LESSONS LEARNED ON
REAL-TIME AND SECURITY

VÍCTOR MAYORAL VILCHES
Robotics engineer
victor@aliasrobotics.com
REMOVING 0-DAYS FROM ROBOTICS

LINK TO SLIDES
1. REAL-TIME OR REAL-FAST?

REAL-TIME DEFINITION
(control)

Real-time control system means that the control system must provide the control responses or actions to the stimulus or requests within specific times, which therefore depend not just on what the system does but also on how fast it reacts.


REAL-TIME DEFINITION
(security)

...Real-time, zero-latency technologies capable of detecting attacks that target running applications and protecting against those attacks.

REAL-TIME ROBOT STACK

APPLICATION

ROS API (C/C++ LIBRARIES)

ROS MIDDLEWARE (RMW)

DDS

PROTOCOLS (UDP/IP)

HARDWARE AND CONNECTIVITY (e.g. ETHERNET)

OS + DRIVERS

DEV. TOOLS

PRESENTATION

SESSION

TRANSPORT

NETWORK

DATA LINK

PHYSICAL
2. HARDWARE RELEVANCE

BOTTOM UP HARDWARE AND SOFTWARE COMPLIANCE WITH DEADLINES

SIL: Safety Integrity Level
3. HARDWARE COMM. LEVEL
REAL-TIME (CAPABLE) LINK LAYER
REAL-TIME ROBOT STACK

REAL TIME CAPABLE LINK LAYER

- DDS
- UDP/IP
- ETHERNET
- DATA LINK
- PHYSICAL
- TRANSPORT
- NETWORK
- SESSION
- PRESENTATION
- APPLICATION
- ROS 2
## HARDWARE COMM. LEVEL

Towards a real-time capable link layer

### REAL-TIME ETHERNET SOLUTIONS

<table>
<thead>
<tr>
<th>BASED ON TCP/IP NON REAL-TIME PROTOCOLS</th>
<th>BASED ON TCP/IP REAL-TIME</th>
<th>STANDARD ETHERNET IEEE 802.3</th>
<th>MODIFIED ETHERNET MEDIA ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFINET ETHERNET/IP DDS NON REAL-TIME PROTOCOLS</td>
<td>PROFINET ETHERNET/IP DDS REAL-TIME PROTOCOLS</td>
<td>POWER LINK PROFINET RT</td>
<td>ETHERCAT SERCOS III PROFINET IRT</td>
</tr>
<tr>
<td>TCP/UDP/IP</td>
<td>TCP/UDP/IP</td>
<td>ETHERNET</td>
<td>MODIFIED ETHERNET</td>
</tr>
<tr>
<td>ETHERNET</td>
<td>ETHERNET</td>
<td>ETHERNET</td>
<td>MODIFIED ETHERNET</td>
</tr>
</tbody>
</table>

**ETHernet Cabling**

---

## HARDWARE COMM. LEVEL

Towards a Real Time **Capable Link Layer**

### Real-Time Ethernet Solutions

<table>
<thead>
<tr>
<th>Based on TCP/IP</th>
<th>Based on TCP/IP</th>
<th>Standard Ethernet</th>
<th>Modified Ethernet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Real-Time Protocols</td>
<td>Non Real-Time Protocols</td>
<td>IEEE 802.3</td>
<td>Media Access</td>
</tr>
<tr>
<td>PROFINET Ethernet/IP DDS</td>
<td>PROFINET Ethernet/IP DDS</td>
<td>Power Link PROFINET RT</td>
<td>ETHERCAT SERCOS III PROFINET IRT</td>
</tr>
<tr>
<td>TCP/UDP/IP</td>
<td>TCP/UDP/IP</td>
<td>Ethernet</td>
<td>Modified Ethernet</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Ethernet</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Ethernet CABLING

HARDWARE COMM. LEVEL

TOWARDS A REAL TIME **CAPABLE LINK LAYER**

### REAL-TIME ETHERNET SOLUTIONS

<table>
<thead>
<tr>
<th>BASED ON TCP/IP</th>
<th>BASED ON TCP/IP</th>
<th>STANDARD ETHERNET</th>
<th>MODIFIED ETHERNET</th>
</tr>
</thead>
<tbody>
<tr>
<td>NON REAL-TIME PROTOCOLS</td>
<td>REAL-TIME</td>
<td>IEEE 802.3</td>
<td>MEDIA ACCESS</td>
</tr>
<tr>
<td>PROFINET</td>
<td>PROFINET</td>
<td>POWER LINK</td>
<td>ETHERCAT</td>
</tr>
<tr>
<td>ETHERNET/IP</td>
<td>ETHERNET/IP</td>
<td>PROFINET RT</td>
<td>SERCOS III</td>
</tr>
<tr>
<td>DDS</td>
<td>DDS</td>
<td></td>
<td>PROFINET IRT</td>
</tr>
<tr>
<td>NON REAL-TIME PROTOCOLS</td>
<td>REAL-TIME PROTOCOLS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP/UDP/IP</td>
<td>TCP/UDP/IP</td>
<td>ETHERTNET</td>
<td>MODIFIED ETHERNET</td>
</tr>
<tr>
<td>ETHERNET</td>
<td>ETHERTNET</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ETHERNET CABLING**

