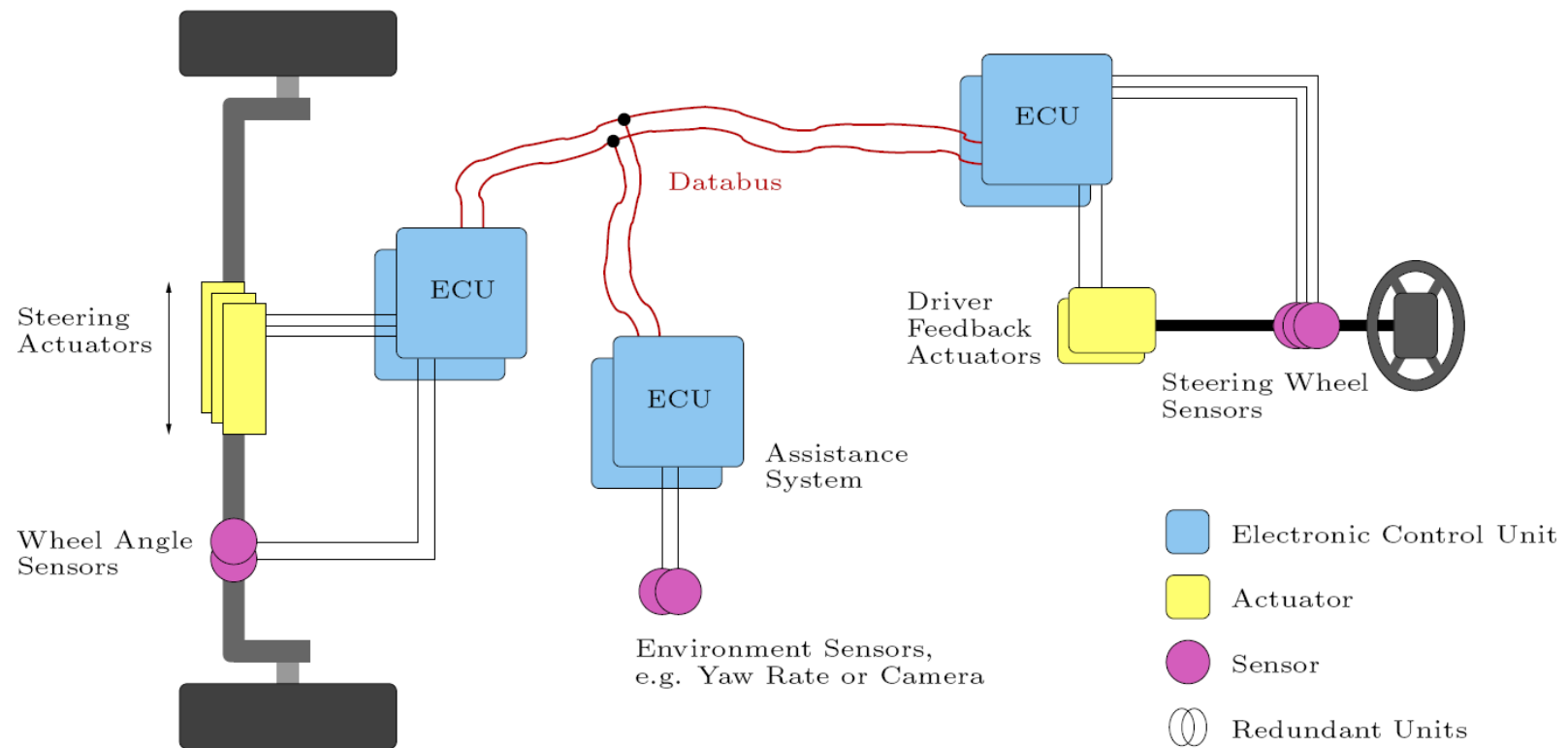


Fault Tolerance Analysis of the FlexRay Startup Procedure Using Model Checking

Sven Bünte <sven@vmars.tuwien.ac.at>

Motivation

- X-by-Wire Systems



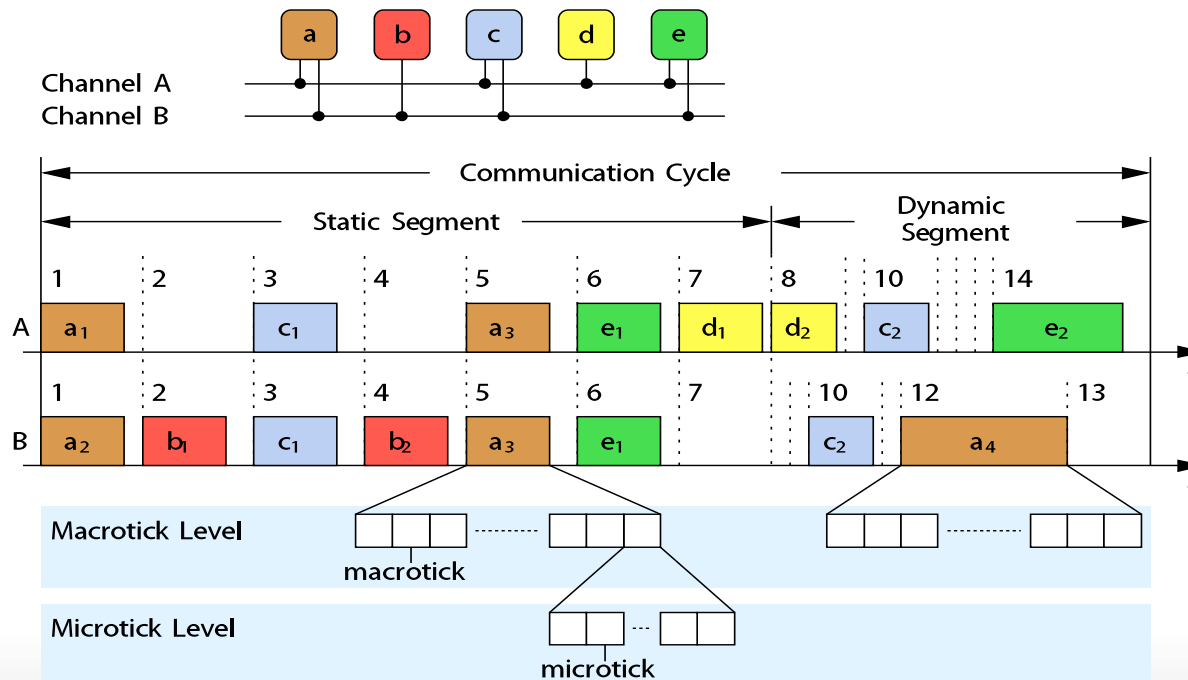
FlexRay: Overview

- Communication Protocol
 - Time and event triggered
 - Deterministic
 - Fault tolerant
- Application in the automotive domain
- Data rate of 10 Mbit/s
- Intended to support x-by-wire technology



FlexRay: Medium Arbitration

- Follows the *Time Division Multiple Access (TDMA)* principle
- Includes ideas from the *ByteFlight* protocol




