Automotive and highly dependable Networks

H. Kopetz, TU Wien (see references in the introduction)

Excellent surveys:

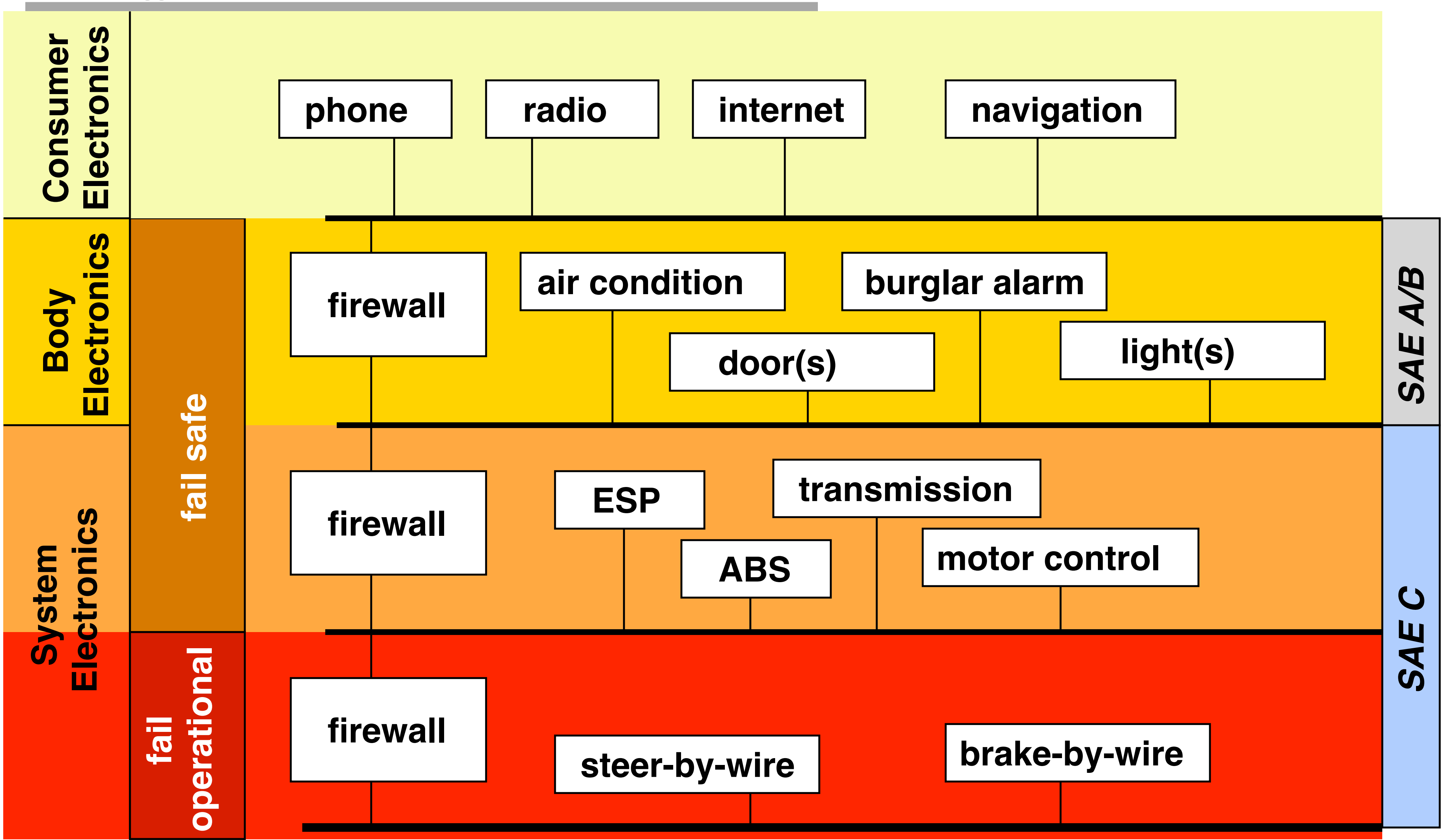
TTP:
Hermann Kopetz, Günther Bauer:
"The Time-Triggered Architecture"
http://www.tttech.com/technology/docs/history/HK_2002-10-TTA.pdf

