

A Flexible Slotting-Scheme for TDMA-based Protocols

Jens Chr. Lisner

`lisner@informatik.uni-essen.de`

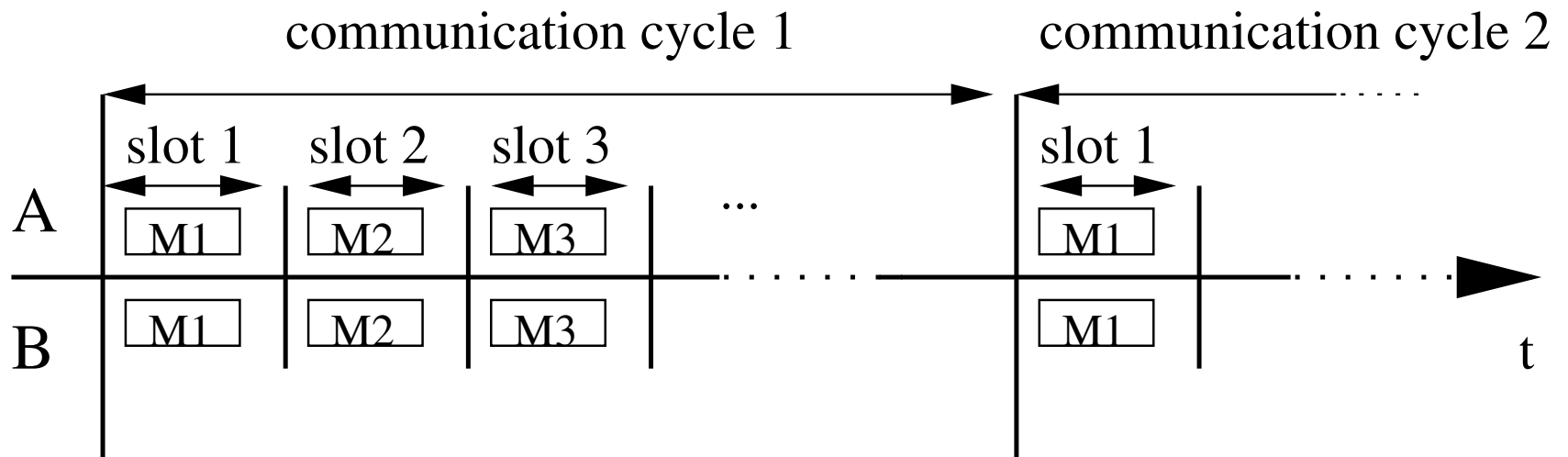
ICB / University of Duisburg-Essen

Motivation

TDMA: “time division multiple access”

- Communication is done in *cycles*
- Cycles are divided into *slots*
- The slots are assigned to the senders

Example: TTP



Motivation II

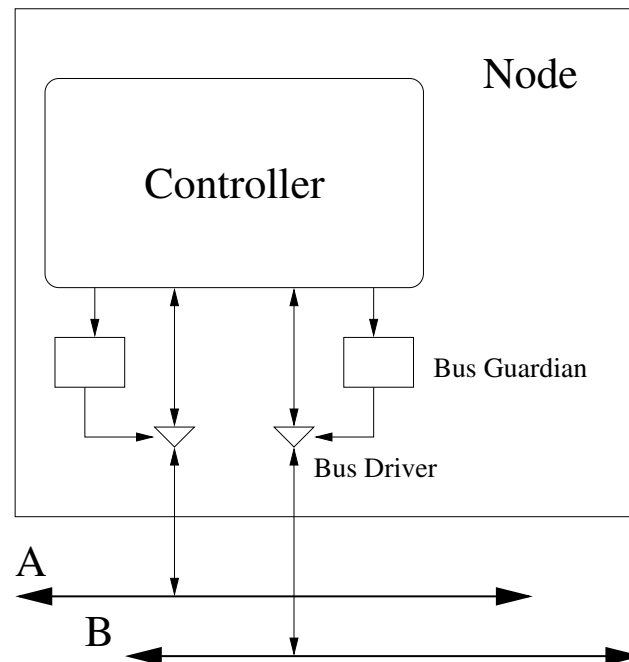
Advantage:

Media can be protected by bus guardians.

