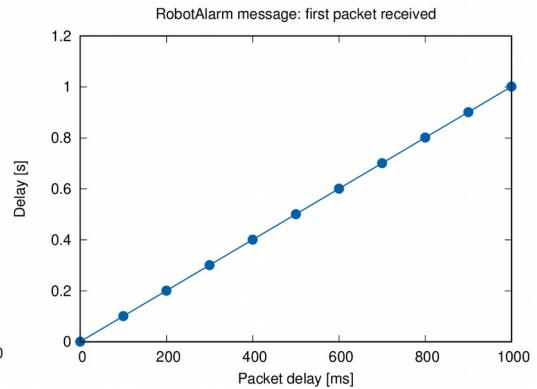
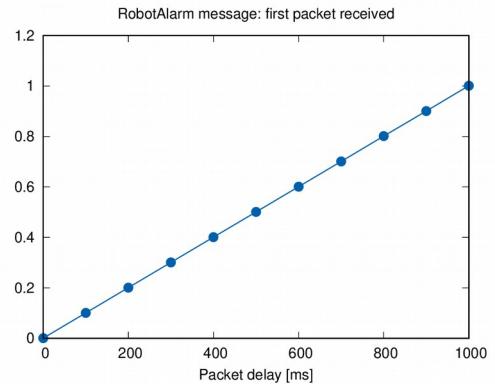
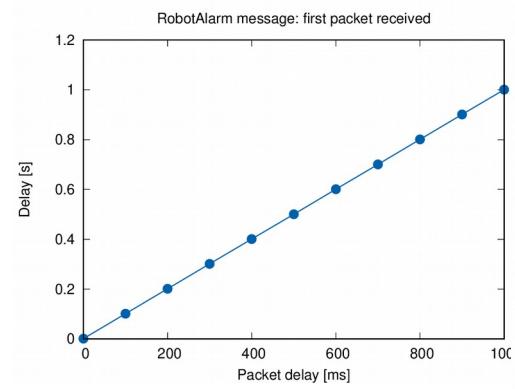


