End-to-end Real-time Guarantees in Wireless Cyber-physical Systems

Romain Jacob    Marco Zimmerling
Pengcheng Huang  Jan Beutel
Lothar Thiele

RTSS 16 - IoT and Networking Session
December 1, 2016
Predictability is key!
A Cyber-physical System goes beyond a real-time wireless network

Predictability

Real-time guarantees

> Deadlines are met

Wireless network
- Low-power
- Multi-hop

Network deadline
A Cyber-physical System goes beyond a real-time wireless network

Predictability

Real-time guarantees > *End-to-end deadlines are met*

- **Actuator**
- **Sensor**

Wireless network
- Low-power
- Multi-hop

Network deadline

End-to-end deadline
A Cyber-physical System goes beyond a real-time wireless network

Predictability

Real-time guarantees > *End-to-end deadlines are met*
Buffer management > *No buffer overflow*
Design goals

**Predicatability**
- Real-time guarantees
- Buffer management

**Adaptability**
- Efficiency
- Composability

| Network | Device | System |
First piece of a **predictable and adaptive** wireless network.
First piece A *predictable* and *adaptive* wireless network

- Low-power
- Multi-hop

Wireless network

Controller

Actuator

Sensor
State-of-the-art wireless protocols are *predictable* or *adaptive*

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Predictability</th>
<th>Efficiency</th>
<th>Adaptivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splash, RAP</td>
<td>Efficient</td>
<td>Adaptive</td>
<td><em>not Predictable</em></td>
</tr>
<tr>
<td>WirelessHART</td>
<td>Predictable</td>
<td>Efficient</td>
<td><em>not Adaptive</em></td>
</tr>
</tbody>
</table>
Blink  A real-time, reliable and adaptive wireless protocol

Adaptive Based on Glossy
> Flooding primitive

Reliable Average 99.97% reception rate
> Multiple testbeds
> Tested up to 94 nodes

Real-time Online scheduling
> EDF-based Lazy Scheduling

Blink

A real-time, reliable and adaptive wireless protocol

**Blink**  
A real-time, reliable and adaptive wireless protocol

Abstraction

Low-power Wireless Bus (LWB)

> MAC protocol

Communication in rounds  

TDMA-based

> Sleep between rounds

> Wireless yields sync!

---

Design goals

- Real-time guarantees
- Buffer management
- Adaptability
- Efficiency
- Composability

<table>
<thead>
<tr>
<th></th>
<th>Network</th>
<th>Device</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blink</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>na</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>na</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>na</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Network

- ✔️ Real-time guarantees
- ✔️ Buffer management
- ✔️ Adaptability
- ✔️ Efficiency
- na Composability
Second piece **A dual-processor architecture** to mitigate local interference

- Wireless network
  - Low-power
  - Multi-hop

Actuator

Sensor

Controller
Second piece A dual-processor architecture to mitigate local interference

- Wireless network
  - Low-power
  - Multi-hop
Bolt is a dual-processor architecture to mitigate local interference.

- Real-time behavior
- Efficient: $\mu W > \text{sleep}$, $mW > \text{active}$
- Composable: Hardware/Software free composition

Formally verified, implemented, tested, deployed

### Design goals

<table>
<thead>
<tr>
<th>Design Goal</th>
<th>Blink</th>
<th>Bolt</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network</strong></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Device</strong></td>
<td>na</td>
<td>✔</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>na</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Real-time guarantees</strong></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Buffer management</strong></td>
<td>na</td>
<td></td>
</tr>
<tr>
<td><strong>Adaptability</strong></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Composability</strong></td>
<td>na</td>
<td>✔</td>
</tr>
</tbody>
</table>
### Design goals → How can we fill the blanks?

<table>
<thead>
<tr>
<th></th>
<th>Blink</th>
<th>Bolt</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time guarantees</td>
<td>✔</td>
<td>✔</td>
<td>?</td>
</tr>
<tr>
<td>Buffer management</td>
<td>na</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Adaptability</td>
<td>✔</td>
<td>✔</td>
<td>?</td>
</tr>
<tr>
<td>Efficiency</td>
<td>✔</td>
<td>✔</td>
<td>?</td>
</tr>
<tr>
<td>Composability</td>
<td>na</td>
<td>✔</td>
<td>?</td>
</tr>
</tbody>
</table>
### Distributed Real-time Protocol (DRP)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Blink</th>
<th>Bolt</th>
<th>DRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time guarantees</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Buffer management</td>
<td>na</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Adaptability</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Efficiency</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Composability</td>
<td>na</td>
<td>✓</td>
<td>?</td>
</tr>
</tbody>
</table>
DRP is based on three main concepts

Communication is constrained within *registered flows* only

*Global requirements are split* across distributed components

Interaction between components is based on *contracts*
DRP is based on three main concepts