Bus Guardian:

Denys access to the media, if the sender is not scheduled for this slot.

→ Fault tolerant distributed real-time communication possible



Motivation III

Drawback:

- Schedule is statically preconfigured
- Temporary additional slots for senders are not available

Example: Exception Handling

Exceptions should be thrown w/o disturbing the “regular” communication

Motivation IV

Possible solutions with TTP:

Configure additional slots

- wastes bandwidth
- bus guardians have to follow dynamic schedule → more complex component required

FlexRay protocol:

- *static segment*: TTP like scheduling
- *dynamic segment*: Flexible slotting scheme, but bus guardians are switched off!
→ No fault tolerant communication

Goal

1. All controllers are able to use one additional slot if necessary within a communication cycle.
2. Fault tolerant communication
 - One faulty channel: message corruption, messages are delivered to a subset of receivers (byzantine faults possible)
 - One faulty controller
 - Both faulty channel and controller

New Slotting Scheme

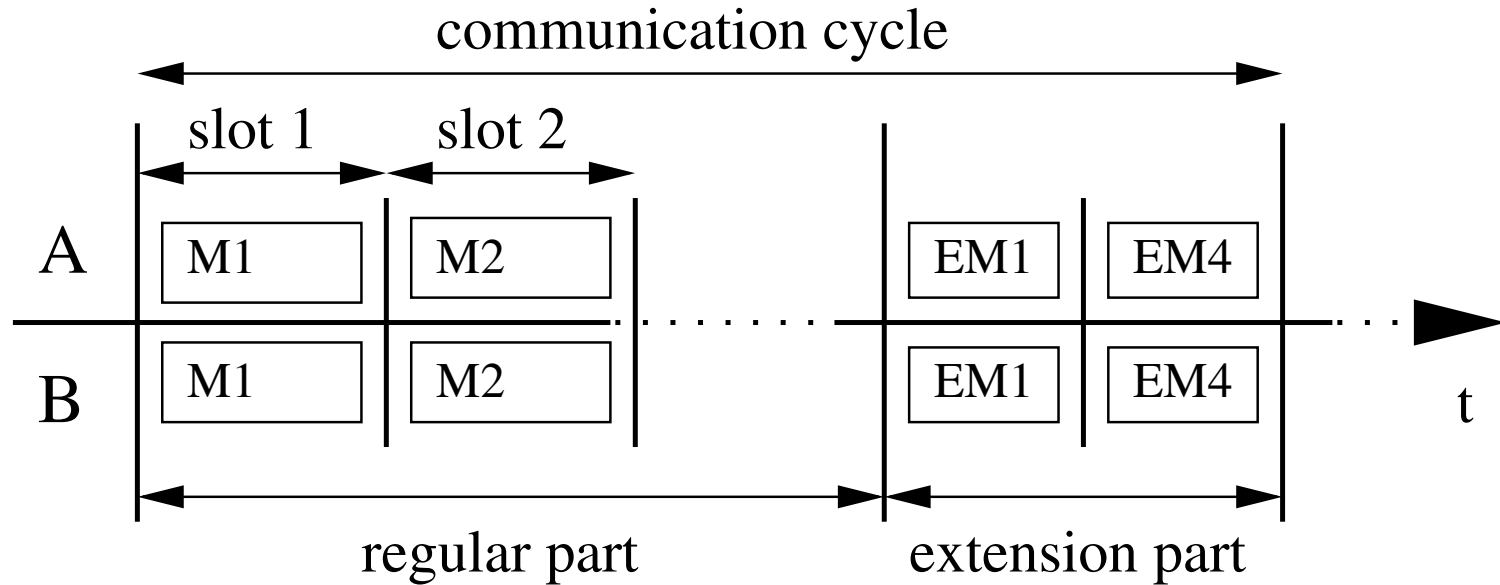
Communication cycle is splitted

- Regular part:
 - Static scheduling
 - Controllers can request additional slot
 - At end: Agreement about which controllers will send necessary
- Extension part
 - Contains additionally requested slots
 - Length depends on the number of allocated slots
 - Is left out, if no slots requested