### 3. Delay:first-received

# ros2:fastrtps

# ros2:opensplice

ros2:connect

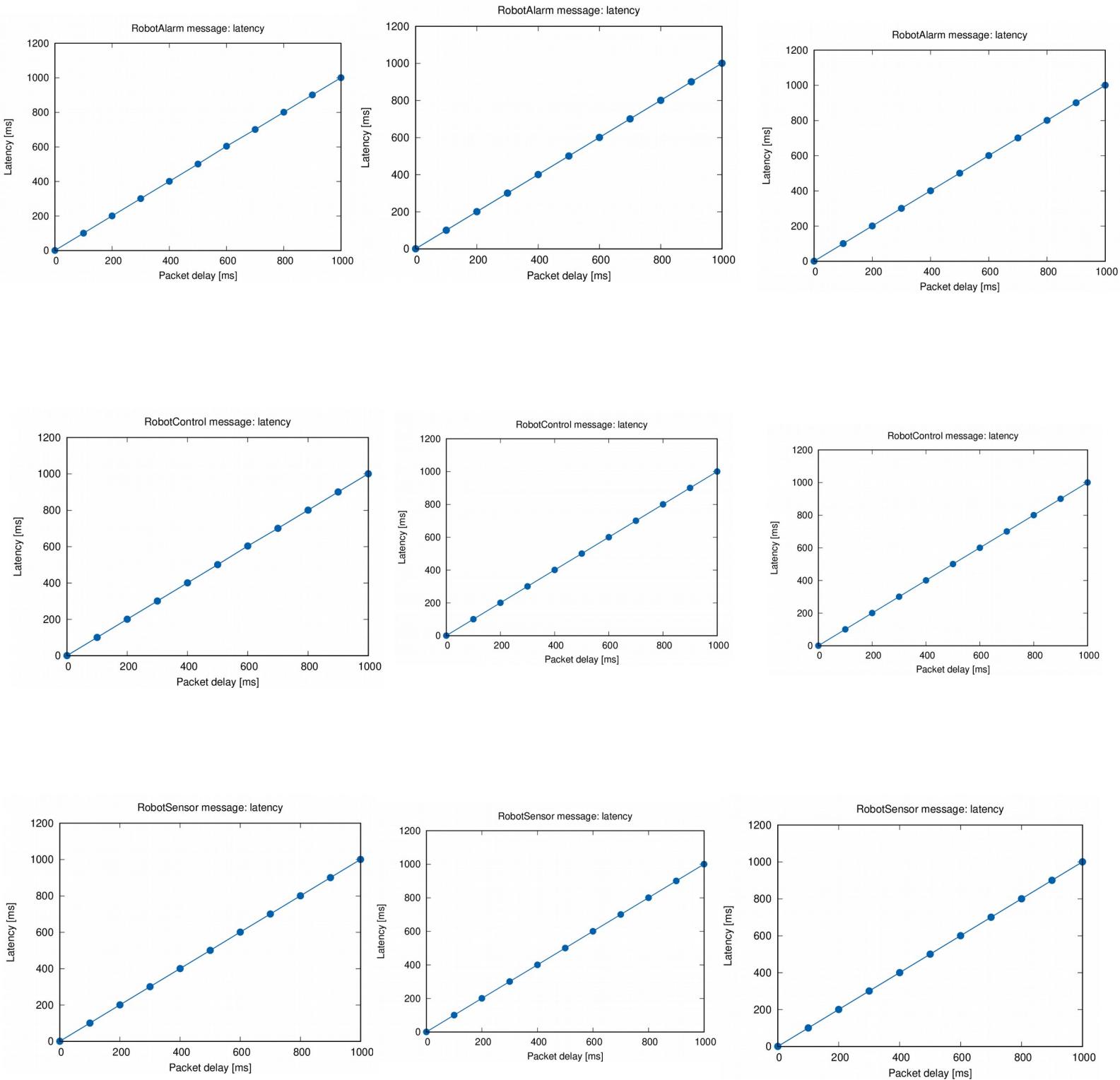


### 3. Delay:latency

ros2:fastrtps

ros2:opensplice

ros2:connext

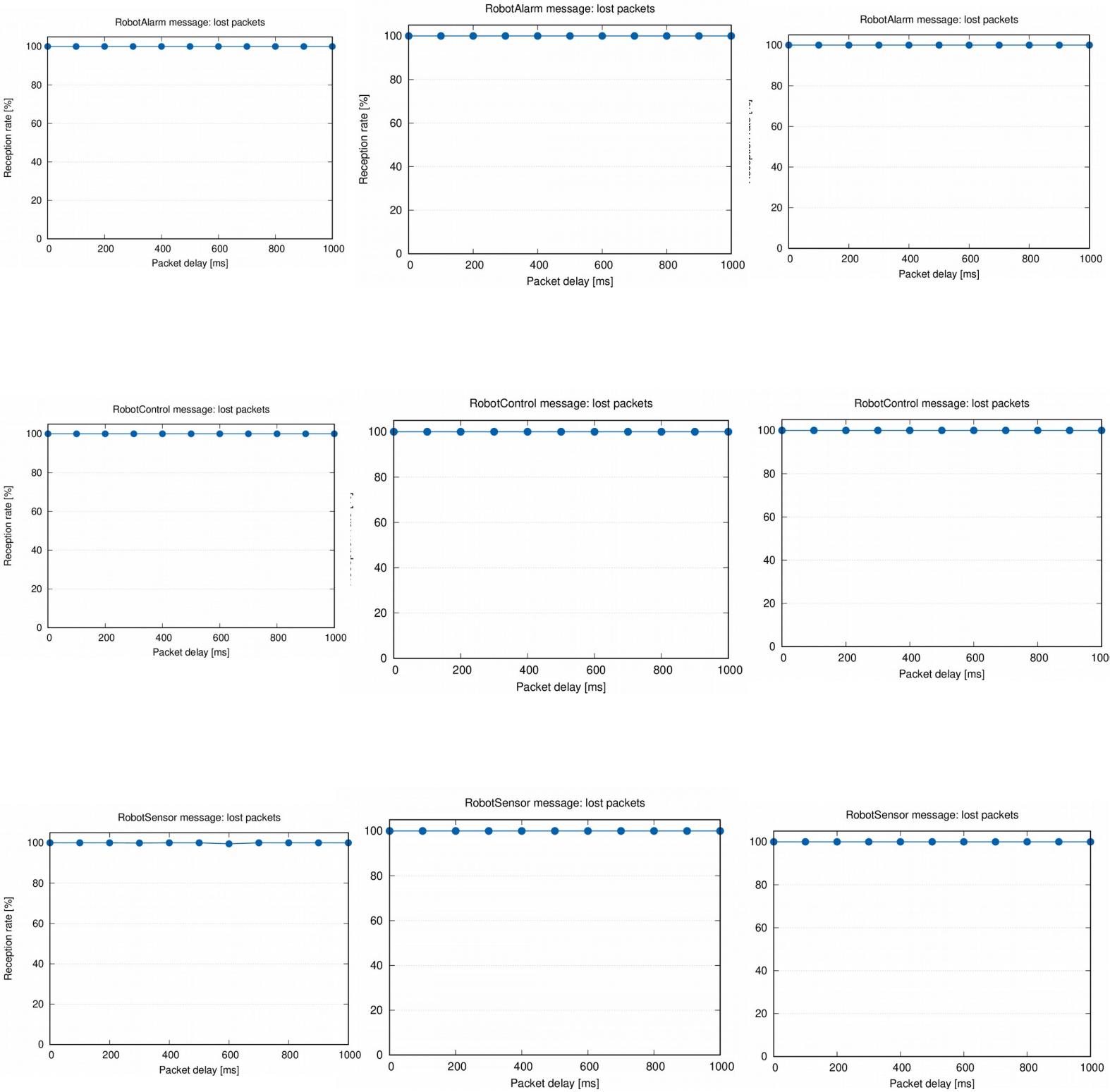


### 3. Delay:lost-packets

ros2:fastrtps

ros2:opensplice

ros2:connext

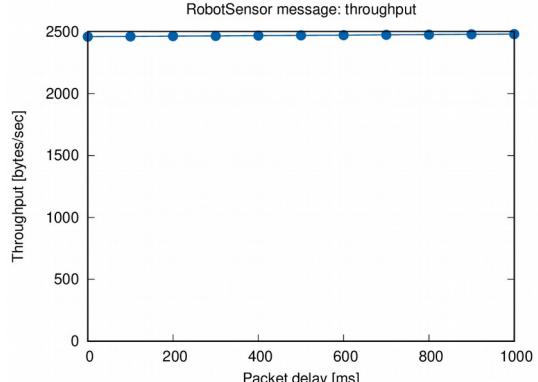
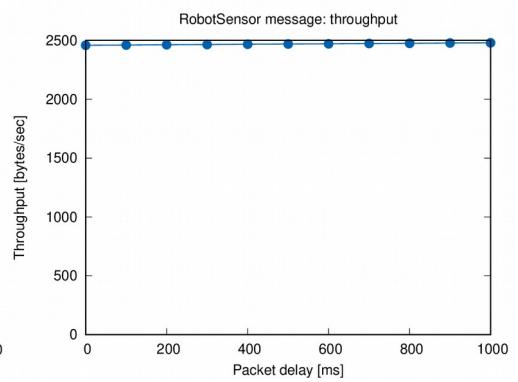
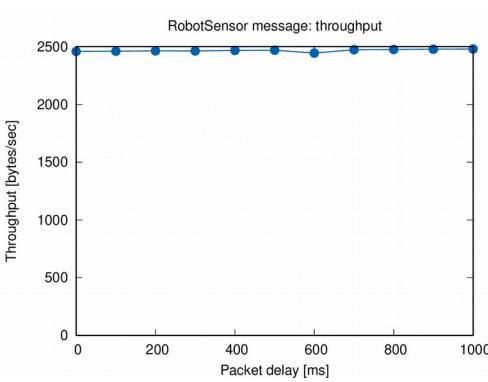
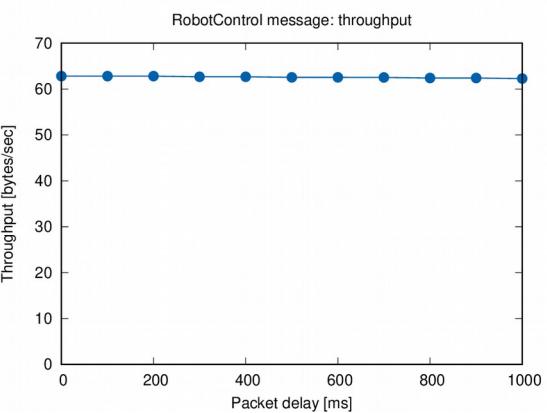
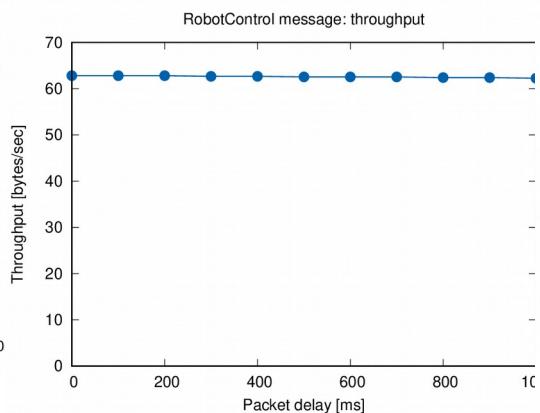
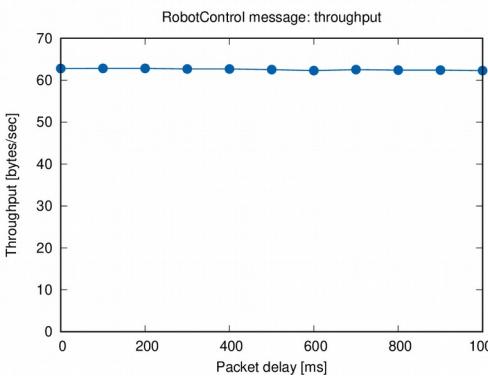
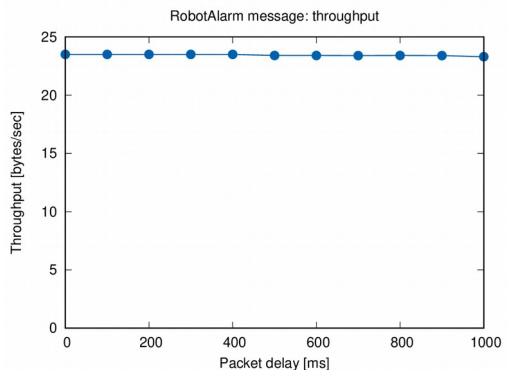
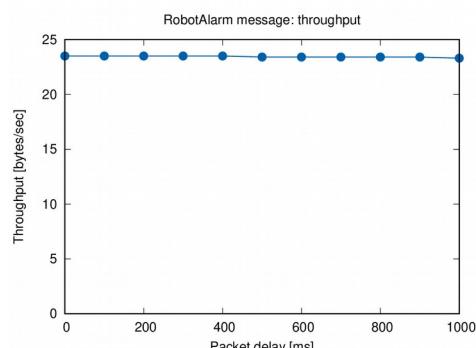
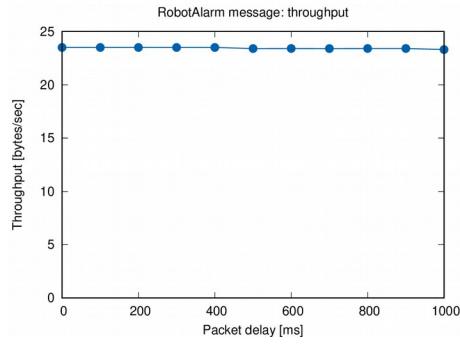


