Efficiency of Dynamic Arbitration in TDMA Protocols

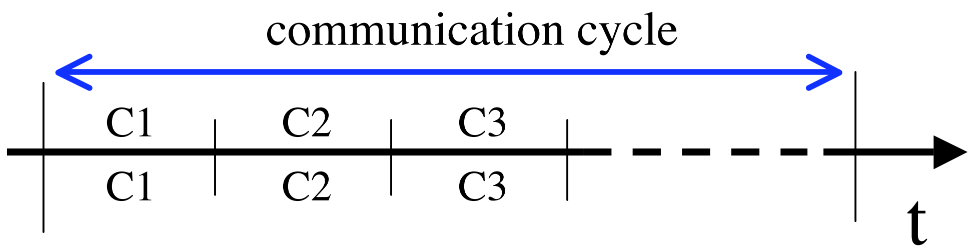
April 22, 2005

Jens Chr. Lisner
University of Duisburg-Essen

Introduction

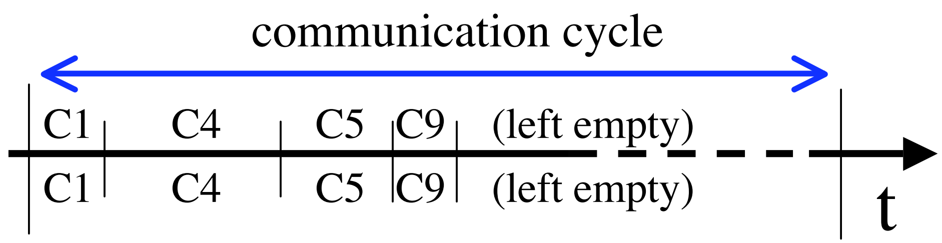
Arbitration methods in TDMA-based protocols

Static arbitration



- fixed length of slots
- fixed schedule of senders configured

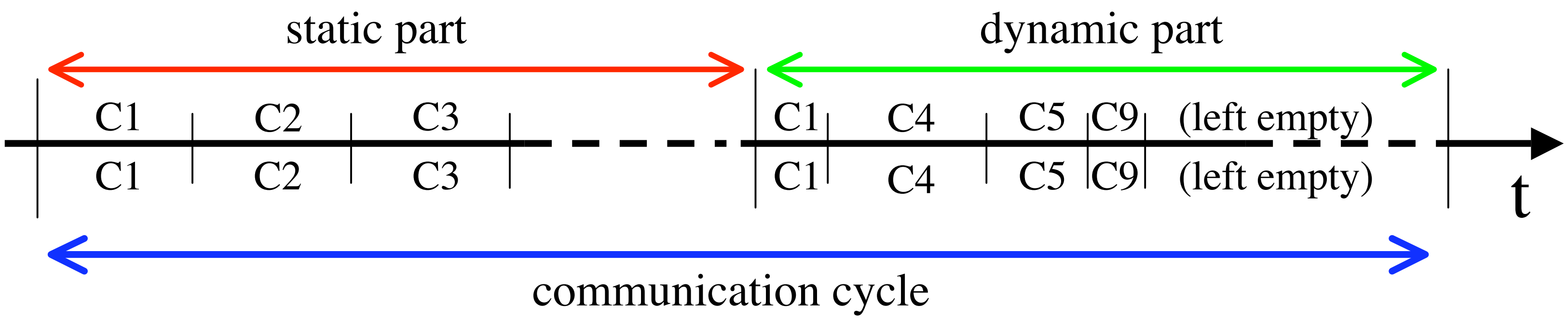
Dynamic arbitration



- slots have dynamic length
- schedule determined at runtime for every cycle

Introduction

<u>Mixed-mode protocols provide both methods</u>



- Improved flexibility
- Ability to send additional information
- Example: exception handling

