QoS Provisioning in Industrial Wireless Sensor Networks

Samuele Zoppi, H. Murat Gürsu, Wolfgang Kellerer

Chair of Communication Networks
Technical University of Munich, Germany

Munich, 1st December 2017
Background

Next-generation industrial automation systems will be **wirelessly** interconnected [HPO16].
Background

Next-generation industrial automation systems will be **wirelessly** interconnected [HPO16].

Networked Control Systems (NCS): control loops *closed* over the network.
Background

Next-generation industrial automation systems will be **wirelessly** interconnected [HPO16].

Networked Control Systems (NCS): control loops *closed* over the network.

Stochastic LTI control system:

\[
x_{k+1} = Ax_k + Bu_k + w_k,
\]
\[
u_k = -Kx_k,
\]

\(x_k\) plant dynamic, \(u_k\) control law.
Background

Next-generation industrial automation systems will be **wirelessly** interconnected [HPO16].

Networked Control Systems (NCS): control loops *closed* over the network.

Stochastic LTI control system:

\[
\begin{align*}
x_{k+1} &= Ax_k + Bu_k + w_k, \\
u_k &= -Kx_k,
\end{align*}
\]

$x_k$ plant dynamic, $u_k$ control law.

Sensor sends $x_k$ to the Controller.

Controller computes and sends $u_k$ to the Actuator.
Inverted pendulum as benchmark NCS application.
Background (2)

Inverted pendulum as benchmark NCS application.

Sampling frequency: 20 Hz.
Inverted pendulum as benchmark NCS application.

Sampling frequency: 20 Hz.

Uplink traffic:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \\ \theta_k \\ \dot{\theta}_k \end{bmatrix} \rightarrow 256 \text{ bits} @ 20\text{Hz} = 5 \text{ kbps.}$$
Inverted pendulum as benchmark NCS application.

Sampling frequency: 20 Hz.

Uplink traffic:

\[
\mathbf{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \\ \theta_k \\ \dot{\theta}_k \end{bmatrix} \rightarrow 256 \text{ bits @ 20Hz} = 5 \text{ kbps.}
\]

Downlink traffic:

\[ u \rightarrow 64 \text{ bits @ 20Hz} = 1.25 \text{ kbps.} \]
Inverted pendulum as benchmark NCS application.

Sampling frequency: 20 Hz.

Uplink traffic:

\[
\mathbf{x}_k = \begin{bmatrix}
  x_k \\
  \dot{x}_k \\
  \theta_k \\
  \dot{\theta}_k
\end{bmatrix} \rightarrow \text{256 bits @ 20Hz = 5 kbps.}
\]

Downlink traffic:

\[u \rightarrow \text{64 bits @ 20Hz = 1.25 kbps.}\]

WSN (PHY IEEE 802.15.4) link \(\rightarrow\) 250 kbps.
Motivation

Wireless Sensor Networks (WSN) can support NCS traffic.
Motivation

Wireless Sensor Networks (WSN) can support NCS traffic.

Control loops pose strict QoS requirements on wireless communications.
Motivation

Wireless Sensor Networks (WSN) can support NCS traffic.

Control loops pose strict **QoS requirements** on wireless communications.

WSN suffers from external interference and unreliable links [GVZK16].
Motivation

Wireless Sensor Networks (WSN) can support NCS traffic.

Control loops pose strict **QoS requirements** on wireless communications.

WSN suffers from external interference and unreliable links [GVZK16].

**Problem:** Current WSN lack dynamic real-time QoS provisioning.
Motivation

Wireless Sensor Networks (WSN) can support NCS traffic.

Control loops pose strict **QoS requirements** on wireless communications.

WSN suffers from external interference and unreliable links [GVZK16].

**Problem:** Current WSN lack dynamic real-time QoS provisioning.

**Approach:**

1. Definition of a QoS provisioning framework for IWSN.
2. Implementation of the framework in a testbed.
Outline

Background & Motivation

QoS Provisioning Framework

Implementation

Conclusions & Further Work
Network Architecture

Centralized, star topology.
Network Architecture

Centralized, star topology.