### 3. Delay:throughput

ros2:fastrtps

ros2:opensplice

ros2:connext

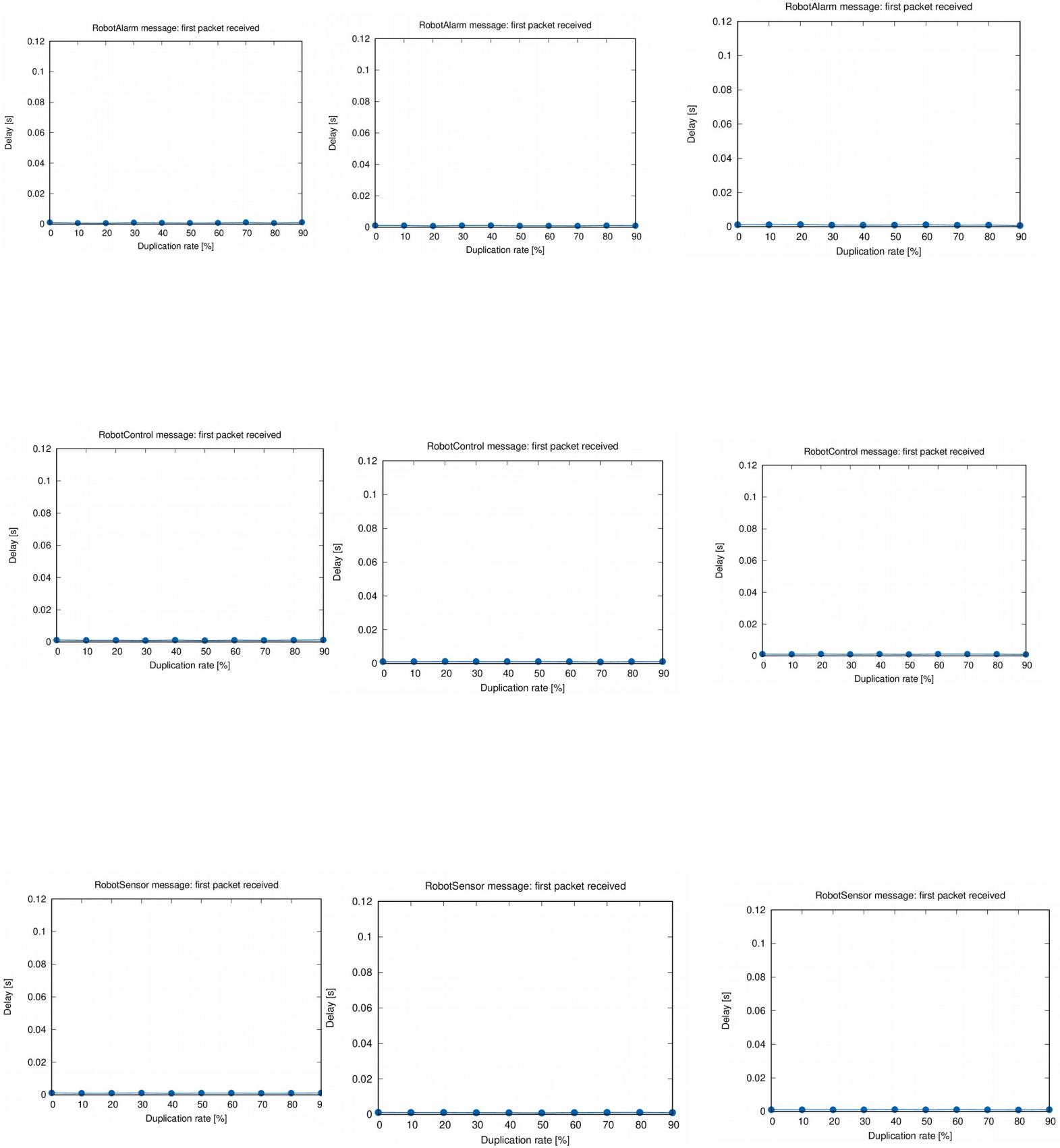


## 4. Duplication:first-received

ros2:fastrtps

ros2:opensplice

ros2:connext

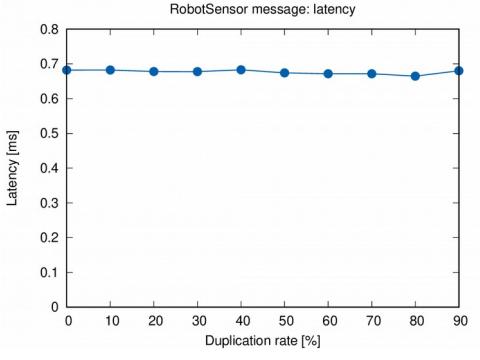
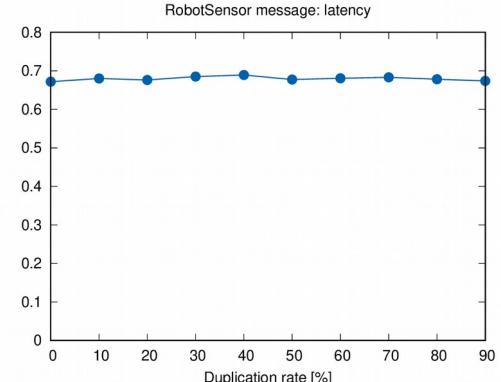
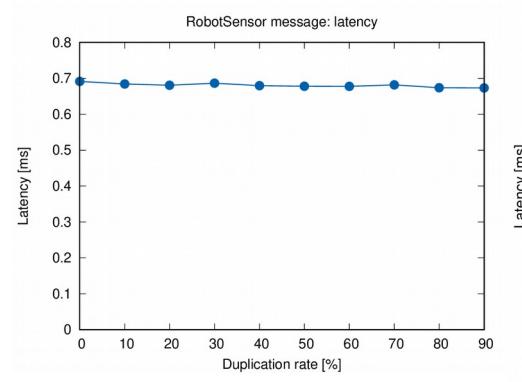
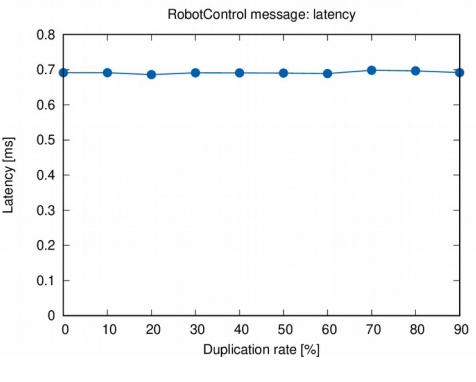
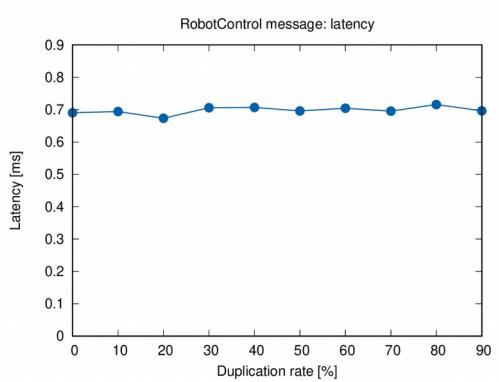
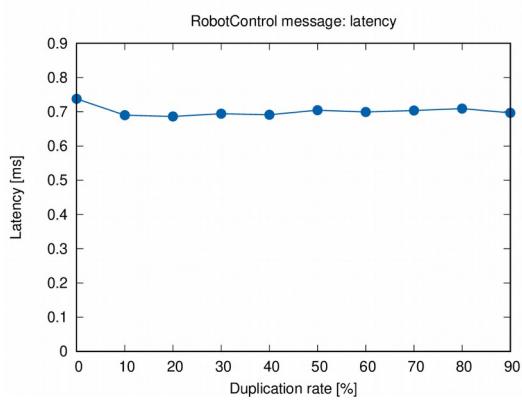
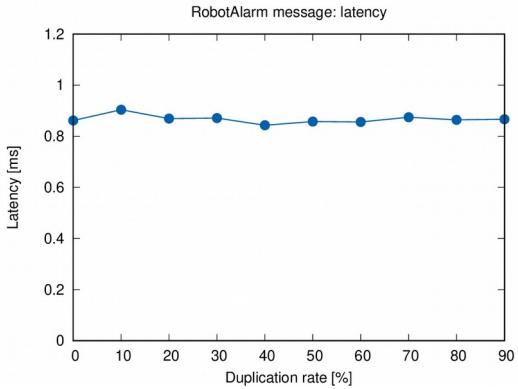
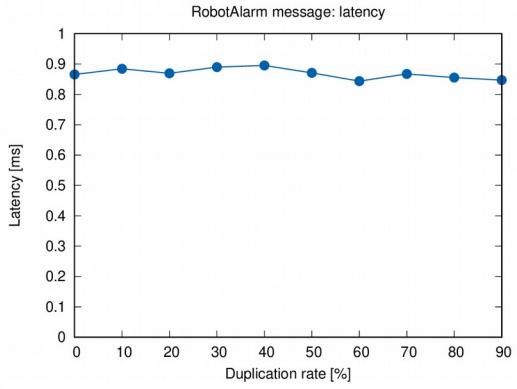
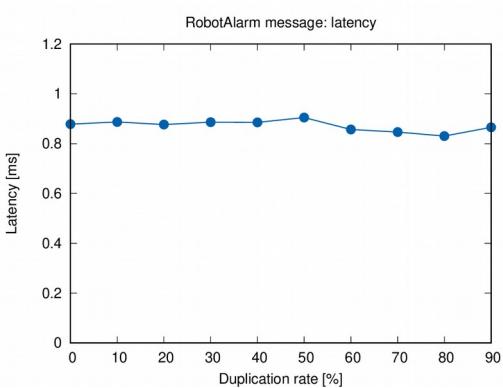