Fault-tolerance

<u>Common problem: “babbling idiot” fault</u>

Solution:
Guardians protecting the channels from faulty controllers

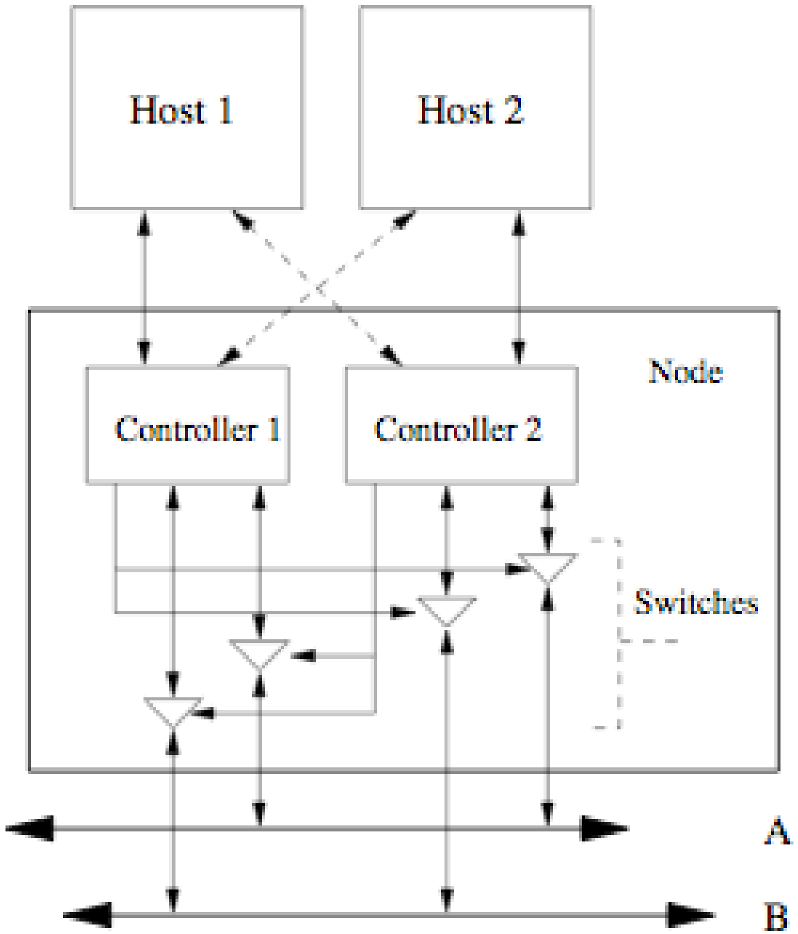
- Independent guardians
  Each node has one guardian for each channel.
- Central guardians
  Guardians reside on a hub and are controlled by a protocol controller.

Fault-toerance

<u>Controllers guard each other</u>

- Each controller acts as guardian for its neighbour
- Controllers are fully independent
- Controllers serve different hosts

Guaranteed fail-silent behaviour in case of controller faults.