Communication is constrained within *registered flows* only

*Global requirements are splited* across distributed components

Interaction between components is based on *contracts*
Communication is constrained within *registered flows* only

Flow $i$  \[ F_i = ( n_i^s, n_i^d, T_i, J_i, D_i ) \]

Source
Destination
Min. release interval
Jitter
End-to-end deadline

Release model *Sporatic* with jitter
DRP is based on three main concepts

Communication is constrained within *registered flows* only

*Global requirements are split* across distributed components

Interaction between components is based on *contracts*
Global requirements are split across distributed components

Real-time guarantee

\[
\text{DRP} : D^s_i + D^\text{net}_i + D^d_i = D
\]

Source | Wireless protocol | Destination

end-to-end deadline : \( D \)
DRP is based on three main concepts

Communication is constrained within *registered flows* only

*Global requirements are splitted* across distributed components

Interaction between components is based on *contracts*
Interaction between components is based on *contracts*.

- Node ↔ Network
- Max resource demand
- Min service delivered
- Predictability
- Performance
Example

Scheduling of Communication Processors (CP)

- **Real-time guarantees**
  - Participate in rounds
    - $T_{\text{net}}$ Blink schedule
  - Flush before rounds
  - Write after rounds
    - $T_{\text{net}}$ Blink schedule

- **Buffer management**
  - Flush Bolt regularly
    - $T_f^S$ Comm. Processor schedule
Example

Scheduling of Communication Processors (CP)

How can we guarantee that all tasks are schedulable?

Participate in rounds
Flush before rounds
Write after rounds
Flush Bolt regularly

\[ T_{net} \]

\[ C_f \quad C_{net} \quad C_w \]

\[ T_f \]

\[ T_{f_s} \]

\[ t \]
How can we guarantee that all tasks are schedulable?

Participate in rounds
Flush before rounds
Write after rounds
Flush Bolt regularly

Example
Scheduling of Communication Processors (CP)
Example

Scheduling of Communication Processors (CP)

How to guarantee that all tasks are schedulable?

Participate in rounds
Flush before rounds
Write after rounds
Flush Bolt regularly

Variable!

Conflict!
Example

Scheduling of Communication Processors (CP)

How to guarantee that all tasks are schedulable?

Solution  TDMA approach

\[ C_{CP} \triangleq C_f + C_{net} + C_w \]
Example

Scheduling of Communication Processors (CP)

How to guarantee that all tasks are schedulable?

Participate in rounds
- Flush before rounds
- Write after rounds
- Flush Bolt regularly

Solution
- TDMA approach

\[
C_{CP} \triangleq C_f + C_{net} + C_W
\]

\[
T_f^S \triangleq C_{CP}
\]

\[
T_{net} > T_f^S
\]
Example

Scheduling of Communication Processors (CP)

How to guarantee that all tasks are schedulable?

Solution

TDMA approach

\[ C_{CP} \triangleq C_f + C_{net} + C_w \]

\[ T^S_f \triangleq C_{CP} \]

\[ T_{net} \triangleq k \cdot C_{CP} \]
Predictability of Network + Predictability of Devices + DRP contracts = \textit{System predictability}

Ultimately

\begin{align*}
\text{End-to-end latency} & = f(\text{Local parameters}) \\
\text{Buffer size} & = \text{End-to-end deadline} \\
\text{Memory space} & = \text{Memory space}
\end{align*}

\text{Can we set local parameters such that...}

If \text{YES} \quad \text{System predictability by design}

\begin{tabular}{|l|c|}
\hline
Flush interval & $T_f^s$ \\
Flush interval & $T_f^d$ \\
Network deadline & $D_{i}^{\text{net}}$ \\
\hline
\end{tabular}
## Design goals

<table>
<thead>
<tr>
<th></th>
<th>Blink</th>
<th>Bolt</th>
<th>DRP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network</strong></td>
<td>✔</td>
<td>✔</td>
<td>?</td>
</tr>
<tr>
<td><strong>Device</strong></td>
<td>na</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

- **Real-time guarantees**
- **Buffer management**
- **Adaptability**
- **Efficiency**
- **Composability**
## Design goals

### Predicatability

<table>
<thead>
<tr>
<th></th>
<th>Blink</th>
<th>Bolt</th>
<th>DRP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Real-time guarantees</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Buffer management</strong></td>
<td>na</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Adaptability</strong></td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td><strong>Composability</strong></td>
<td>na</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
From there, *adaptability* is one (close) step away

*Can we set local parameters such that...*

End-to-end latency
Buffer size

\(\lor\)

End-to-end deadline
Memory space
From there, *adaptability* is one (close) step away.