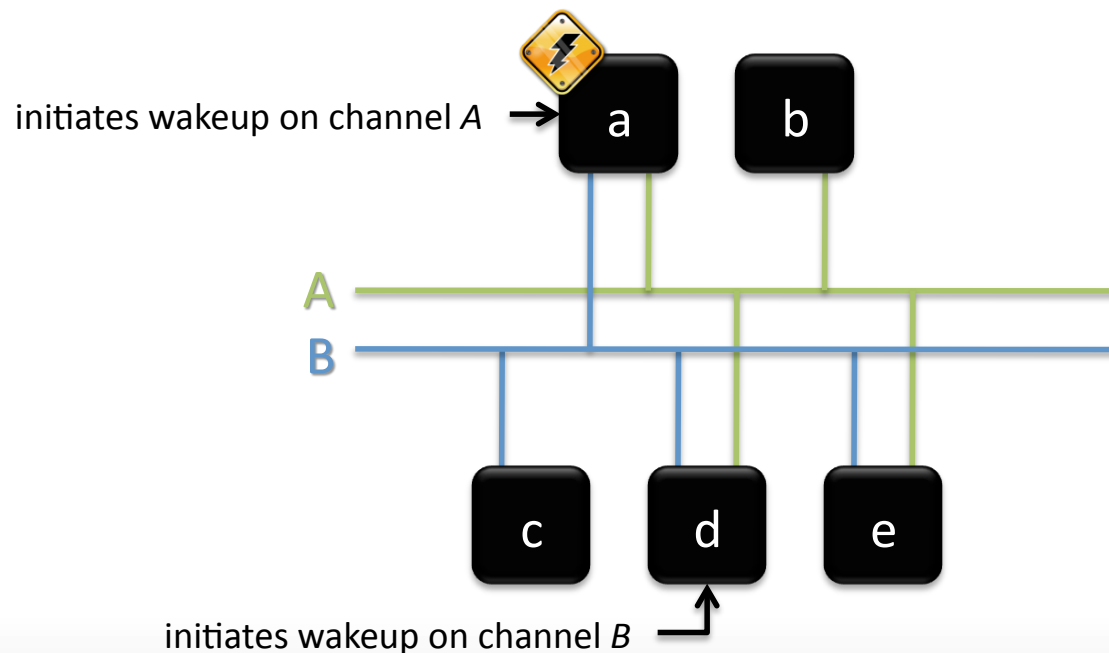
FlexRay: Medium Arbitration




- Follows the *Time Division Multiple Access (TDMA)* principle
- Includes ideas from the *ByteFlight* protocol
- Initialization of TDMA schedule complex
 - **Really deterministic?**
 - **Really fault-tolerant?**

The Wakeup Procedure

- Cluster nodes can enter a sleep state to save energy 
- Wakeup transforms all nodes to a *ready* state

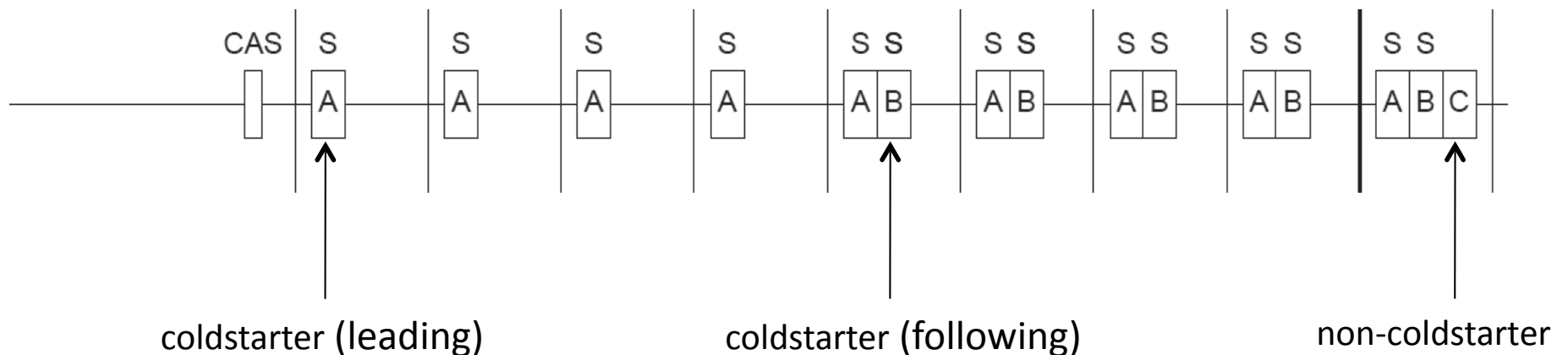


The Wakeup Procedure

- Cluster nodes can enter a sleep state to save energy 
- Wakeup transforms all nodes to a *ready* state
 - Essential for TDMA initialization to start
 - Does the procedure always terminate in a bounded time interval?