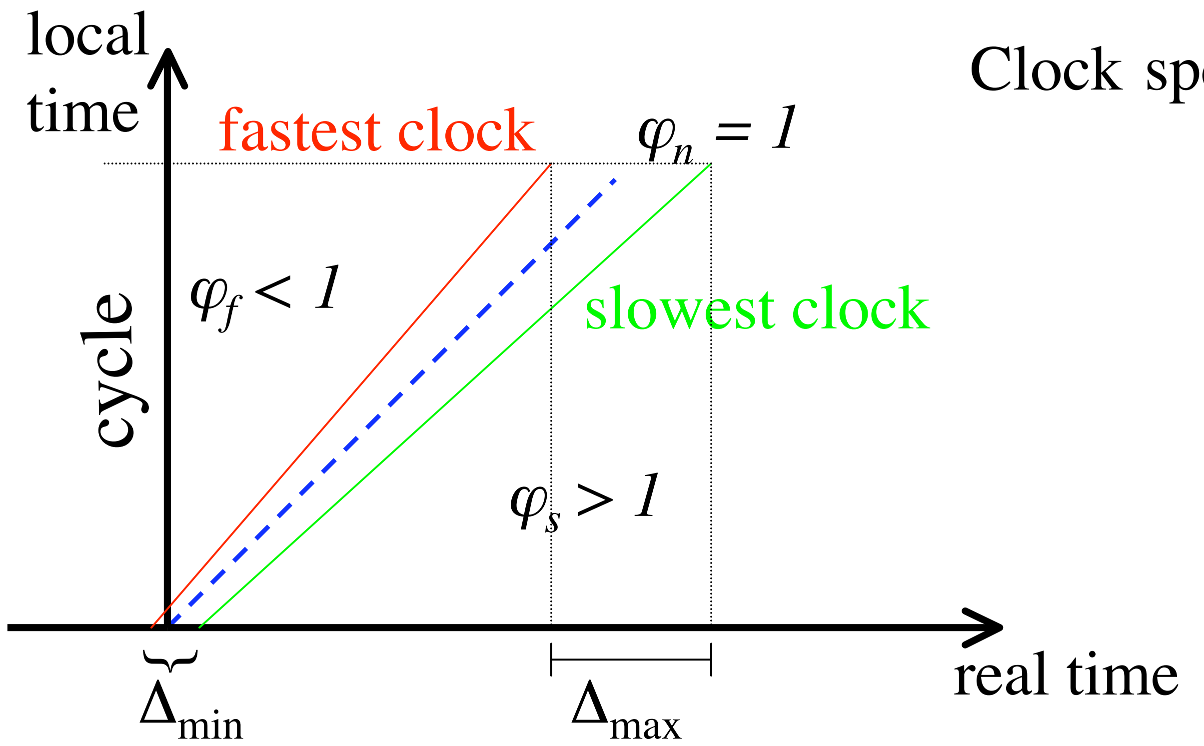
The Tea Protocol

- Mixed-mode media access
  - regular part (static)
  - extension part (dynamic)
- Double-controller architecture
- Extension schedule is determined in regular part
- “Agreement-based scheduling”
- Fault-tolerant and efficient solution

Time Model

r_j Deviation from nominal frequency of oscillator j (ppm)

Clock speed $\varphi_j = 1 + r_j$

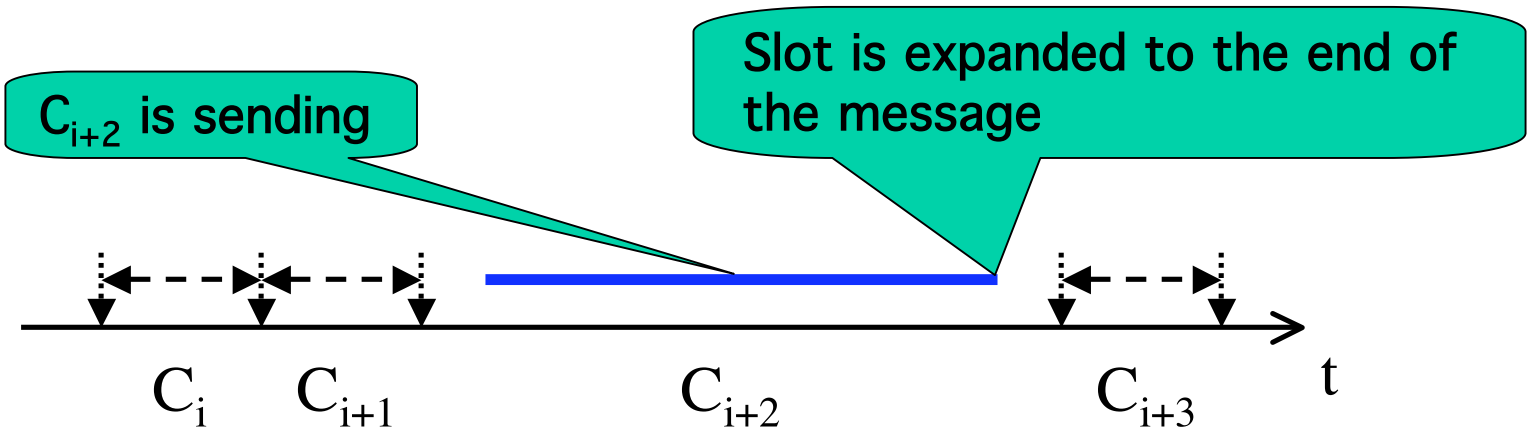


Media Access Methods

Minislotting with assigned slots

Pre-configured schedule:

- Controllers are statically assigned to minislots
- Controllers may send or not

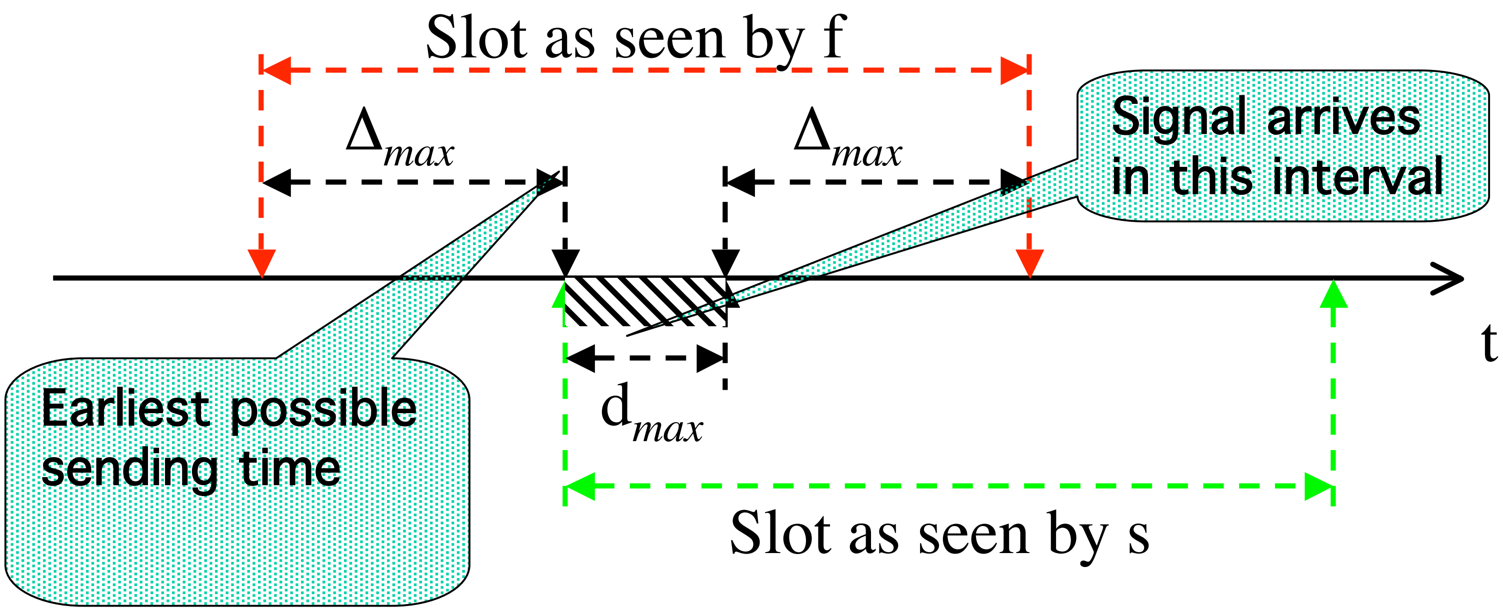


Media Access Methods

Minislotting with assigned slots:

Timing constraints for slot of controller f

d_{max} maximum signal delay



Media Access Methods

Minislotting with assigned slots

Minimum minislot length $\lambda_{minislot} \geq 2\Delta_{max} + d_{max} + \varepsilon$

ε small error term (rounding error, descretization, …)

Slot length if controller is sending

$\lambda_{message}(i)$ length of message of controller i in real time

$$\lambda_{slot}(i) = \left(\left\lceil \frac{\lambda_{message}(i)}{\lambda_{minislot}} \right\rceil + 1 \right) \lambda_{minislot} \geq 2\lambda_{minislot}$$

