Reconfigurable Communication Middleware for FlexRay-based Distributed Embedded Systems

Diptesh Majumdar, Licong Zhang, Purandar Bhaduri, Samarjit Chakraborty

1 Department of Computer Science and Engineering, IIT Guwahati, India
2 Institute for Real-Time Computer Systems, TU Munich, Germany

RTCSA 2015
Aug. 20, 2015, Hong Kong
Motivation

Automotive E/E architecture increases rapidly in scale and complexity

Increasingly more and complex software and data

Constrained communication resources

Static design and development of safety-critical buses (e.g., FlexRay)

Multi-mode, Plug-and-Play applications and software installation/update after sales
Motivation
- Automotive E/E architecture increases rapidly in scale and complexity
- Increasingly more and complex software and data
- Constrained communication resources
- Static design and development of safety-critical buses (e.g., FlexRay)
- Multi-mode, Plug-and-Play applications and software installation/update after sales

Problem
- Online communication reconfiguration of FlexRay-based ECU network
- Communication resource re-allocation for
  - multi-mode applications
  - newly activated applications
Overview

- **Motivation**
  - Automotive E/E architecture increases rapidly in scale and complexity
  - Increasingly more and complex software and data
  - Constrained communication resources
  - Static design and development of safety-critical buses (e.g., FlexRay)
  - Multi-mode, Plug-and-Play applications and software installation/update after sales

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for
    - multi-mode applications
    - newly activated applications

- **Approach**
  - Insertion of a middleware layer
  - Reconfigurable data-to-schedule mapping
  - Online configuration calculation and deployment
## Background

- **FlexRay-based ECU networks**
  - Hardware architecture
Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications

![Diagram showing FlexRay-based ECU networks](image-url)
Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications

![Diagram showing FlexRay-based ECU networks with ECU nodes and FlexRay communication paths]
Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications
  - Task mapping / bus communication
Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications
  - Task mapping / bus communication
  - Multi-mode application

<table>
<thead>
<tr>
<th>Mode</th>
<th>Resource</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>2</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>3</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Off</td>
<td>none</td>
<td>none</td>
</tr>
</tbody>
</table>

Mode1

Mode2

Mode3

Off
Background

- **FlexRay communication**
  - Hybrid protocol: mixed time-triggered and event-triggered paradigm
  - Communication cycle, static segment (ST) and dynamic segment (DYN)
**Background**

- **FlexRay communication**
  - Hybrid protocol: mixed time-triggered and event-triggered paradigm
  - Communication cycle, static segment (ST) and dynamic segment (DYN)
  - 64-cycle-sequence, FlexRay Schedule $\Theta_i = (S_i, B_i, R_i)$

```
<table>
<thead>
<tr>
<th>cycle</th>
<th>Communication Cycle</th>
<th>T_{bus}</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>ST</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- $S_i$: slot number
- $B_i$: base cycle
- $R_i$: repetition rate
Background

- **FlexRay communication**
  - Hybrid protocol: mixed time-triggered and event-triggered paradigm
  - Communication cycle, static segment (ST) and dynamic segment (DYN)
  - 64-cycle-sequence, FlexRay Schedule $\Theta_i = (S_i, B_i, R_i)$

```
<table>
<thead>
<tr>
<th>Cycle</th>
<th>Communication Cycle</th>
<th>$T_{bus}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- $S_i$: slot number
- $B_i$: base cycle
- $R_i$: repetition rate

```
$\Theta_1 = (2, 0, 2)$
$\Theta_2 = (4, 1, 4)$
```
### Motivational Example

#### Applications

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>$2T_{bus}$</td>
</tr>
</tbody>
</table>

$\alpha$: mode  
$w$: data size  
$p$: period  
$J$: performance  

2 static slots of 8 bytes payload
Motivational Example

### Applications

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2 $T_{bus}$</td>
</tr>
</tbody>
</table>

$\alpha$: mode  
$w$: data size  
$p$: period  
$J$: performance

Possible mode combinations

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

2 static slots of 8 bytes payload
## Motivational Example

### Applications

<table>
<thead>
<tr>
<th>Mode</th>
<th>Application</th>
<th>Data Size</th>
<th>Period</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)</td>
<td>1</td>
<td>8</td>
<td>(T_{bus})</td>
<td>100</td>
</tr>
<tr>
<td>(a_2)</td>
<td>2</td>
<td>4</td>
<td>(T_{bus})</td>
<td>80</td>
</tr>
<tr>
<td>(a_3)</td>
<td>3</td>
<td>4</td>
<td>(2T_{bus})</td>
<td>50</td>
</tr>
</tbody>
</table>