## 4. Duplication:latency

ros2:fastrtps

ros2:opensplice

ros2:connext

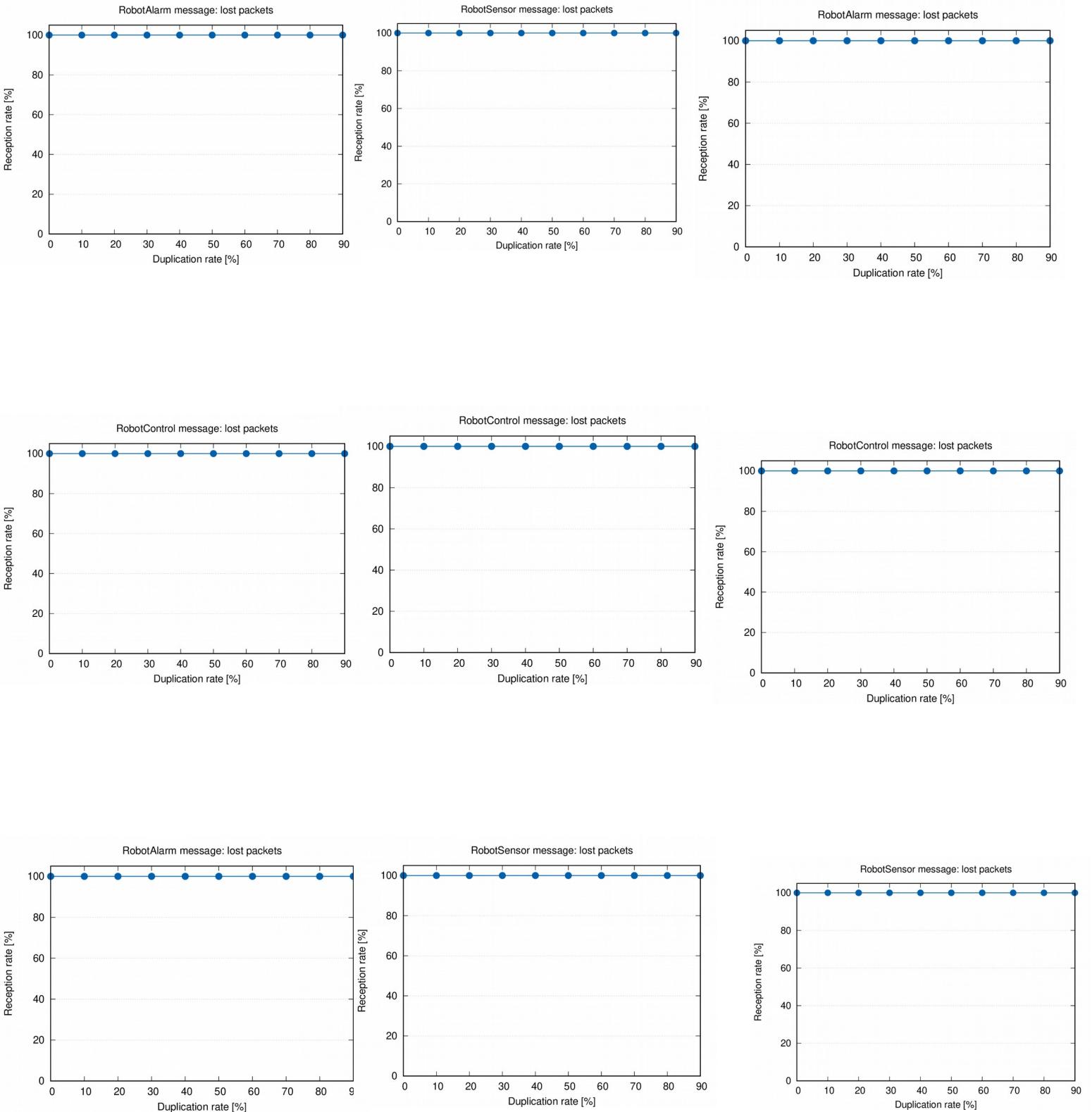


## 4. Duplication:lost-packets

ros2:fastrtps

ros2:opensplice

ros2:connext

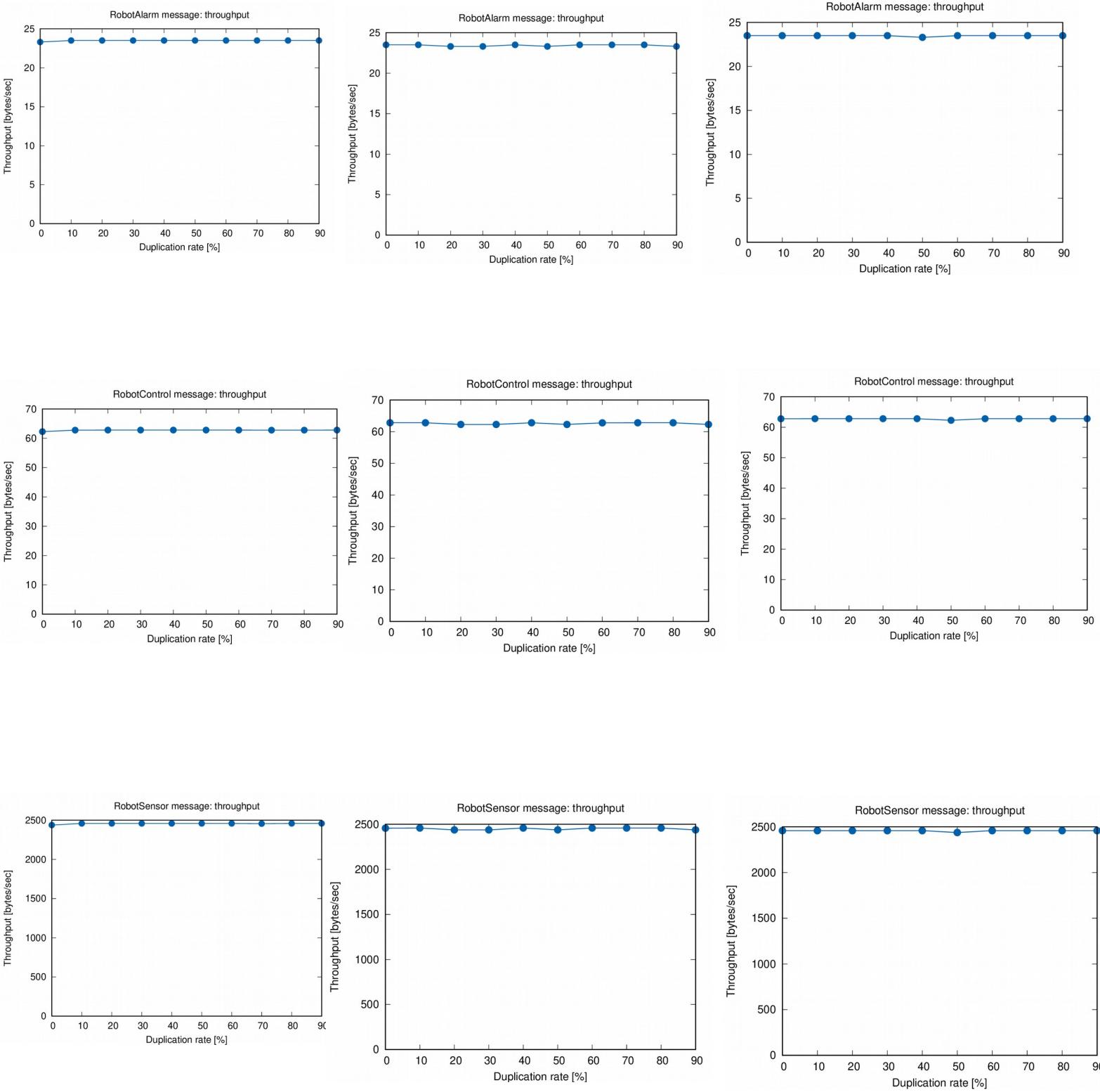


## 4. Duplication:throughput

ros2:fastrtps

ros2:opensplice

ros2:connext

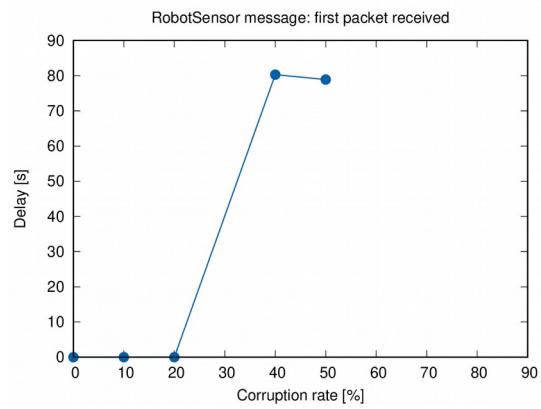
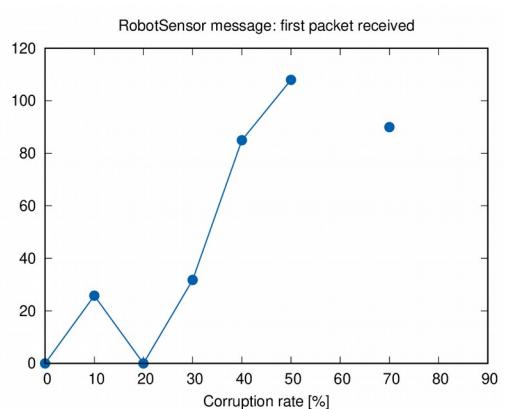
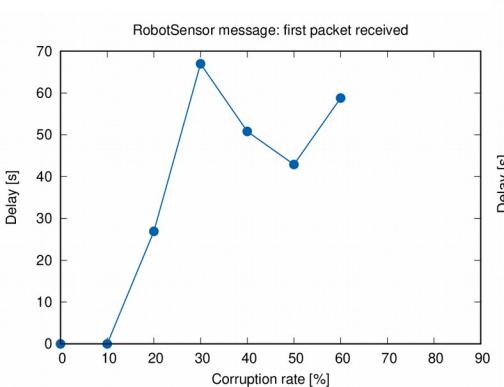
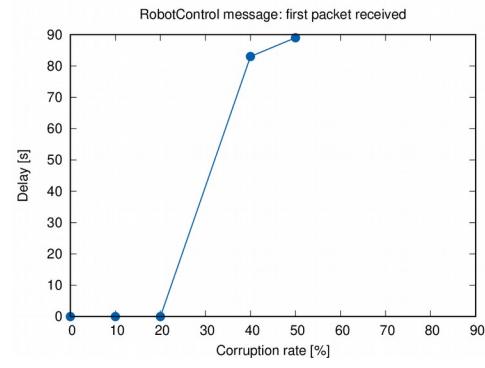
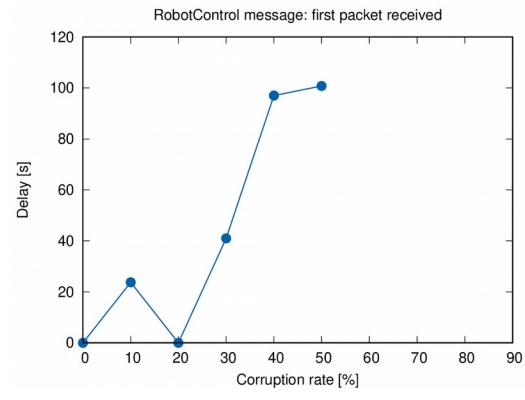
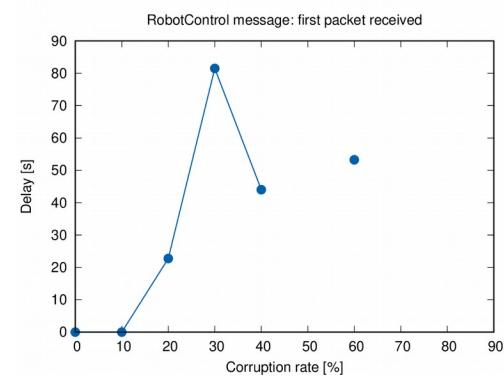