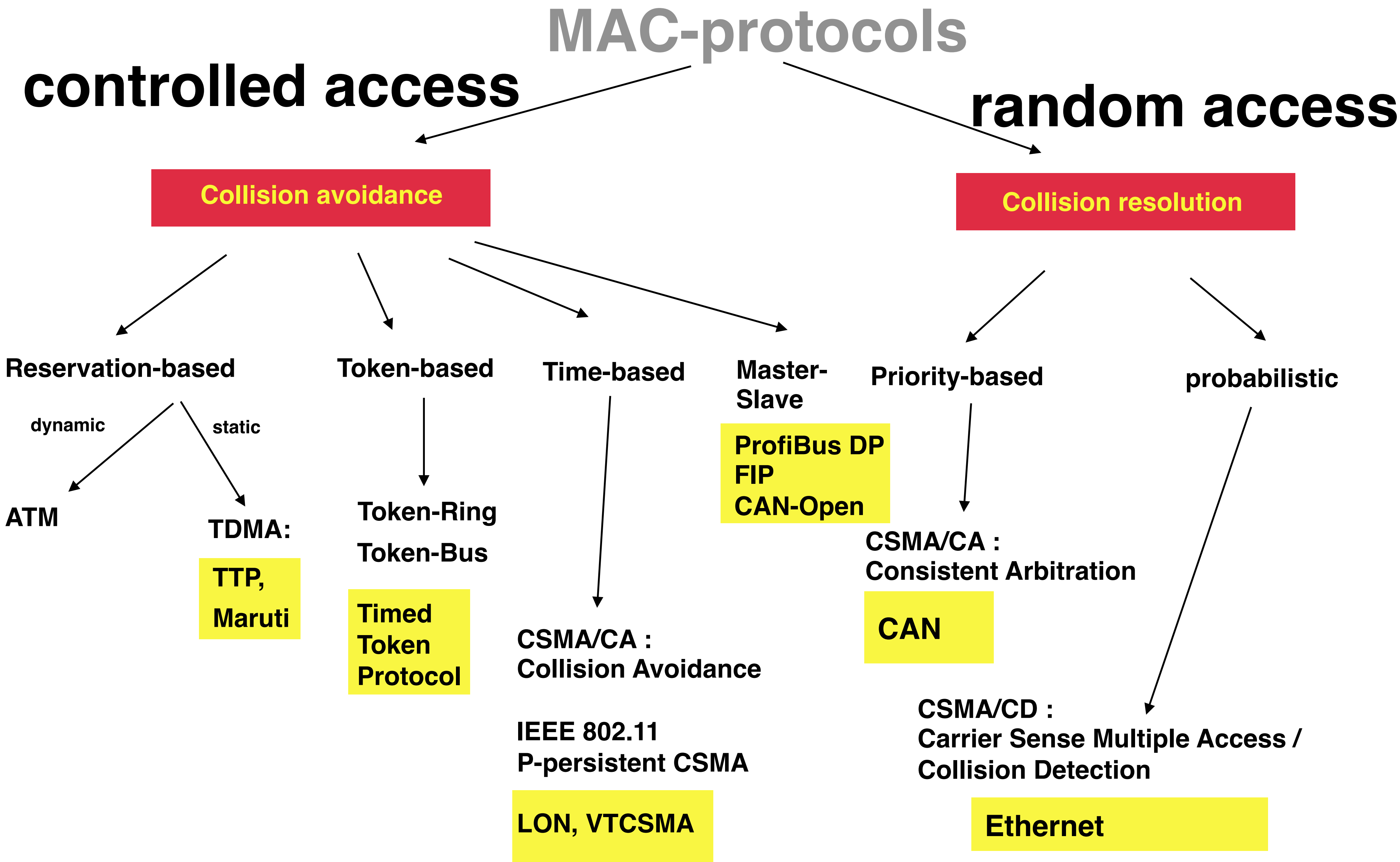
Networks for safety critical applications in general:
John Rushby:
"Bus Architectures for Safety-Critical Embedded Systems"
http://www.csl.sri.com/users/rushby/papers/emsoft01.pdf