\(\alpha\) mode | \(w\) data size | \(p\) period | \(J\) performance

### Possible mode combinations

<table>
<thead>
<tr>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(a_3)</th>
<th>(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Online switch not possible

2 static slots of 8 bytes payload
### Motivational Example

#### Applications

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2 $T_{bus}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2 $T_{bus}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4 $T_{bus}$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8 $T_{bus}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$T_{bus}$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2 $T_{bus}$</td>
</tr>
</tbody>
</table>

- $\alpha$: mode
- $w$: data size
- $p$: period
- $J$: performance

#### Possible mode combinations

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Newly installed application

<table>
<thead>
<tr>
<th>$a_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Online switch not possible

2 static slots of 8 bytes payload

Licong Zhang/ RCS,TUM
## Motivational Example

### Applications

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th></th>
<th>$a_2$</th>
<th></th>
<th>$a_3$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$ w  $p$ $J$</td>
<td></td>
<td>$\alpha$ w  $p$ $J$</td>
<td></td>
<td>$\alpha$ w  $p$ $J$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8  $T_{bus}$ 100</td>
<td></td>
<td>1 4 2 $T_{bus}$ 100</td>
<td></td>
<td>1 8  $T_{bus}$ 100</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4  $T_{bus}$ 80</td>
<td></td>
<td>2 4 4 $T_{bus}$ 80</td>
<td></td>
<td>2 4  $T_{bus}$ 80</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4 2 $T_{bus}$ 50</td>
<td></td>
<td>3 4 8 $T_{bus}$ 50</td>
<td></td>
<td>3 4 2 $T_{bus}$ 50</td>
<td></td>
</tr>
</tbody>
</table>

- $\alpha$: mode
- $w$: data size
- $p$: period
- $J$: performance

#### Possible mode combinations

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 2</td>
<td></td>
<td></td>
<td>280</td>
</tr>
<tr>
<td>2</td>
<td>2 2 1 1</td>
<td></td>
<td></td>
<td>280</td>
</tr>
<tr>
<td>3</td>
<td>1 1 2 2</td>
<td></td>
<td></td>
<td>260</td>
</tr>
<tr>
<td>4</td>
<td>2 2 2 1</td>
<td></td>
<td></td>
<td>260</td>
</tr>
</tbody>
</table>

#### Newly installed application

<table>
<thead>
<tr>
<th>$a_4$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$ w  $p$ $J$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 4 2 $T_{bus}$ 100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **online switch not possible**
- **online activation not possible**

Although enough resource to map it on combination 1
Middleware

- Software architecture
  - Simple software architecture

- FlexRay parameters, FlexRay schedules and data-to-frame mapping statically configured offline
Middleware

**Software architecture**

- Proposed software architecture

![Diagram showing the software architecture with Application, Data Mapping, Operating System and FlexRay Controller layers]

- Middleware between application and OS layers
  - Data mapping
    - Data-to-frame mapping based on a configuration
**Middleware**

- **Software architecture**
  - Proposed software architecture

- Middleware between application and OS layers
  - Data mapping
  - Reconfiguration request
  - Configuration calculator
  - State management
  - Deployment management
Middleware

- **Application, task, message, mode and manifest**
  - Application $a_i(\alpha_i) = (T_i(\alpha_i), M_i(\alpha_i), J_i(\alpha_i))$
  - Task $\tau_j \in T_i$, message $m_j \in M_i$, performance $J_i \in J_i$, mode $\alpha_i$
  - Characteristics of $T_i$, $M_i$ and $J_i$ are known and provided by the application developer as application manifest
Middleware

- Application, task, message, mode and manifest
  - Application $a_i(\alpha_i) = (T_i(\alpha_i), M_i(\alpha_i), J_i(\alpha_i))$
  - Task $\tau_j \in T_i$, message $m_j \in M_i$, performance $J_i \in J_i$, mode $\alpha_i$
  - Characteristics of $T_i$, $M_i$ and $J_i$ are known and provided by the application developer as application manifest
  - An example in terms of communication

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$w$</th>
<th>$p$</th>
<th>$J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>$T_{bus}$</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$T_{bus}$</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>$2T_{bus}$</td>
<td>50</td>
</tr>
</tbody>
</table>
Middleware