The Startup Procedure

- Initializes synchronized communication w.r.t. TDMA schedule
- Master clock principle not fault tolerant
→ Set of master clock nodes: *Coldstarters*



Specification and Description Language (SDL)

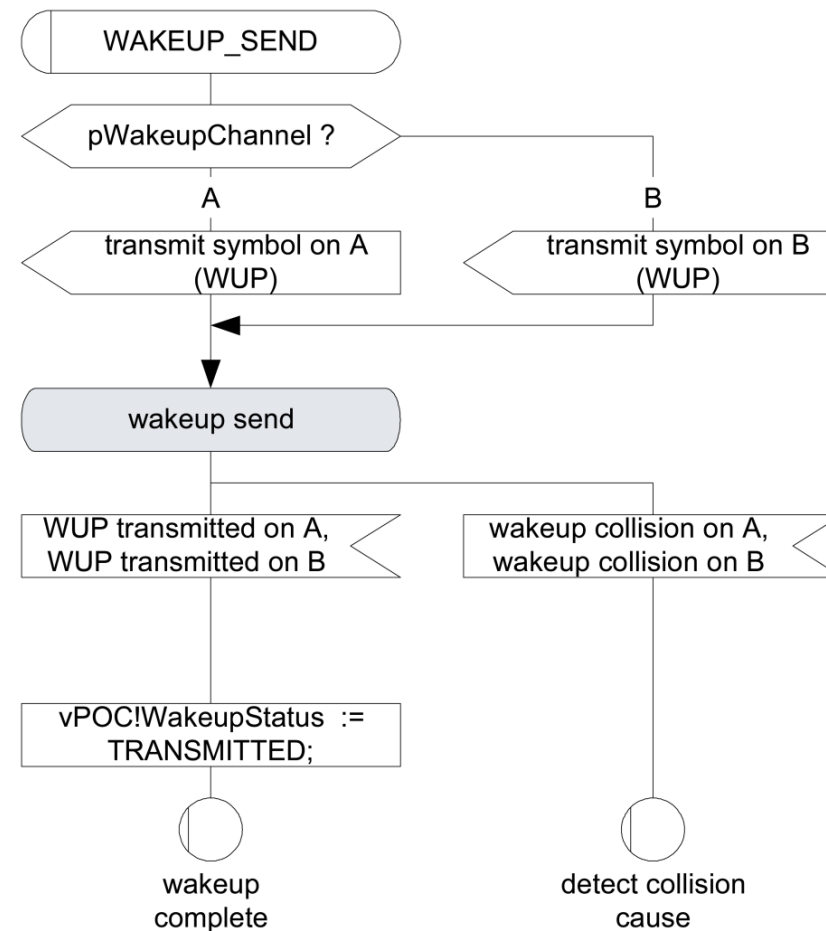
- FlexRay specified via SDL
- SDL originally intended for telephony applications (hardware)
- To describe the behavior of communicating processes
- FSM approach
- Includes real-time characteristics
- Discrete and bounded

Specification and Description Language (SDL)



Real Time Systems Group

Example from
FlexRay specification:

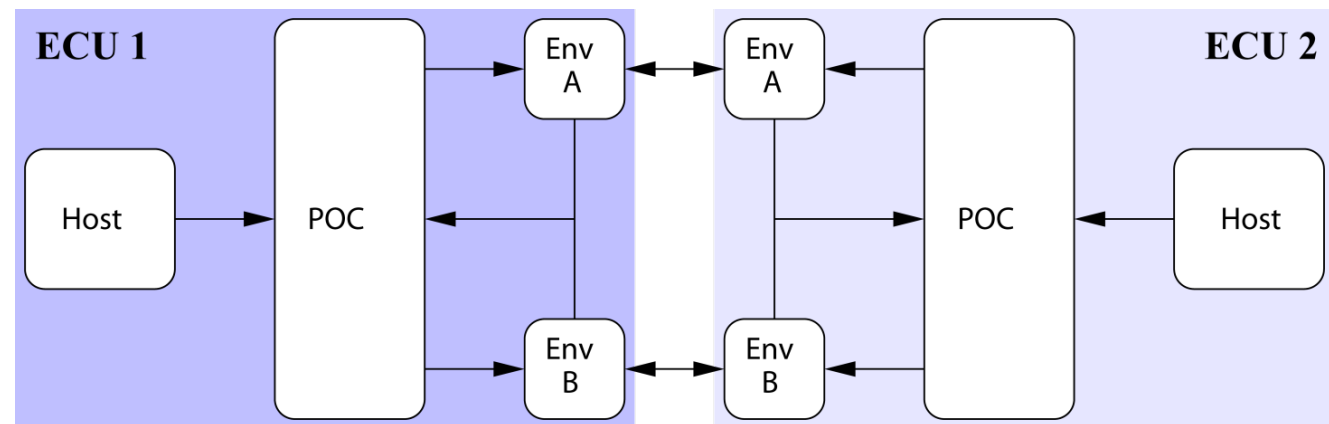


From SDL to DT-PROMELA/DT-SPIN

- DT-PROMELA, Discrete Time PROcess MEta LAnguage
 - Input language for model checker DT-SPIN
 - based on *C* and *CSP*
 - Intended to model communicating concurrent processes
 - Models are strictly discrete and bounded
 - Able to model discrete time
 - Automatic translation from SDL to DT-Promela (`sdl2if` and `if2p1`)
- Translations were done by hand though:
 - Only graphical SDL specification of FlexRay available
 - **Further optimizations and abstractions are essential**

Experimental Results: Wakeup

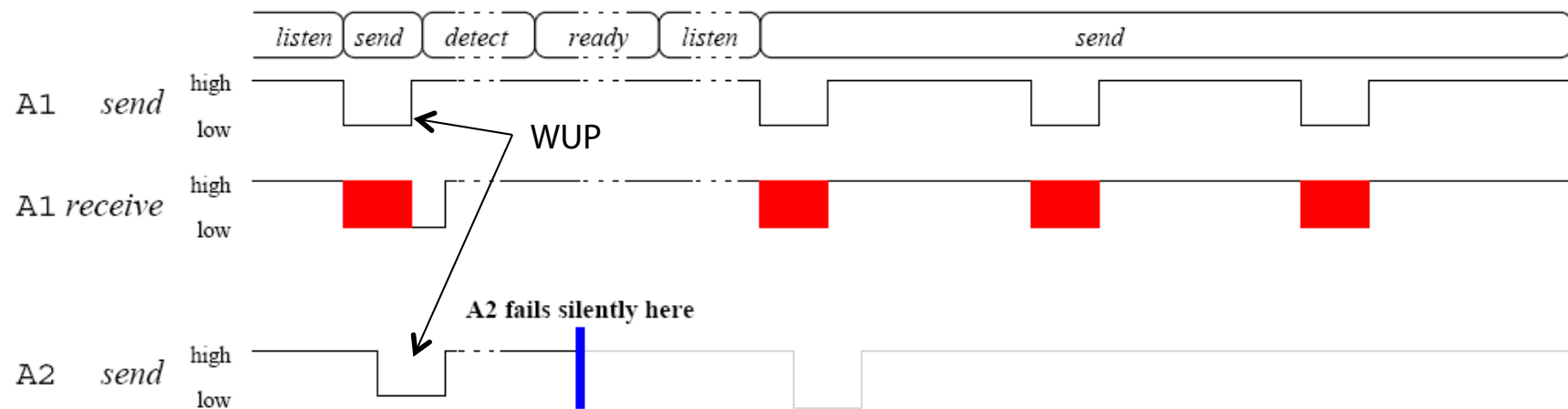
Model Architecture:



- Fault-free scenario: Model checking results
 - WUP is always transmitted after two cycles
 - Host is notified correctly about transmission
 - Prerequisite: WUP has to consist of at least 3 wakeup symbols
 - Verification takes half a second

Experimental Results: Wakeup

- Fail-silent/resetting scenario
 - System wakeup always successful in the fail-silent scenario if host configured properly



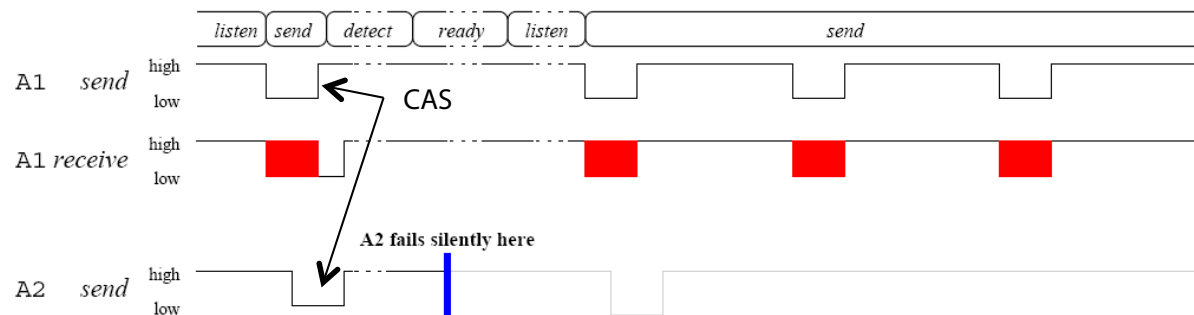
- A resetting node can cause a non-terminating wakeup

Experimental Results: Wakeup

- CAS-Babbler

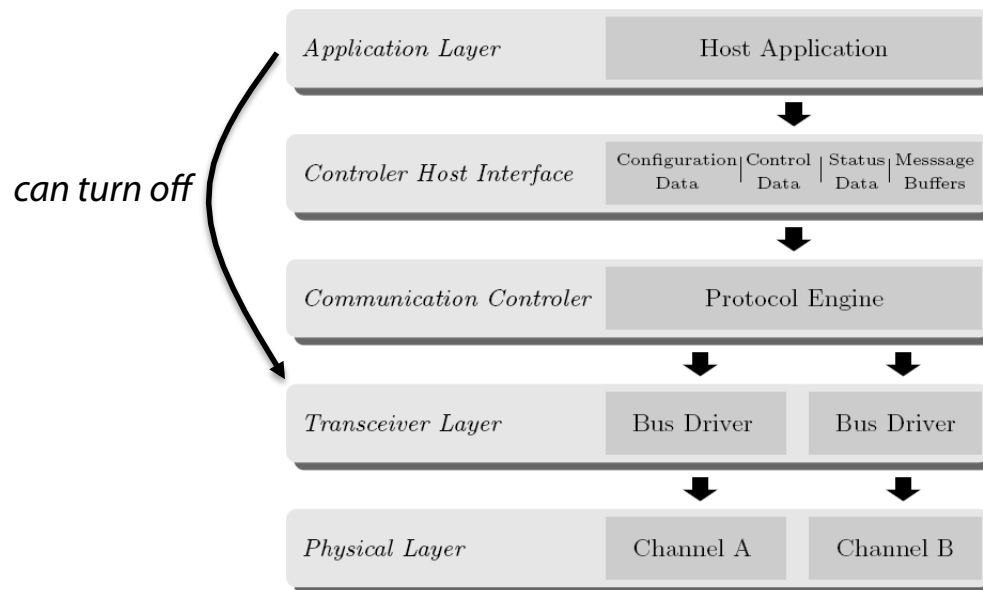
- Violation of timeliness properties:

- Error cycle from the “resetting scenario” can be reconstructed with a CAS instead of a WUP
 - A very high babbling frequency occupies the physical medium totally