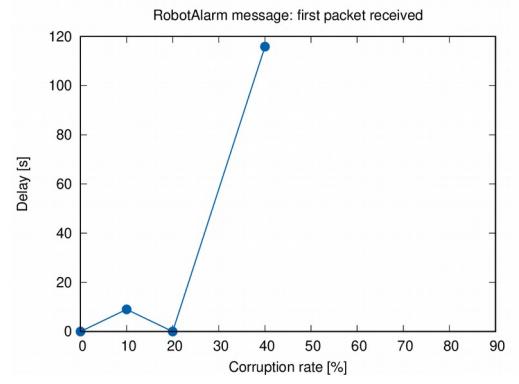
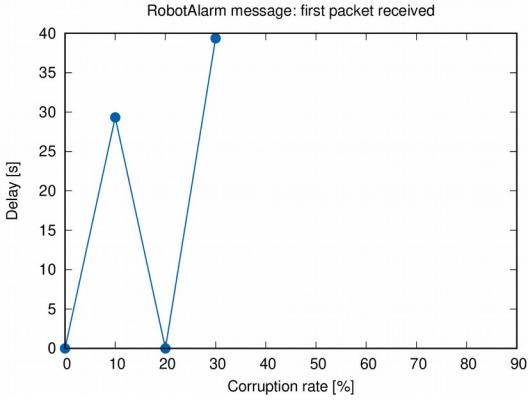
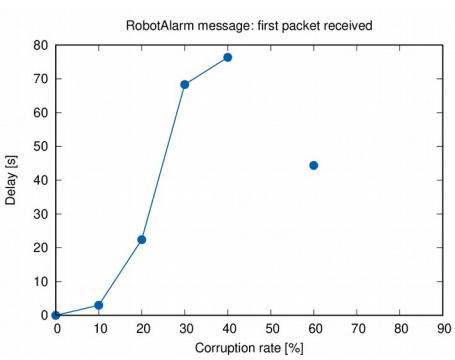


## 5. Corruption: first-received

ros2:fastrtps

ros2:opensplice

ros2:connext

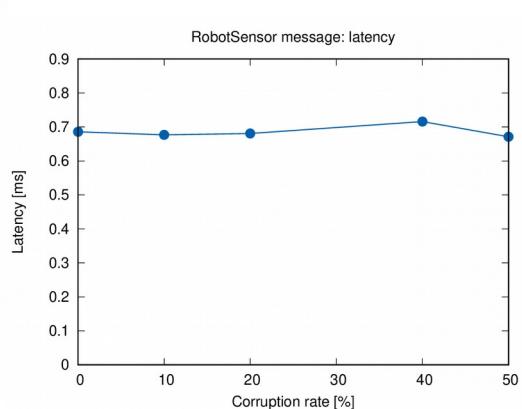
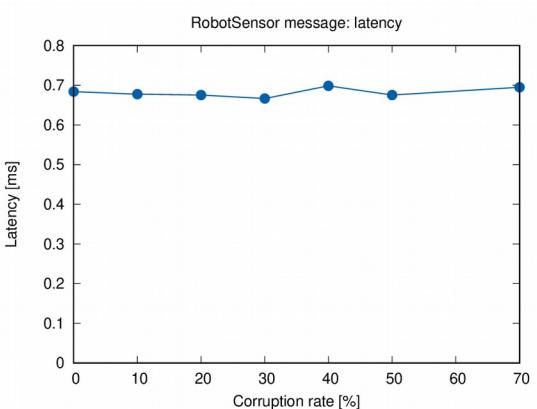
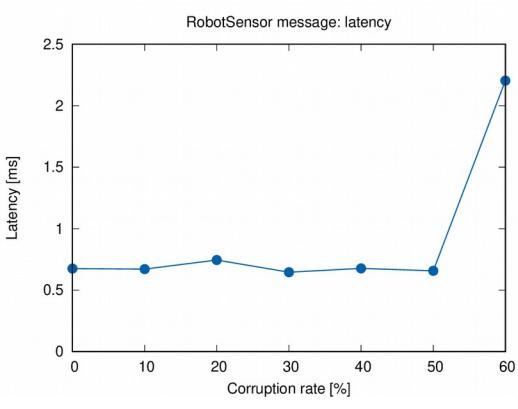
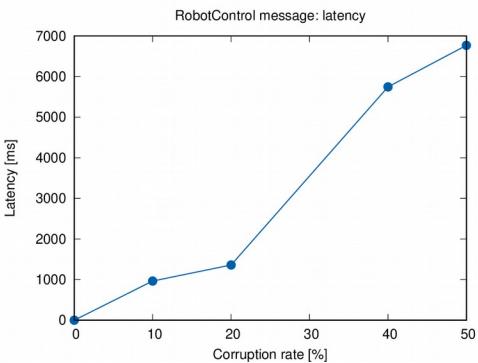
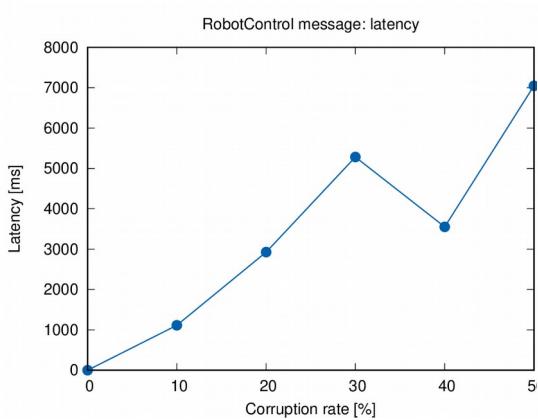
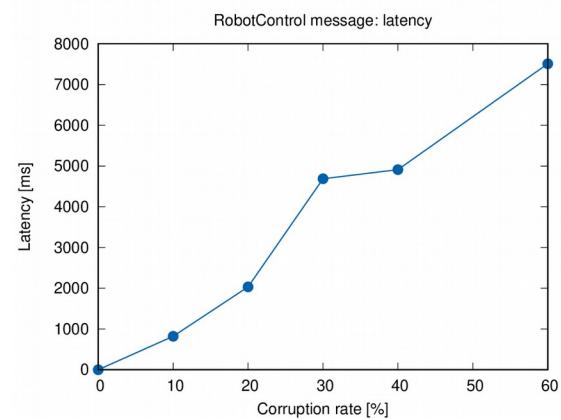
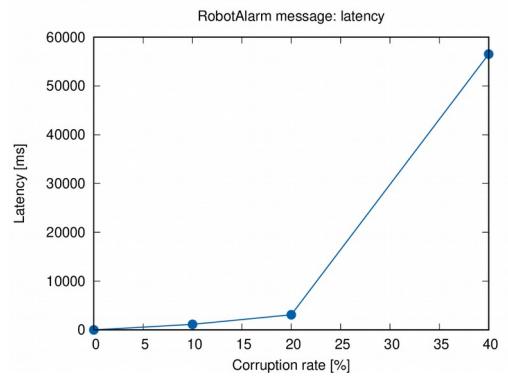
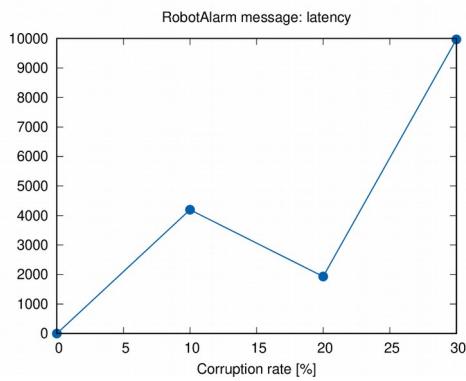
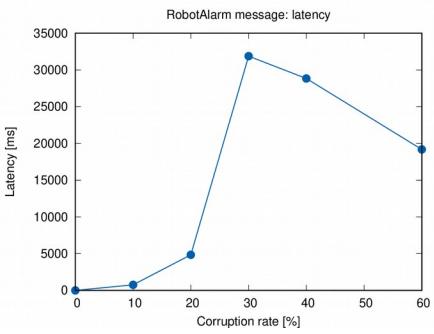