- **Application, task, message, mode and manifest**
  - Application $a_i(\alpha_i) = (T_i(\alpha_i), M_i(\alpha_i), J_i(\alpha_i))$
  - Task $\tau_j \in T_i$, message $m_j \in M_i$, performance $J_i \in J_i$, mode $\alpha_i$
  - Characteristics of $T_i$, $M_i$ and $J_i$ are known and provided by the application developer as application manifest
  - An example in terms of communication

- **Configuration**
  - Application configuration $C_a = \{\alpha_i | a_i \in A\}$
  - Communication configuration $C_c = \{M_i | a_i \in A\}$
  - $C_c$ contains the mapping of messages to FlexRay schedule $m_j \rightarrow \Theta_j$

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$</th>
<th>$w$</th>
<th>$p$</th>
<th>$J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
<td>$T_{bus}$</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>$T_{bus}$</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>$2T_{bus}$</td>
<td>50</td>
</tr>
</tbody>
</table>
Middleware

- Data mapping
Data mapping

- For each static slot $S_j$, a FlexRay frame with $\Theta_{j, base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible.
Data mapping

- For each static slot $S_j$, a FlexRay frame with $\Theta_j,\text{base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible.
- Data are mapped into $\Theta_j,\text{base}$ by payload and slot multiplexing within an ECU.

$\Theta_j$

$m_j$

Data (Msg)

application layer scheduling

Signal

Frame

communication layer scheduling

Payload

Frame

Header

Trailer

Sender ECU

Receiver ECUs

$a_1$ $a_2$ $a_3$ $a_4$

$m_1$ $m_3$ $m_4$
Data mapping

- For each static slot $S_j$, a FlexRay frame with $\Theta_{j,base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible.
- Data are mapped into $\Theta_{j,base}$ by payload and slot multiplexing within an ECU.
### Data mapping

- For each static slot $S_j$, a FlexRay frame with $\Theta_{j,base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible.
- Data are mapped into $\Theta_{j,base}$ by payload and slot multiplexing within an ECU.

![Diagram](image-url)

<table>
<thead>
<tr>
<th>Sender ECU</th>
<th>Payload</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$m_1$</td>
<td>$m_1$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$m_2$</td>
<td>$m_3$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$m_3$</td>
<td>$m_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$m_4$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Receiver ECU</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
</tr>
<tr>
<td>$a_2$</td>
</tr>
<tr>
<td>$a_3$</td>
</tr>
<tr>
<td>$a_4$</td>
</tr>
</tbody>
</table>
Middleware

- **Re-configuration request**
  - Generates a request for reconfiguration
  - Can be triggered by pre-programmed sequence, application requiring mode switch or activation of newly installed application
Middleware

- **Reconfiguration request**
  - Generates a request for reconfiguration
  - Can be triggered by pre-programmed sequence, application requiring mode switch or activation of newly installed application

- **Configuration calculation**
  - To synthesize configuration $C_a = \{a_i | a_i \in A\}$ and $C_c = \{M_i | a_i \in A\}$, where the request and constraints are satisfied and overall performance maximized
  - Can be divided into a two layer problem
    - Layer one: all possible combinations of application modes are traversed
    - Layer two: to determine a feasible $C_c$
      - An exhaustive search
      - A classical bin packing ILP formulation for FlexRay slot packing [8]
Middleware

- Configuration deployment and state management

Diagram:
- Normal Operation
- Configuration Calculation
- Configuration Deployment
- Wait for Re-config

Arrows:
- Request
- Denial
- Reconfiguration request
- Reconfiguration finished
- New configuration ready
- Configuration deployment finished
Middleware

- Configuration deployment and state management

Normal Operation

actions
system state \[\text{NO}\]
config applied
Middleware

- Configuration deployment and state management

```
Normal Operation  ➔  Configuration Calculation

reconfiguration request

actions
system state
config applied
```

```
64T_{bus}
T_{bus}
... ...

NO CC

reconfiguration request
```
Middleware

- Configuration deployment and state management

Normal Operation → Configuration Calculation → Configuration Deployment

- request
- denial
- reconfiguration request
- new configuration ready

System state:
- NO
- CC
- CD

Old config:

K - 1
K

Actions:
- reconfiguration request
- new configuration ready

64T_{bus}
T_{bus}
Middleware

- Configuration deployment and state management

- Normal Operation
  - Configuration Calculation
    - Configuration Deployment
      - Wait for Re-Config

- reconfiguration request
- new configuration ready
- configuration deployment finished