Experimental Results: Wakeup

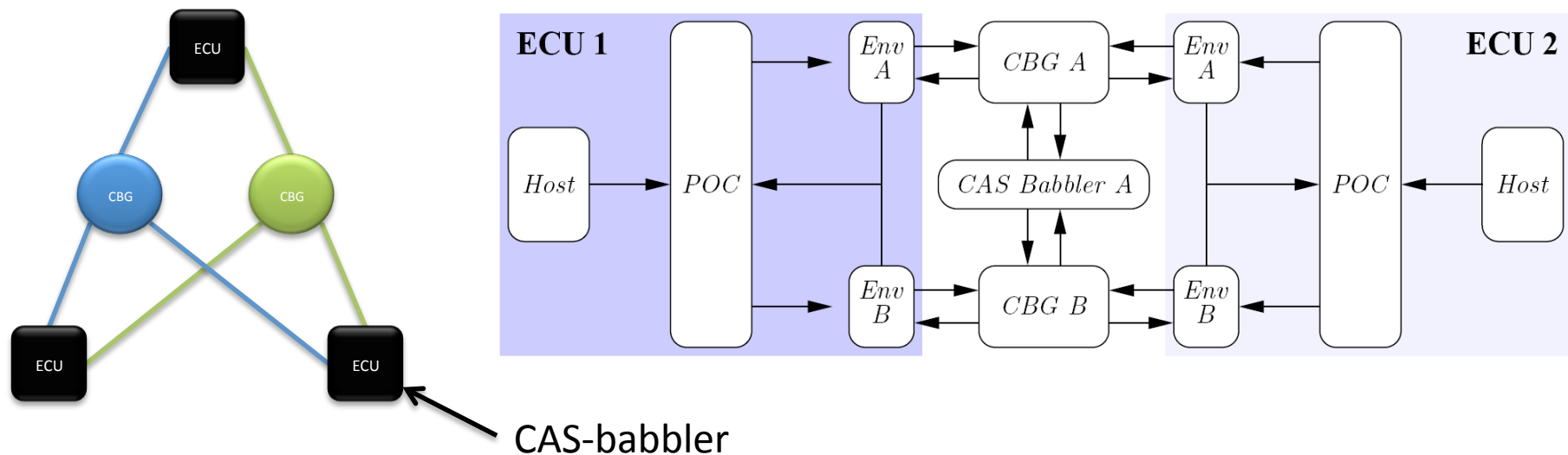
- Host Interaction
 - FlexRay offers an optional interface between the host application and the bus driver:



- Nodes turn themselves off if no progress is detected and enter with an individual offset
- Assumes that the Controller Host Interface provides correct information to the host application
- Timeliness verified by SPIN
- **Increases fault-tolerance!**