Network elements:

1. **Application (App)**: industrial NCS application
2. **Network Manager (NM)**: manager of the Network Resources of the entire WSN
3. **Gateway (GW)**: interface btw the WSN devices, the NM and Apps
4. **Sensor (s)**: WSN device
Network Architecture

Centralized, star topology.

Network elements:

1. **Application (App)**: industrial NCS application

2. **Network Manager (NM)**: manager of the Network Resources of the entire WSN

3. **Gateway (GW)**: interface btw the WSN devices, the NM and Apps

4. **Sensor (s)**: WSN device

Data links btw NM and WSN devices through the GW.

Control links btw App and WSN devices through the GW.
QoS Framework (1)

Radio Resource Model

Wireless DetServ

QoS Requirement

Scheduler / Radio Res. Manager

Link Quality Info.

WSN Device

GW

NM

f

Data

Control

t

App

SAMUELE ZOPPI | ITG Fachausschuss 5.2 workshop on “Cellular Internet of Things” | Munich, Germany
QoS Framework (1)

Radio Resource Manager inputs:

1. QoS requirements from the application.
2. QoS Model of the MAC radio resources.
3. Link Quality Information of the radio resources.
QoS Framework (1)

Radio Resource Manager **inputs:**
1. QoS requirements from the application.
2. QoS Model of the MAC radio resources.
3. Link Quality Information of the radio resources.

Radio Resource Manager **outputs:**
1. Radio resources for Data packets (application).
2. Radio resources for Control packets (schedules, LQI probes, ...).
QoS Framework (2)

Dynamic scheduling is possible in a TDMA-FDMA radio resource grid model.

Dynamic scheduling protocol.
Dynamic scheduling is possible in a TDMA-FDMA radio resource grid model.

Dynamic scheduling protocol:

1. Acquisition of Link Quality Information (input) → estimated Packet Delivery Ratio → EWMA for estimation
QoS Framework (2)

Dynamic scheduling is possible in a TDMA-FDMA radio resource grid model.

Dynamic scheduling protocol:

1. Acquisition of Link Quality Information (input)
   → estimated Packet Delivery Ratio
   → EWMA for estimation

2. Acquisition of QoS requirements (input)
   → Target application reliability (i.e. 90%)
   → Target delay bound (deadline)

Dynamic scheduling protocol.
QoS Framework (2)

Dynamic scheduling is possible in a TDMA-FDMA radio resource grid model.

Dynamic scheduling protocol:

1. Acquisition of Link Quality Information (input)
   → estimated Packet Delivery Ratio
   → EWMA for estimation

2. Acquisition of QoS requirements (input)
   → Target application reliability (i.e. 90%)
   → Target delay bound (deadline)

3. Distribution of new schedules (output)
   → sequence of radio resources (time-freq. pairs)
   → distributed using the beacon
   → calculated with a reliability-based scheduler
QoS Framework (3) - Scheduling algorithm

Reliability is provided allocating multiple transmissions in the frame.
QoS Framework (3) - Scheduling algorithm

Reliability is provided allocating multiple transmissions in the frame.

The radio resources are modeled using a scheduling graph:

- Nodes represent time instants before/after time slots.
- Edges represent different frequencies and they are weighted by their PDR.
QoS Framework (3) - Scheduling algorithm

Reliability is provided allocating multiple transmissions in the frame.

The radio resources are modeled using a scheduling graph:
- Nodes represent time instants before/after time slots.
- Edges represent different frequencies and they are weighted by their PDR.

A Constrained Shortest Path scheduling algorithm finds the schedule (path) fulfilling the target reliability. \( \rightarrow \{(0,1),(1,3),(2,0)\}\)
QoS Framework (4) - Results

Simulation results of dynamic scheduling with latency and reliability constraints.

Reliability-based scheduling [eaED].
QoS Framework (4) - Results

Simulation results of dynamic scheduling with latency and reliability constraints.