- request denial

- NO CC CD WR

- old config
Configuration deployment and state management

- Normal Operation
- Configuration Calculation
- Configuration Deployment
- Wait for Re-config

- \(64T_{bus}\)
- \(T_{bus}\)

System state:
- NO
- CC
- CD
- WR
- NO

Configuration applied:
- Old config
- New config
Case Study

- **System Description**
  - 2 ECUs, 4 applications

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
<td>$J$</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>$T_{bus}$</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>2$T_{bus}$</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>2$T_{bus}$</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>2$T_{bus}$</td>
<td>25</td>
</tr>
</tbody>
</table>
Case Study

- **System Description**
  - 2 ECUs, 4 applications

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$w$</td>
<td>$p$</td>
<td>$J$</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>$T_{bus}$</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>$2T_{bus}$</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>$2T_{bus}$</td>
<td>25</td>
</tr>
</tbody>
</table>

- **Reconfiguration requests**

<table>
<thead>
<tr>
<th>step</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>$a_1 \rightarrow$ on</td>
<td>$a_2 \rightarrow$ on</td>
<td>$a_3 \rightarrow$ on</td>
<td>$a_3 \rightarrow$ 1</td>
<td>$a_4 \rightarrow$ on</td>
<td>$a_4 \rightarrow$ 1</td>
<td>$a_3 \rightarrow$ off</td>
<td>$a_3 \rightarrow$ on</td>
</tr>
</tbody>
</table>
Case Study

- **System Description**
  - 2 ECUs, 4 applications
  
<table>
<thead>
<tr>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>α</td>
<td>w</td>
<td>p</td>
<td>J</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>T_{bus}</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>2 T_{bus}</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>2 T_{bus}</td>
<td>25</td>
</tr>
</tbody>
</table>

- **Reconfiguration requests**
  
<table>
<thead>
<tr>
<th>step</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>a₁ → on</td>
<td>a₂ → on</td>
<td>a₃ → on</td>
<td>α₃ → 1</td>
<td>a₄ → on</td>
<td>α₄ → 1</td>
<td>a₃ → off</td>
<td>a₃ → on</td>
</tr>
</tbody>
</table>

- **Implementation**
  - EB6120 [11] as ECUs
  - Software developed with Simulink and SIMTOOLS/SIMTARGET [12]
  - Middleware implemented with Simulink
  - Configuration calculation mapped on a dedicated ECU
Experimental Results

- Re-configuration according to request sequence
  - Application and overall performance

<table>
<thead>
<tr>
<th>step</th>
<th>request</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$a_1 \rightarrow \text{on}$</td>
</tr>
<tr>
<td>2</td>
<td>$a_2 \rightarrow \text{on}$</td>
</tr>
<tr>
<td>3</td>
<td>$a_3 \rightarrow \text{on}$</td>
</tr>
<tr>
<td>4</td>
<td>$a_3 \rightarrow 1$</td>
</tr>
<tr>
<td>5</td>
<td>$a_4 \rightarrow \text{on}$</td>
</tr>
<tr>
<td>6</td>
<td>$a_4 \rightarrow 1$</td>
</tr>
<tr>
<td>7</td>
<td>$a_3 \rightarrow \text{off}$</td>
</tr>
<tr>
<td>8</td>
<td>$a_3 \rightarrow \text{on}$</td>
</tr>
</tbody>
</table>
Experimental Results

- **Online configuration synthesis time**
  - EB6120 as ECU
  - Software implementation with Simulink Matlab-embedded functions

<table>
<thead>
<tr>
<th>step</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. apps</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3*</td>
<td>4</td>
<td>4*</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

*one application has a fixed mode*
Concluding Remarks

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for multi-mode applications and newly activated applications
Concluding Remarks

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for multi-mode applications and newly activated applications

- **Approach**
  - A middleware layer with
    - Reconfigurable data-to-schedule mapping
    - Online configuration calculation and deployment
  - Achieves online re-allocation of communication resources with an ECU
Concluding Remarks

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for multi-mode applications and newly activated applications

- **Approach**
  - A middleware layer with
    - Reconfigurable data-to-schedule mapping
    - Online configuration calculation and deployment
  - Achieves online re-allocation of communication resources with an ECU

- **Outlook**
  - Extension to support other communication protocols
  - Extension to allow features like re-routing of messages
  - More efficient configuration calculation algorithms


[12] SIMTOOLS/SIMTARGET. www.simtools.at
The End

Many thanks

Q/A