Experimental Results: Wakeup

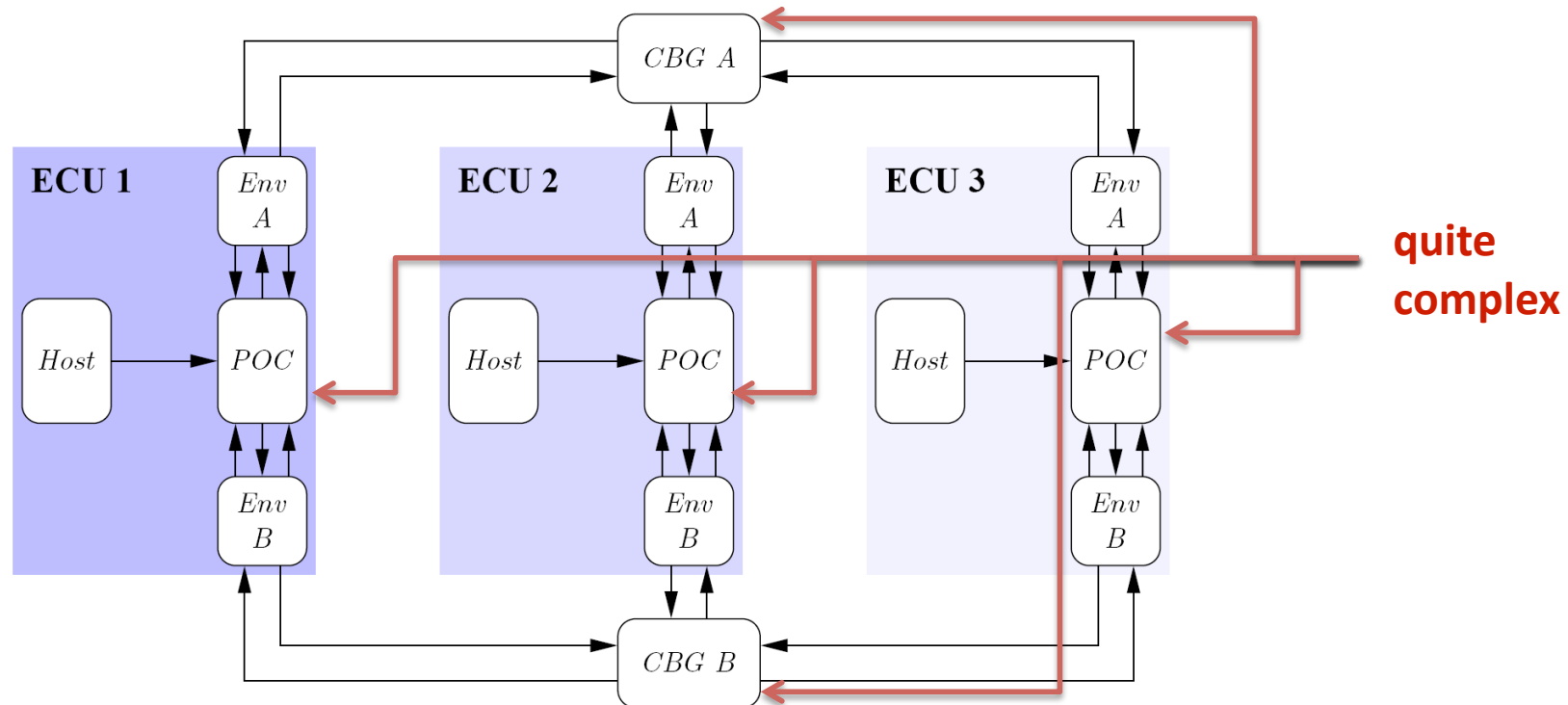
- Central Bus Guardian (CBG)



- FlexRay tolerates a CAS-babbler if host and CBG are configured properly (not specified)

Experimental Results: Startup

- Architecture similar to wakeup (only one more ECU)



- Optimizations and abstractions needed

Experimental Results: Startup

- Fault-free scenario

Coldstarters	Non-coldstarters	Channels	WCET	Memory	Time
2	1	A	15 cycles	3.989MB	0m3s
3	0	A	15 cycles	7.193MB	0m8s
2	1	A & B	15 cycles	1079.538MB	84m2s
3	0	A & B	-	1782.544MB	149m45s

anceled

- Redundant communication medium is hard to verify (non-determinism, state space explosion)

Experimental Results: Startup



- Fail-silent node (+ 2 correct nodes, 1 channel, 1 CBG): **WCET of 29 cycles**
- Fail-silent CBG (+ 1 correct CBG, 2 correct nodes): **WCET of 15 cycles**
- Resetting (leading) coldstarter (+ 2 correct nodes, 1 channel, 1 CBG): **WCET of 41 cycles**
- CAS-babbler (+ 2 correct nodes, 1 CBG, 1 channel): **unbounded WCET**

Conclusion

- Schematic translation from SDL to Promela is a good way to start
 - However, extensive manual refinement is needed in this case study
- CAS-babbling not always tolerated (not even with a CBG)
- Proper host and cluster configuration essential for timeliness properties
 - A host that switches off the bus driver can enhance fault tolerance
- **Models well suited to validate configurations**

Thank you.