HARDWARE COMM. LEVEL

NO REAL TIME CAPABLE LINK

LINK CAPACITY 1Gbps. NETWORK LOAD = 900Mbps

IEEE 802.1QBV Time-Aware Scheduler

HARDWARE
COMM. LEVEL

REAL TIME **CAPABLE LINK**

LINK CAPACITY 1Gbps. NETWORK LOAD = 900Mbps

XILINX PROVIDE **MIXED CRITICALITY SOLUTIONS** THAT CONNECT TO **TSN**
4. RTOS AND NETWORKING STACK
OPTIMIZED LINUX NETWORKING STACK
REAL-TIME ROBOT STACK

OPTIMIZED LINUX NETWORKING STACK

- DDS
- UDP/IP
- ETHERNET
RTOS AND NETWORKING STACK

OPTIMIZED LINUX NETWORKING STACK

**no-rt**
we use a vanilla kernel.

**rt-normal**
we use a PREEMPT-RT kernel without binding the round-trip programs and network IRQs to any CPU.

**rt-affinities**
we bind the IRQ thread of the priority queue and the client and server programs to CPU 1 of each device.

**rt-isolation**
we run the roundtrip application in an isolated CPU.

---

RTOS AND NETWORKING STACK

OPTIMIZED LINUX NETWORKING STACK

RTOS AND NETWORKING STACK

OPTIMIZED LINUX NETWORKING STACK

RTOS AND NETWORKING STACK

OPTIMIZED LINUX NETWORKING STACK

---

RTOS AND NETWORKING STACK

OPTIMIZED LINUX NETWORKING STACK

5. ROBOTICS FRAMEWORK & COMMUNICATION MIDDLEWARE

ROS 2.0 FOR REAL-TIME
REAL-TIME
ROBOT STACK

ROS 2.0 FOR REAL-TIME
ROBOTICS FRAMEWORK & COMMUNICATION MIDDLEWARE

ROS 2.0 FOR REAL-TIME

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications.

REAL-TIME REFERENCE DEMO

Gutiérrez, C. S. V., San Juan, L. U., ROS2, real-time, control, pendulum
https://github.com/ros2-realtime-demo/pendulum
6. TIME SYNCHRONIZATION IN ROBOT MODULES
REAL-TIME ROBOT STACK

SYNCHRONIZATION AT ALL LEVELS

ROS 2
DDS
UDP/IP
ETHERNET

APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
TIME SYNCHRONIZATION IN ROBOT MODULES

TIME SYNCHRONIZATION IN ROBOT MODULES

TIME SYNCHRONIZATION IN ROBOT MODULES

ARRIVAL TIME OFFSET FROM THE EXPECTED PERIOD

status-motor1 (Max=59102us)
status-motor2 (Max=59155us)
status-motor3 (Max=99987us)
status-motor4 (Max=99992us)
status-motor5 (Max=99991us)
status-motor6 (Max=99978us)
ranger-finder (Max=59210us)

TIME SYNCHRONIZATION IN ROBOT MODULES

TIME SYNCHRONIZATION IN ROBOT MODULES

REAL-TIME ROBOT STACK

REAL-TIME RESILIENCE TO SECURITY BUGS?
REAL-TIME SECURITY

Somehow understood as

Fig. 1: Latency per packet size in wired and wireless networks.

Table 4. Security measurement comparison.

<table>
<thead>
<tr>
<th>Security enabled</th>
<th>Latency (average µs)</th>
<th>Throughput (average packets/s)</th>
<th>Speed (average Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>260</td>
<td>70,772</td>
<td>35,669</td>
</tr>
<tr>
<td>Full</td>
<td>1363</td>
<td>14,382</td>
<td>72,485</td>
</tr>
</tbody>
</table>
UNDERSTAND THE ATTACK VECTORS THAT AFFECT REAL-TIME THREAT MODEL
SECURITY SOLUTIONS

ALIAS ROBOTICS

SERVICES

- PHYSICAL ROBOT HACKING
- VIRTUAL ROBOT HACKING
- CODE TESTING
- FORENSICS ROBOT SECURITY

PRODUCTS

- RECORDS
  - Records all robot data
  - Enables forensic investigation

- ROBOT IMMUNE SYSTEM
  - Detects threats by learning usual comms
  - Non-intrusive. Real-time. No latencies
  - Hardware agnostic. Plug & Play
REMOVING 0-DAYS FROM ROBOTICS