New Slotting Scheme II

M_i = Message of controller i

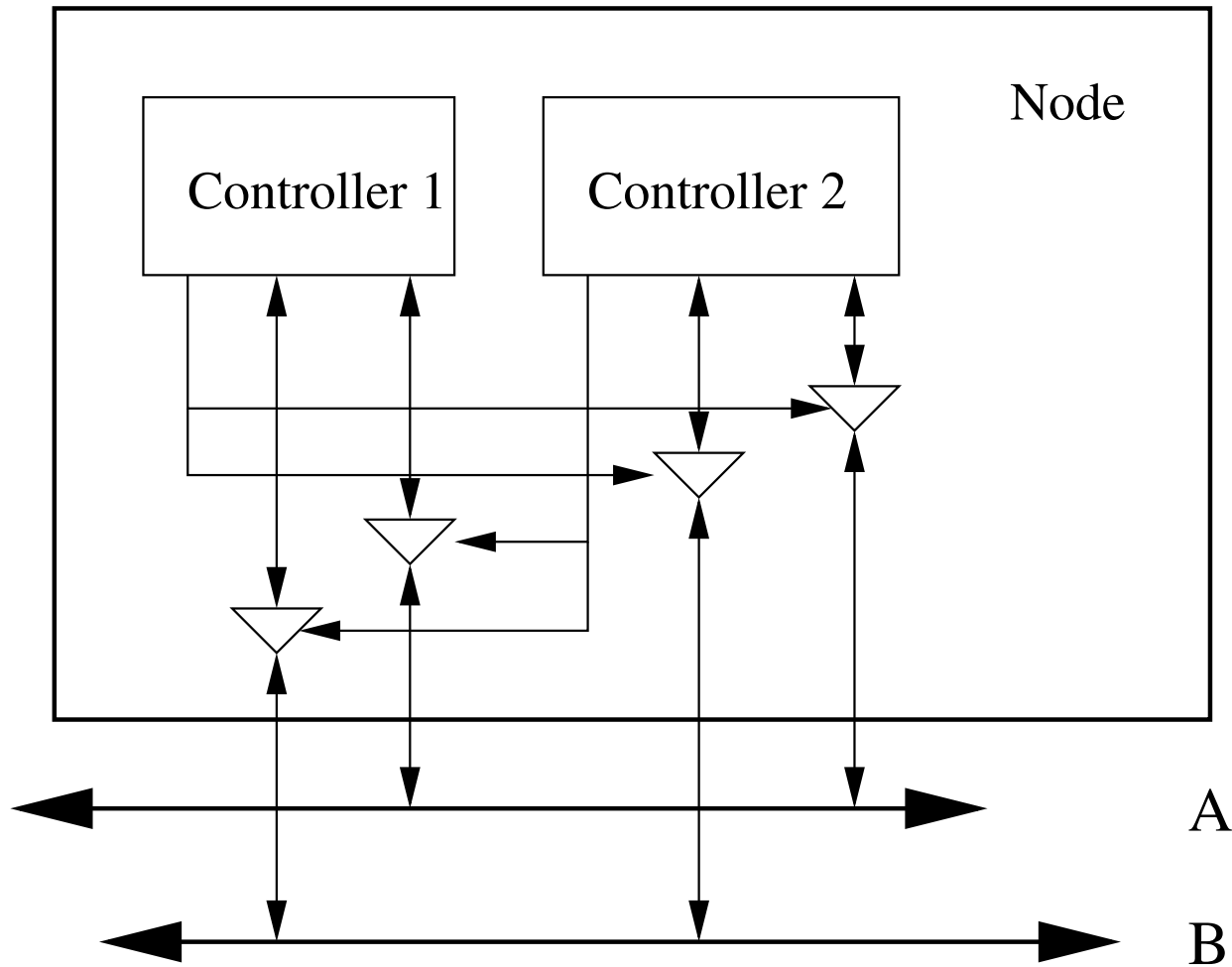
EM_i = Message in extension part



Controllers 1 and 4 requested a slot in the extension.

Architectural Change

A controller is guarded by its “neighbour.”