Media Access Methods

Minislotting with assigned slots

T_{as} Overhead caused by the arbitration method

m_{last} last minislot

Total length of dynamic part used for transmission

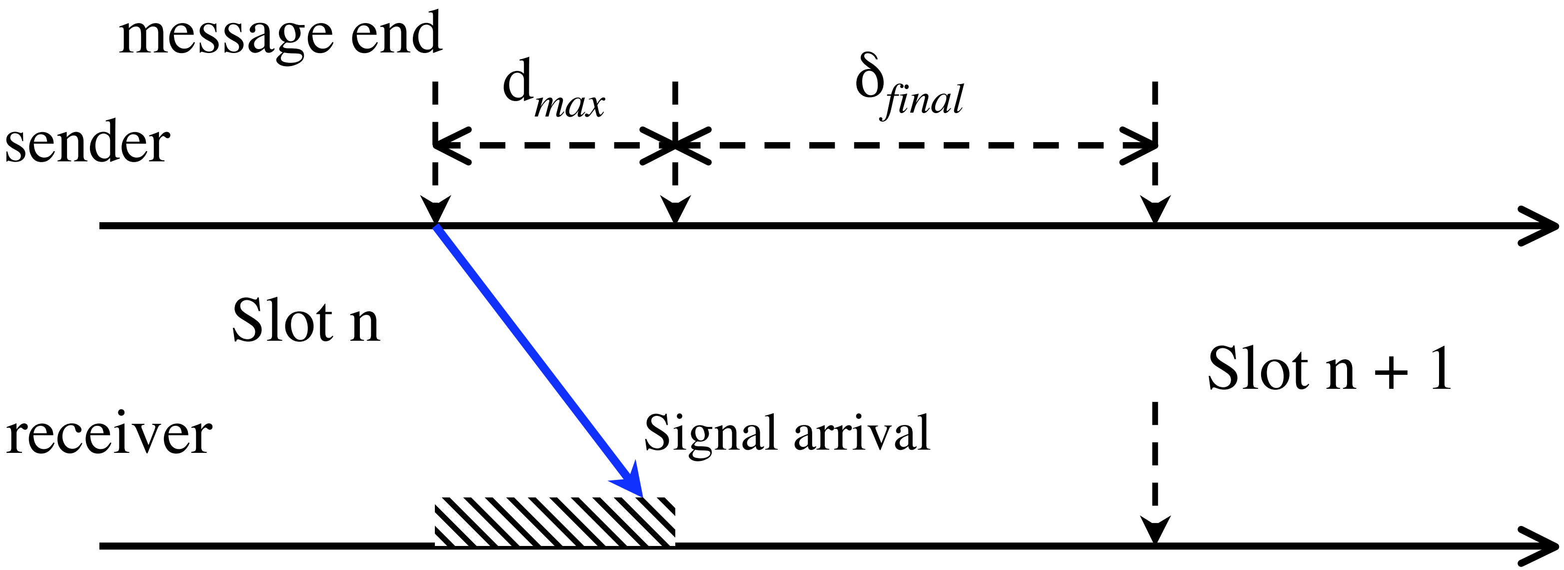
Length of used slots

Overhead at start and end of used slots

$$T_{as} = m_{last}\lambda_{minislot} - \sum_{i=1}^{k}\lambda_{slot}(i) + k2\Delta_{\max}$$

Media Access Methods

Minislotting with relaxed timing



idle state detected after tranmission end: $d_{max} + \delta_{final} + \varepsilon$

Media Access Methods

Minislotting with relaxed timing

T_{ls}	Overall overhead	k	last index of used slots
		h	number of used slots

sum of unused minislots

$$T_{ls} = (k - h)\lambda_{minislot} + h(\delta_{final} + d_{max} + \varepsilon)$$

end of transmission detection

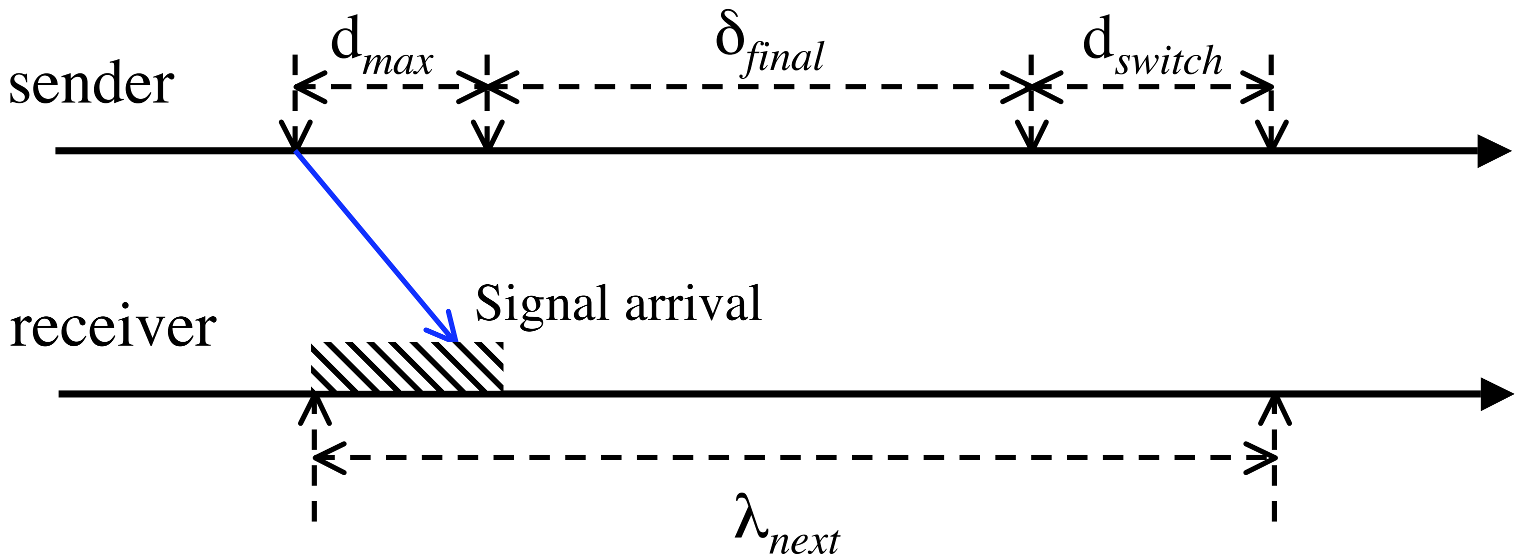
Improvements:

δ_{final} is expected to be lower than Δ_{max}

Media Access Methods

Agreement-based scheduling

dynamic slot length



δ_{final} not necessary, if message reaches maximum length

Media Access Methods

Agreement based scheduling

Schedule previously known
⇒ No additional arbitration mechanisms required

h total number of controllers sending
k number of controller which do not fully utilize maximum slot length ($k \leq h$)

$$T_{ts} = \sum_{h} \left(d_{max} + d_{switch} + \varepsilon \right) + \sum_{k} \delta_{final} + T_{reg}$$

additional overhead in regular part

Fault-tolerance

Tea is able to tolerate double faults

Possible behaviour of ...

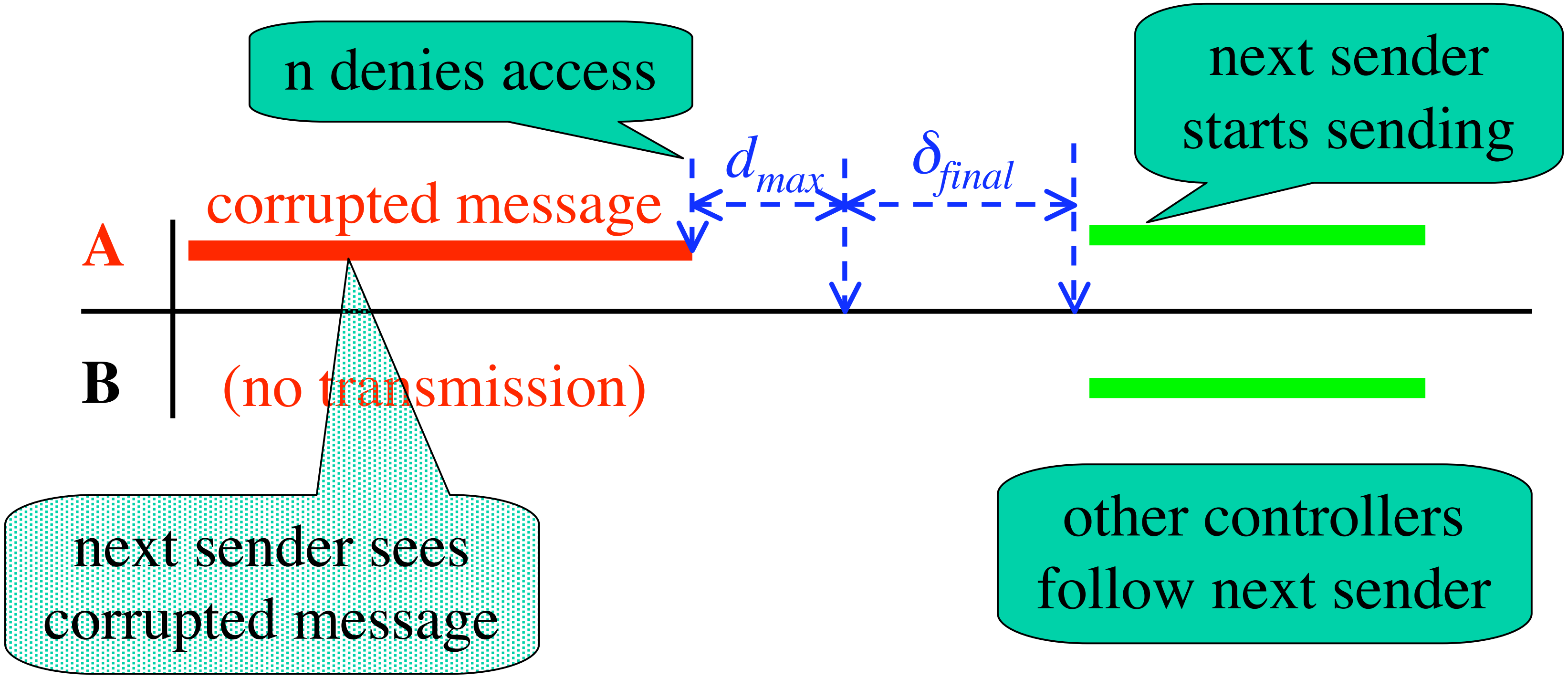
- Faulty controller
  - fail-silent
  - corrupted messages
  - "babbling idiot"
- Faulty channel
  - message corruption
  - byzantine behaviour