**Example**

**Communication Processor**

\[
S_{Bolt} \geq \sum_{F_i \in \mathcal{F}_{new}, \atop n=n_i^s} \left[ \frac{T_i^s + C_w + C_r + J_i}{T_i} \right] \\
S_{CP} \geq \sum_{F_i \in \mathcal{F}_{new}, \atop n=n_i^s} \left[ 1 + \frac{D_i + J_i + C_f}{T_i} \right] + \sum_{F_i \in \mathcal{F}_{new}, \atop n=n_i^d} 1
\]

*ADMISSION TESTS*

Depends only on *local parameters!*
Adaptability is achieved via a *distributed registration scheme*

![Diagram showing the registration process](image-url)
### Design goals

<table>
<thead>
<tr>
<th>Predicatability</th>
<th>Network</th>
<th>Device</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time guarantees</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Buffer management</td>
<td>na</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adaptability</th>
<th>Network</th>
<th>Device</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Composability</td>
<td>na</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
The detailed system analysis allows for *performance optimization*

Minimal admissible end-to-end deadline

\[ D_{min} = \delta_f^{const} + T_{min} + T_{min} + \delta_g^{const} + T_{f,min} \]

Packet flow

**end-to-end latency** \(\geq D_{min}\)
The detailed system analysis allows for *performance optimization*

Minimal admissible end-to-end deadline

\[
D_{\text{min}} = f_{\text{const}} + T_{\text{min}} + T_{\text{min}} + g_{\text{const}} + T_{f,\text{min}}
\]

end-to-end latency \( \geq D_{\text{min}} \)

Source delay

- *Message release by the source*
- *Available for communication*
The detailed system analysis allows for performance optimization

Minimal admissible end-to-end deadline

\[ D_{\text{min}} = \delta_f + T_{\text{min}} + T_{\text{min}} + \delta_g + T_{f,\text{min}} \]

Network delay: Waiting time

- Message available
- Message processed by the network

end-to-end latency \( \geq D_{\text{min}} \)
The detailed system analysis allows for *performance optimization*

Minimal admissible end-to-end deadline

$$D_{\text{min}} = \delta_f^{\text{const}} + T_{\text{min}} + T_{\text{min}} + \delta_g^{\text{const}} + T_{f,\text{min}}$$

**end-to-end latency $\geq D_{\text{min}}$**

**Network delay: Transmission**

- Message processed by the network
- Message transmitted to the destination node
The detailed system analysis allows for **performance optimization**

Minimal admissible end-to-end deadline

\[ D_{min} = \delta_f^{const} + T_{min} + T_{min} + \delta_g^{const} + T_{f,min} \]

Destination delay: Bolt

- Message transmitted to the destination node
- Message available in the Bolt queue
The detailed system analysis allows for performance optimization

Minimal admissible end-to-end deadline

\[ D_{min} = \delta_f^{const} + T_{min} + T_{min} + \delta_g^{const} + T_{f,min} \]

end-to-end latency \( \geq D_{min} \)

Destination delay: Application
- Message available in the Bolt queue
- Message retrieved by the destination application
The detailed system analysis allows for **performance optimization**

Minimal admissible end-to-end deadline

\[
D_{\text{min}} = \delta_f^{\text{const}} + T_{\text{min}} + T_{\text{min}} + \delta_g^{\text{const}} + T_{f,\text{min}}
\]

Given

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet size</td>
<td>32 Bytes</td>
</tr>
<tr>
<td>(C_{\text{net}})</td>
<td>1 s</td>
</tr>
<tr>
<td>(T_f^d)</td>
<td>0.1 s</td>
</tr>
<tr>
<td>(D_{\text{min}})</td>
<td>3.46 s</td>
</tr>
<tr>
<td>Max data rate</td>
<td>29.7 Bps per flow</td>
</tr>
</tbody>
</table>
### Design goals

<table>
<thead>
<tr>
<th>Predicatability</th>
<th>Blink</th>
<th>Bolt</th>
<th>DRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time guarantees</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Buffer management</td>
<td>na</td>
<td>✅</td>
<td>✅</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adaptability</th>
<th>Blink</th>
<th>Bolt</th>
<th>DRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Composability</td>
<td>na</td>
<td></td>
<td>✅</td>
</tr>
</tbody>
</table>
Using *flooding primitives* enables the design of both *adaptive AND predictable* Wireless Cyber-Physical Systems.
End-to-end Real-time Guarantees in Wireless Cyber-physical Systems

Romain Jacob  Marco Zimmerling
Pengcheng Huang  Jan Beutel
Lothar Thiele

RTSS 16 - IoT and Networking Session
December 1, 2016
Simulation correlates closely with the analysis.

Typical simulation trace result:

- Percentage of packets [%]
- End-to-end latency of packets [% of analytic bound]

Analytic bound: 96%