Fault-Tolerant Operation

Faulty controller together with faulty channel:

- Controller sends corrupted message on fault-free channel
- Controller sends correct message on faulty channel
- Controller requests slot
- Faulty channel delivers message to a subset of all controllers

→

- Controllers receiving message, decide to schedule the faulty controller for extension part
- Controllers not receiving message are using a default rule: Controller is not scheduled.

Fault-Tolerant Operation II

Agreement:

Controller sends on both channels in different slots!

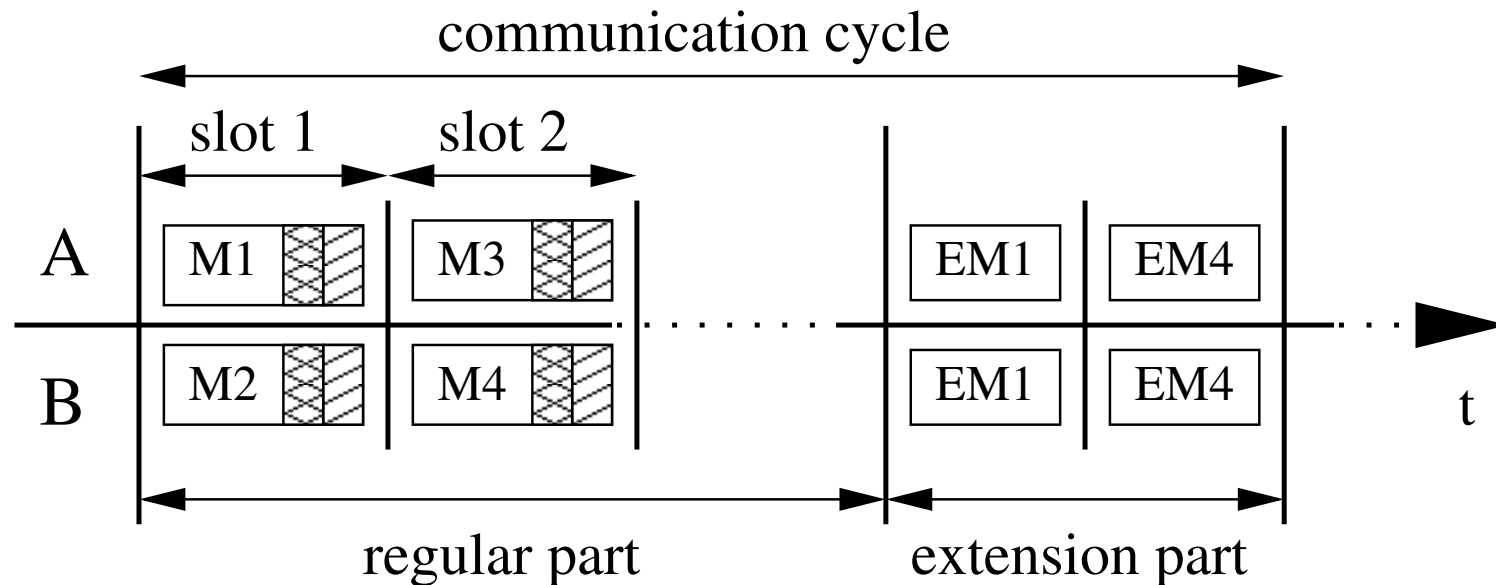
The following controllers repeat the information gathered so far.

 slot request

M_i = Message of controller i

 schedule suggestion

EM_i = Message in extension part



Fault-Tolerant Operation III

Messages

- Contain a local request
- Contain an “estimated schedule” vector

Estimated schedule

Current state of the agreement process.

“What the controller thinks what will happen.”

Updated with every incoming message

Slot Overloading

Real time requirement:

Fixed cycle length!

Slot overloading:

- Extension has fixed length
- Slots can be used if required
- Schedule is chosen dynamically

Conclusion

- Existing fault-tolerant TDMA-based protocols are not suitable for applications requiring both flexibility and fault-tolerance.
- It is possible to develop a dynamic slotting scheme for critical real-time applications.
- Fault-tolerance can be assured.