Messages corruption
- missing messages
- messages with invalid CRC (or similar protection)

Fault-tolerance

Case 1:

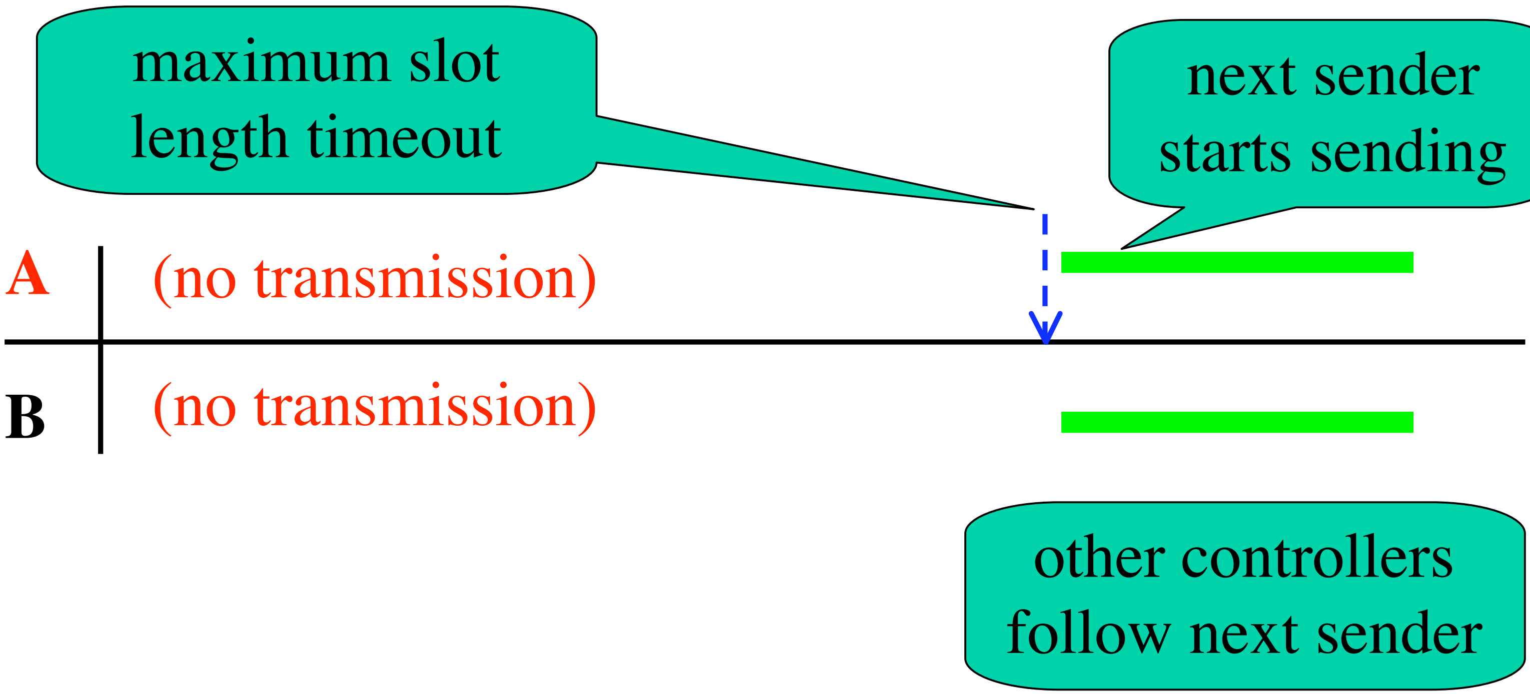
next sender receives transmission on faulty channel