## 5. Corruption:latency

ros2:fastrtps

ros2:opensplice

ros2:connext

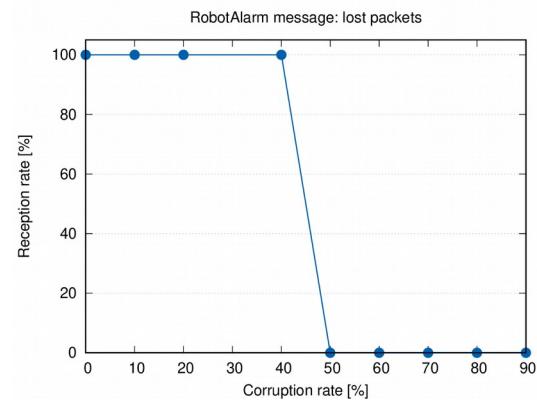
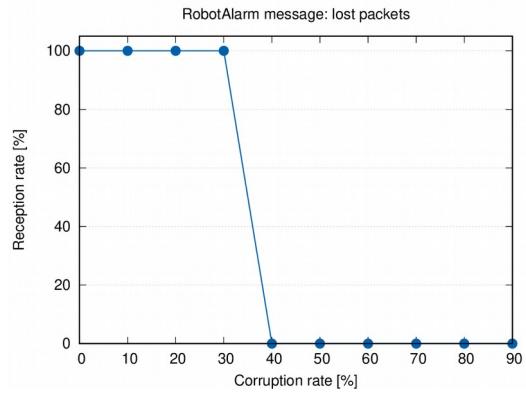
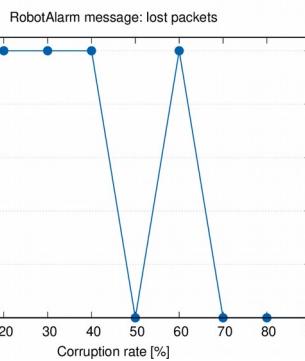