WSN operating in a dynamic interference scenario (Wi-Fi APs, @2.4GHz).

Reliability-based scheduling [eaED].
QoS Framework (4) - Results

Simulation results of dynamic scheduling with latency and reliability constraints.

WSN operating in a dynamic interference scenario (Wi-Fi APs, @2.4GHz).

Dynamic scheduling in presence of increasing Wi-Fi transmission power ($P_{tx}$).

Reliability-based scheduling [eaED].
QoS Framework (4) - Results

Simulation results of dynamic scheduling with latency and reliability constraints.

WSN operating in a dynamic interference scenario (Wi-Fi APs, @2.4GHz).

Dynamic scheduling in presence of increasing Wi-Fi transmission power ($P_{tx}$).

Reliability-based scheduling [eaED].

WDetServ guarantees reliability and delay bounds reacting against interference.

Samuele Zoppi | ITG Fachausschuss 5.2 workshop on “Cellular Internet of Things” | Munich, Germany
Outline

Background & Motivation

QoS Provisioning Framework

Implementation

Conclusions & Further Work
Implementation (1)

Deployment of an WDetServ NCS testbed:

1. Control logic (Controller) in the Cloud.
2. Sensing and Actuation in the WSN devices.
3. Gateway acts as forwarding entity.
4. Inverted Pendulum as benchmark control application.

Network architecture.
Implementation (2)

**Problem:** several HW and SW latency bottlenecks.

Sensor-to-cloud delay measurements [GZO+].
Implementation (2)

**Problem:** several HW and SW latency bottlenecks.

**Solution:** ad-hoc HW solutions for GW and WSN:

- **Gateway**
  - high perf., multi-radio, multi-processor
  - $\rightarrow$ low-latency, multi-channel SDR

- **Sensor**
  - limited perf., single antenna, single processor
  - $\rightarrow$ Zolertia Z1/RE-Mote, TI SimpleLink

<table>
<thead>
<tr>
<th>Sensor to Controller Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4 / L3</td>
</tr>
<tr>
<td>Z1 min</td>
</tr>
<tr>
<td>OM min</td>
</tr>
<tr>
<td>Z1 max</td>
</tr>
<tr>
<td>OM max</td>
</tr>
</tbody>
</table>

Sensor-to-cloud delay measurements [GZO⁺].
Outline

Background & Motivation

QoS Provisioning Framework

Implementation

Conclusions & Further Work
Conclusions

NCS traffic can be supported by WSN if QoS provisioning is implemented.
Conclusions

NCS traffic can be supported by WSN if QoS provisioning is implemented.

Wireless DetServ provides the building blocks for QoS provisioning (latency, reliability, QoC, ...) in WSN.
Conclusions

NCS traffic can be supported by WSN if QoS provisioning is implemented.

Wireless DetServ provides the building blocks for QoS provisioning (latency, reliability, QoC, ...) in WSN.

The implemented reliability-based scheduler is able to react to changes in the wireless environment.
Conclusions

NCS traffic can be supported by WSN if QoS provisioning is implemented.

Wireless DetServ provides the building blocks for QoS provisioning (latency, reliability, QoC, ...) in WSN.

The implemented reliability-based scheduler is able to react to changes in the wireless environment.

Latency is the major issue for HW implementation (radio, processing, ext. interface).
Further Work

Measurements of NCS Inverted Pendulum operating over the testbed will be performed.

NCS cross-layer scheduling algorithms will be developed.

Different Link Quality Estimators will be evaluated in the testbed.

Multi-radio, multi-processor, high-speed interface solutions will be implemented.
References


- Halit Murat Gürsu, Samuele Zoppi, Hasan Yagiz Ozkan, Yadahunandana R. K., and Wolfgang Kellerer. Tactile sensor to cloud delay: A hardware and processing perspective. In IEEE ICC 2018 SAC Symposium Internet of Things Track (ICC’18 SAC-6 IoT), SUBMITTED.