Fault-tolerance

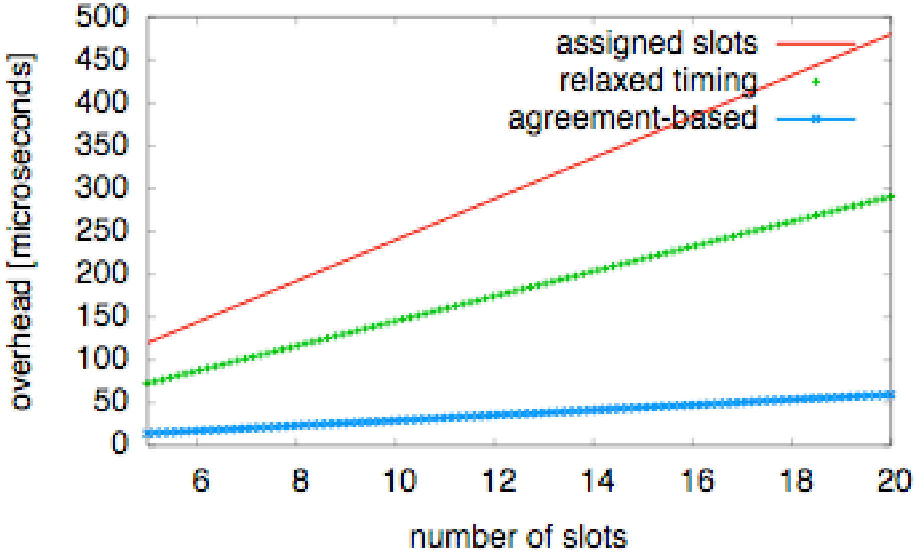
Case 2:

next sender receives no transmission on faulty channel



Examples

Scenario 1: Maximum utilization

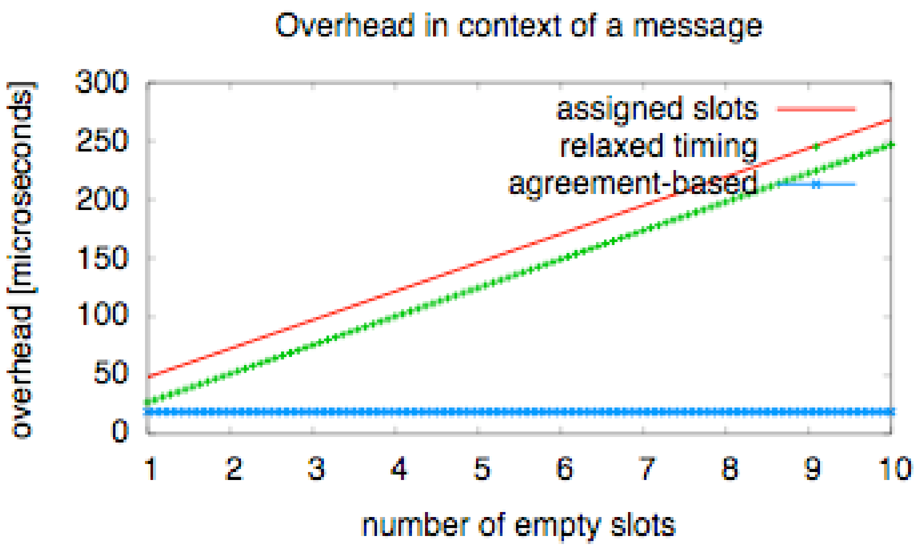


cycle length	160 ms
Δ_{max}	12 μs
ε	1 μs
d_{max}	0.44 μs
δ_{final}	1.1 μs
d_{switch}	0.25 μs

- High Δ_{max} dominates in assigned slots and relaxed timing methods
- Network constants d_{max} and δ_{max} have minimal impact on overhead
- Improve clock synchronization for assigned slots and relaxed timing to reduce overhead

Examples

Scenario 2: Overhead caused by empty slots



- Overhead in assigned slots and relaxed timing grow with number of empty slots (mainly depending on Δ_{max} in longer cycles)
- Constant overhead for agreement-based scheduling

Conclusion

- Efficient solution for mixed-mode TDMA protocols available
- The Tea protocol uses pre-determined schedules for dynamic arbitration
- Agreement method overhead depends mainly on network constants
- Methods with assigned slots and relaxed timing require accurate clocks
- Agreement methods needs additional overhead for majority voting
- Agreement method requires minimum overhead compared to other methods