## 5. Corruption:lost-packets

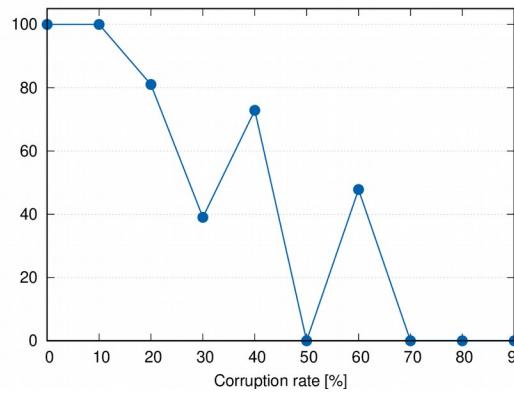
ros2:fastrtps

ros2:opensplice

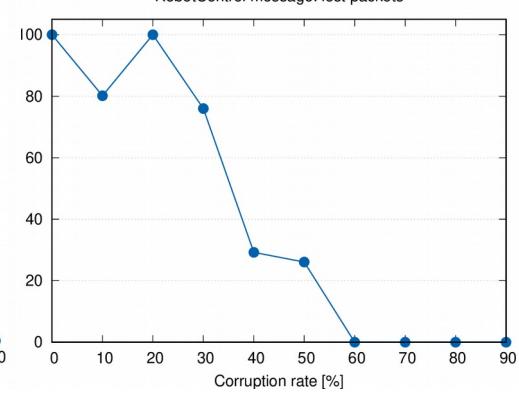
ros2:connext



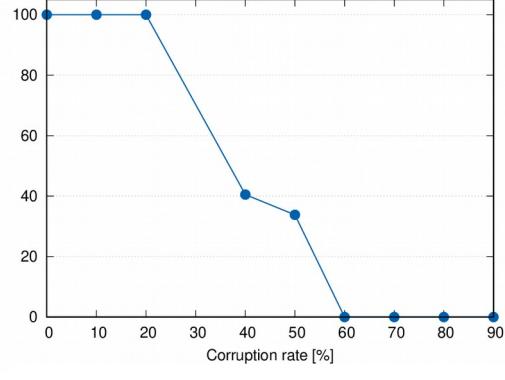
RobotControl message: lost packets



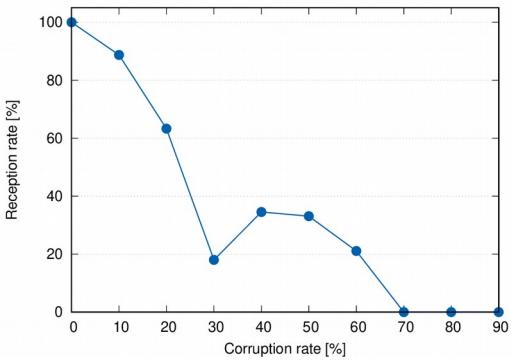
RobotControl message: lost packets



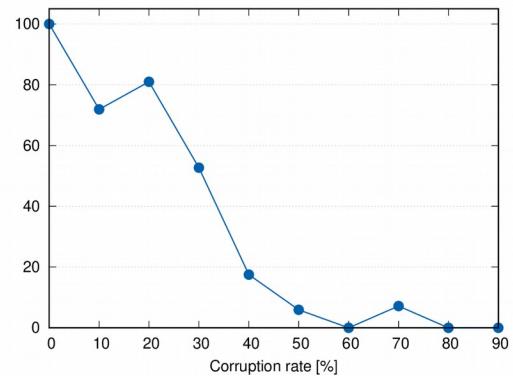
RobotControl message: lost packets



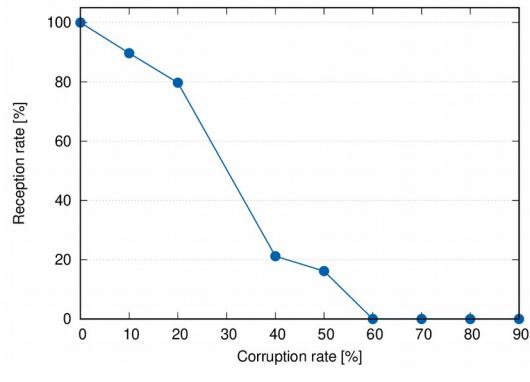
RobotSensor message: lost packets



RobotSensor message: lost packets



RobotSensor message: lost packets

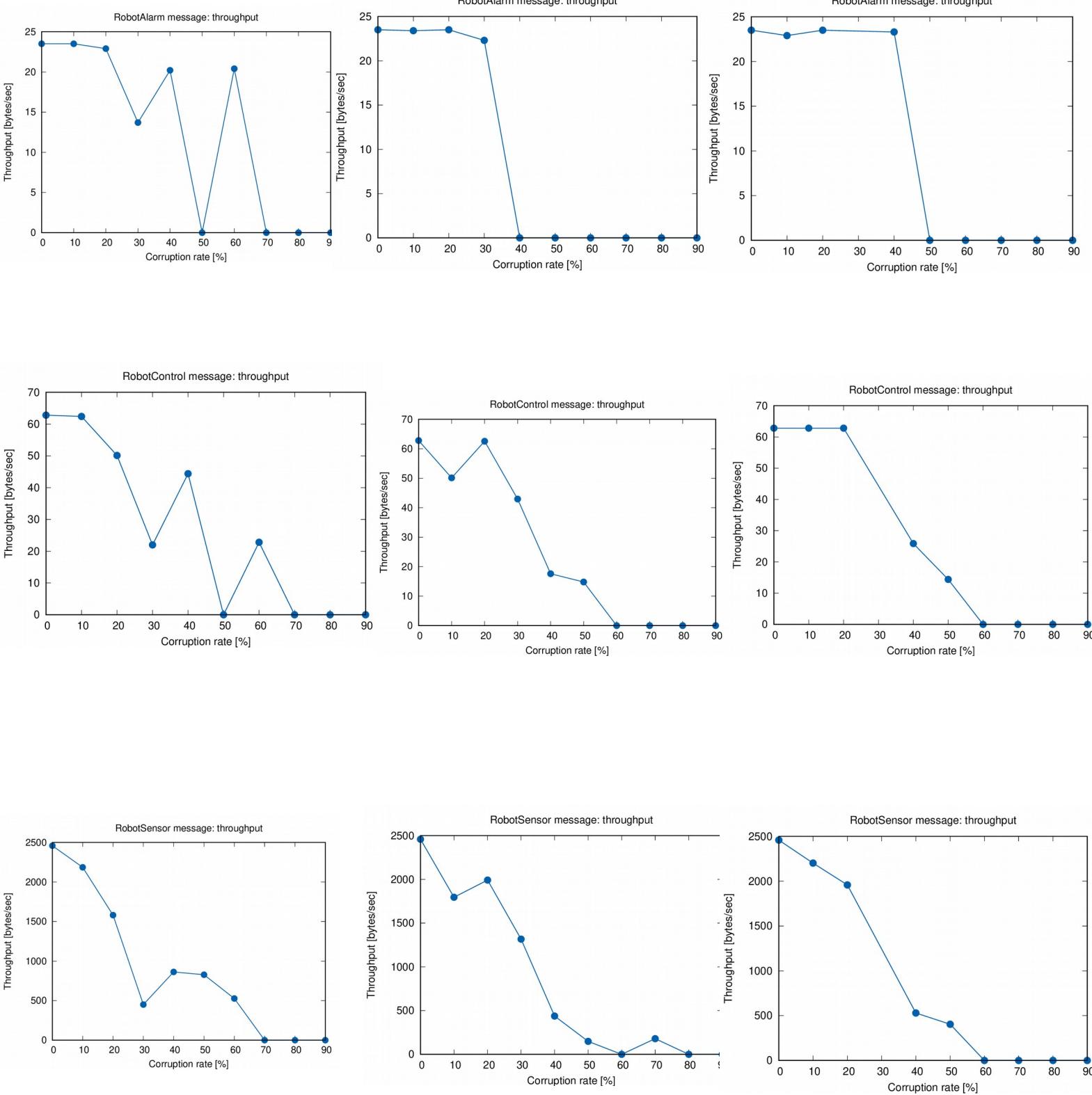


## 5. Corruption:throughput

ros2:fastrtps

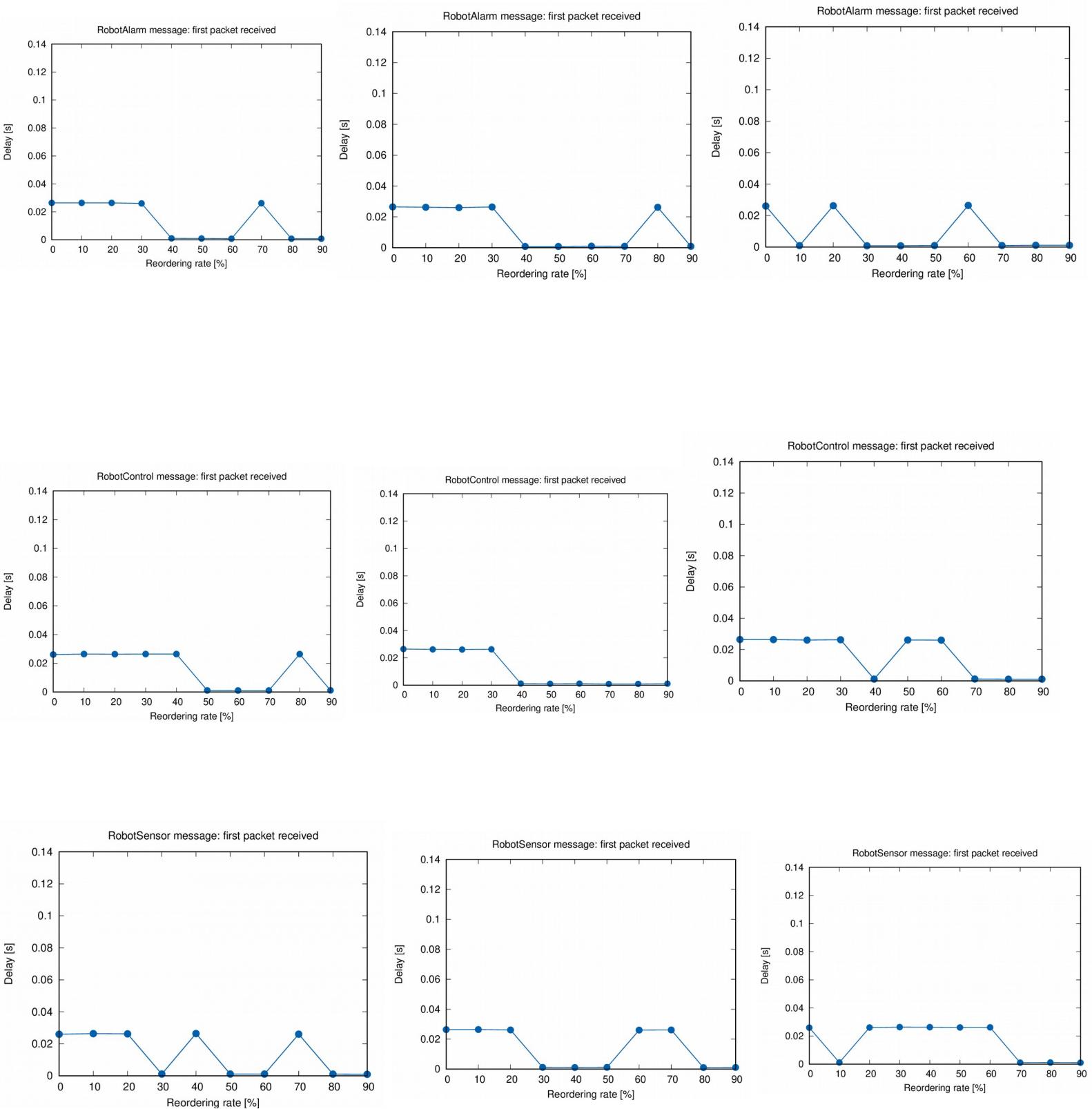
ros2:opensplice

ros2:connext



## 6. Reorder:first-received

ros2:fastrtps      ros2:opensplice      ros2:connext

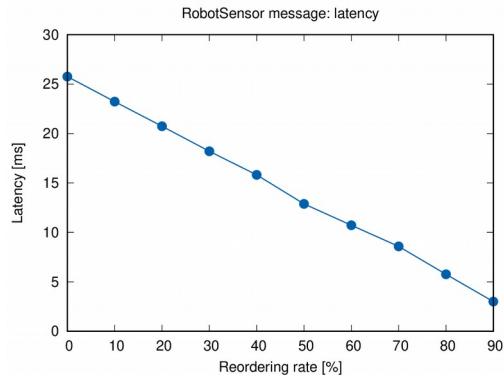
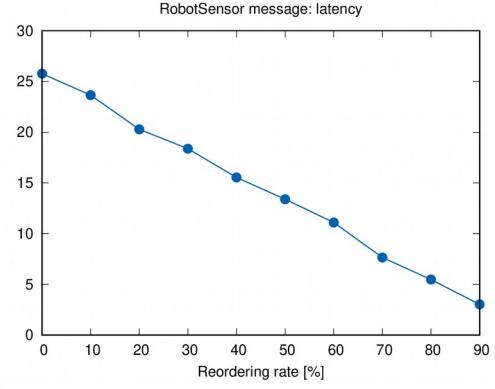
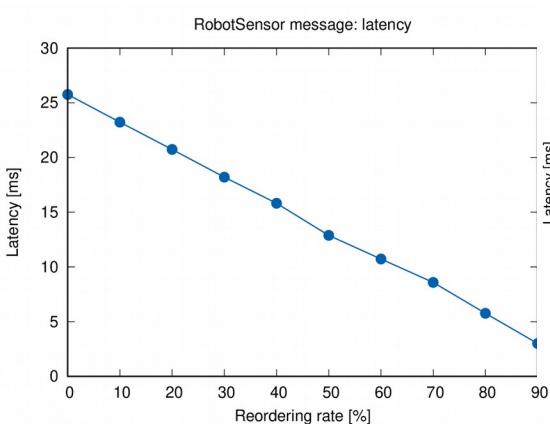
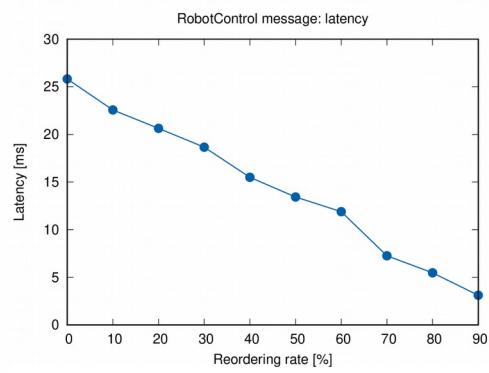
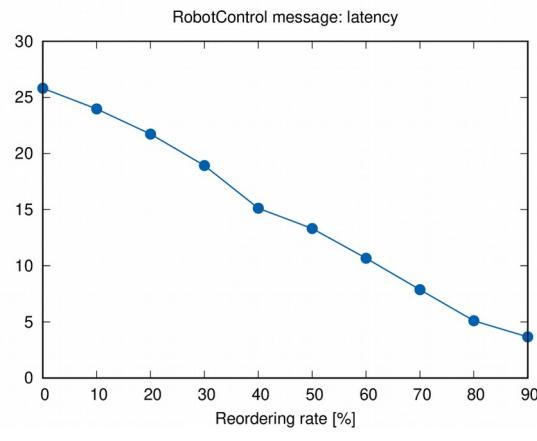
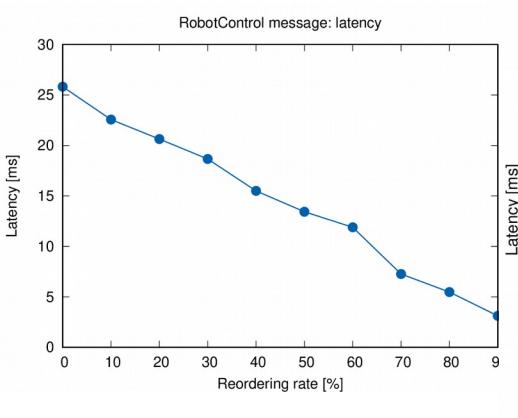
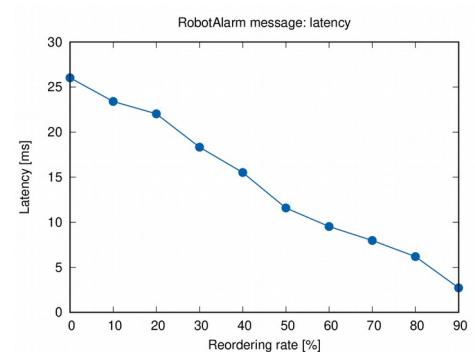
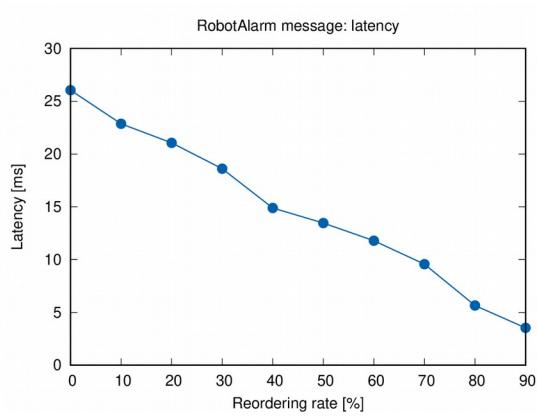
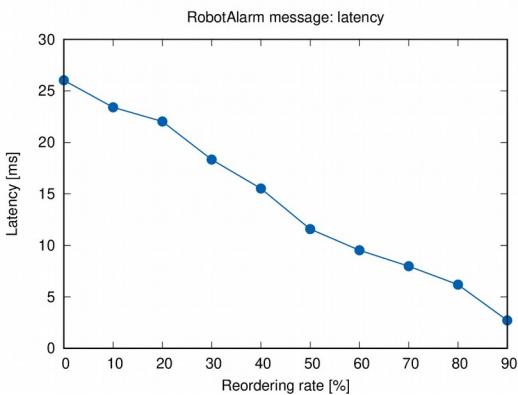


## 6. Reorder:latency

ros2:fastrtps

ros2:opensplice

ros2:connext

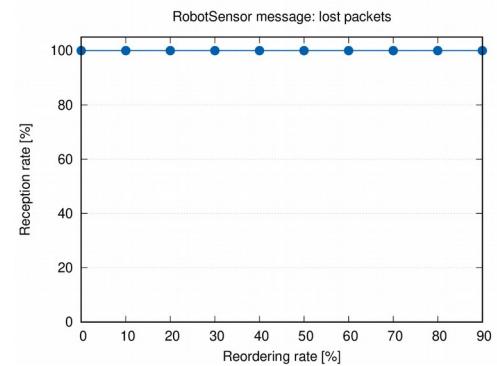
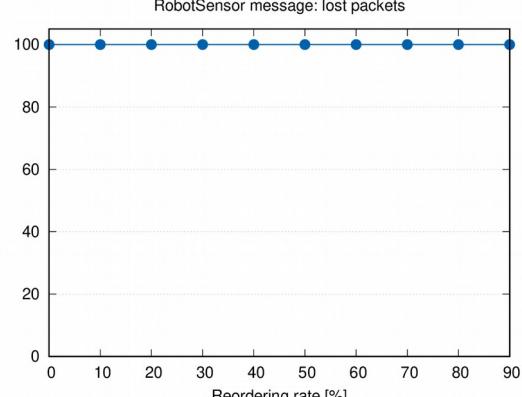
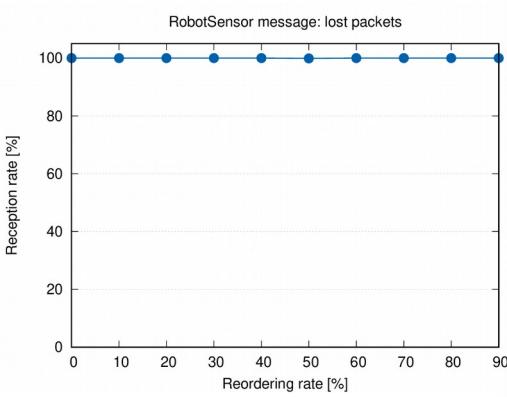
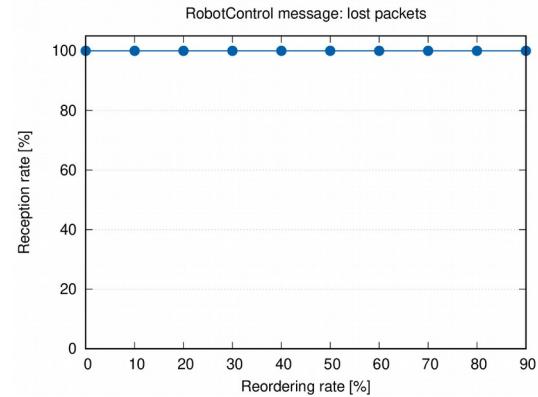
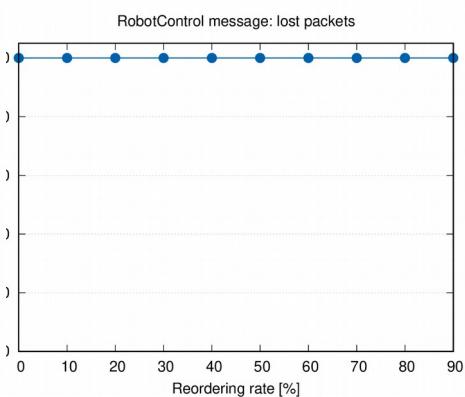
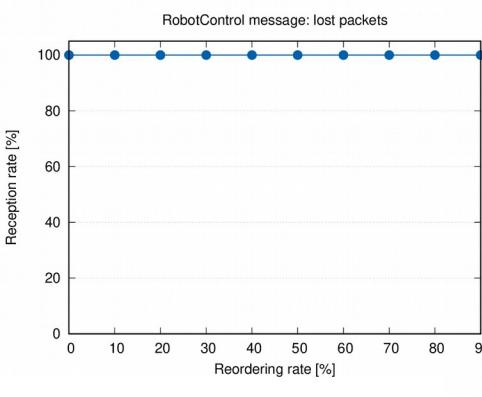
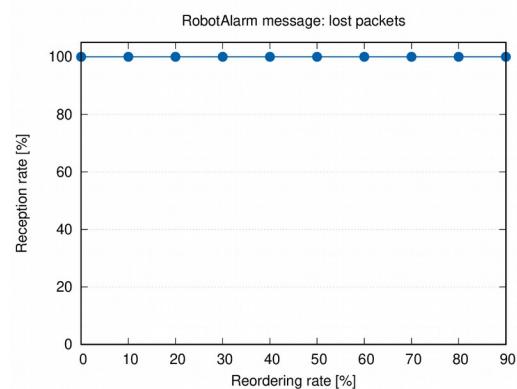
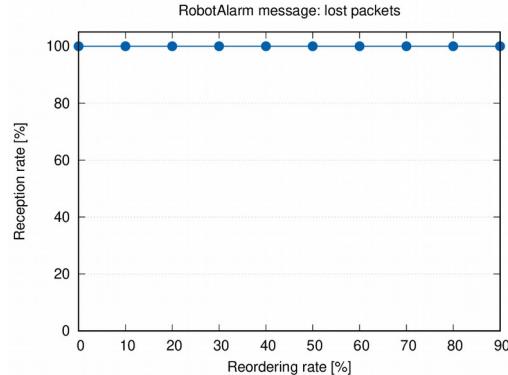
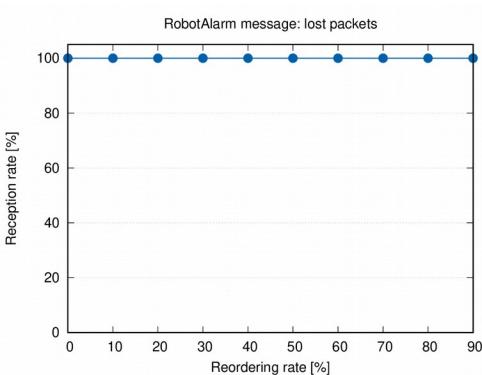


## 6. Reorder:lost-packets

ros2:fastrtps

ros2:opensplice

ros2:connext



## 6. Reorder:throughput

ros2:fastrtps

ros2:opensplice

ros2:connext