Products:
http://www.tttech.com/

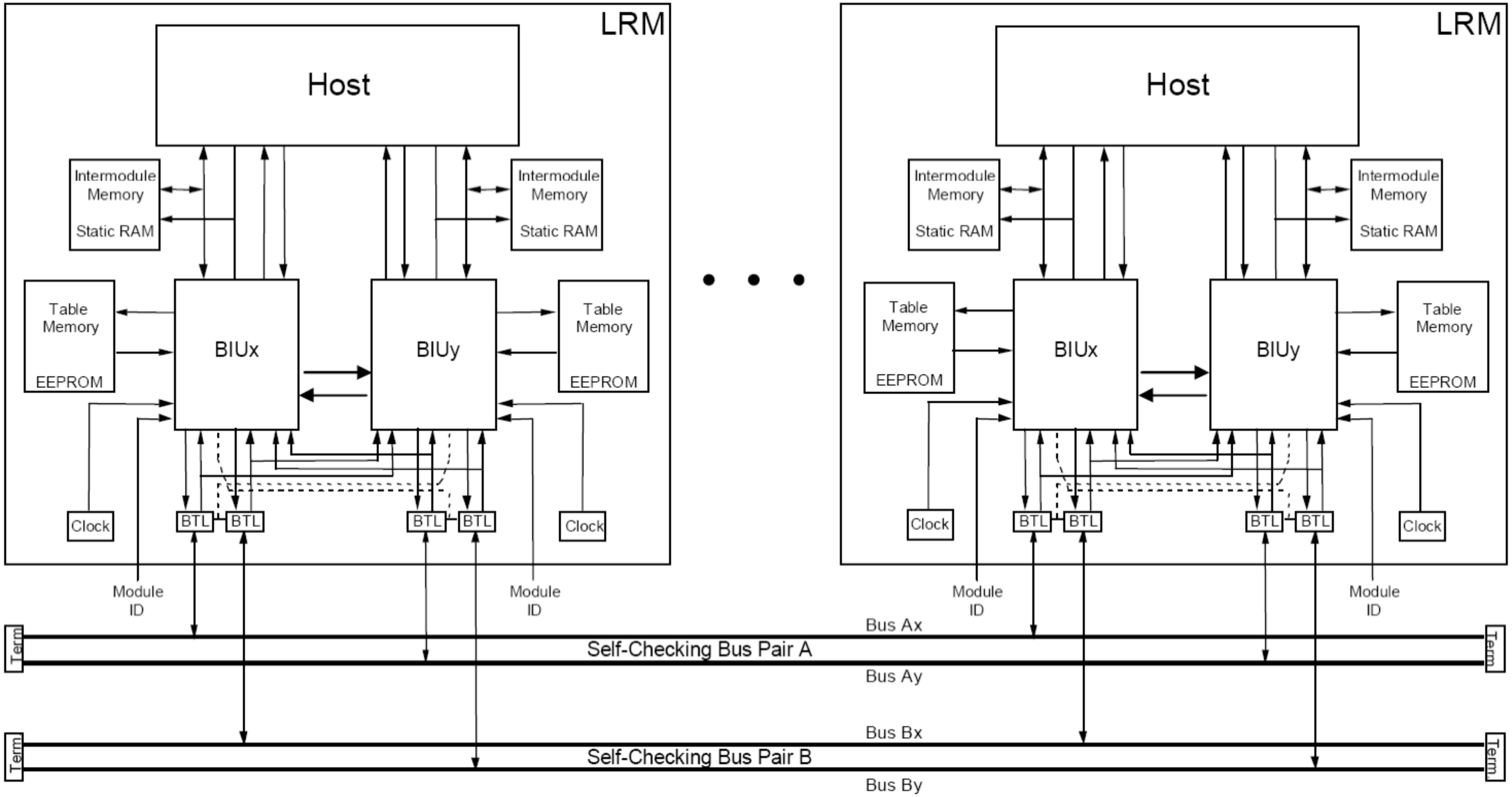
Communication levels in a car

(T. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, M. Walther: „Time Triggered Communication on CAN“)




MAC-protocols
controlled access
random access
Collision avoidance
Collision resolution
Reservation-based
dynamic
static
ATM
TDMA:
TTP,
Maruti
Token-based
Token-Ring
Token-Bus
Timed
Token
Protocol
Time-based
CSMA/CA :
Collision Avoidance
IEEE 802.11
P-persistent CSMA
LON, VTCSMA
Master-
Slave
ProfiBus DP
FIP
CAN-Open
Priority-based
CSMA/CA :
Consistent Arbitration
CAN
probabilistic
CSMA/CD :
Carrier Sense Multiple Access /
Collision Detection
Ethernet

Hardware-Structure of the SAFEbus



Brendan Hall, Kevin Driscoll, Michael Paulitsch, Samar Dajani-Brown, "Ringing out Fault Tolerance. A New Ring Network for Superior Low-Cost Dependability," dsn, pp. 298-307, 2005 International Conference on Dependable Systems and Networks (DSN'05), 2005

Automotive and highly dependable Networks

TTP/C
Byteflight
FlexRay
Braided Ring

Time Triggered CAN (TTCAN)
TTP/A
LIN

Time Triggred Protocol (TTP)

Objectives:

- **Predictable, guaranteed message delay**
- **No single fault should lead to a total network failure**
- **Fault-Tolerance**
  - **Fault detection on the sender and the receiver side**
  - **Forward error recocery**
  - **Treating massive temporary faults (Black-out)**
  - **Distributed redundancy management**

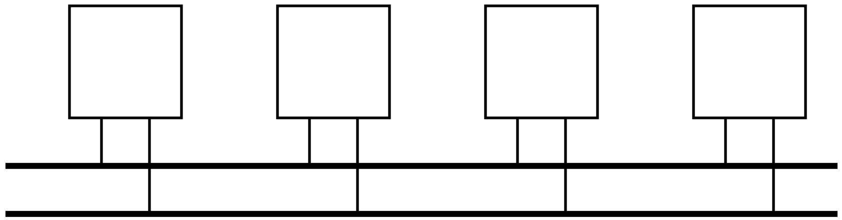
- **Clock synchronization**
- **Membership-service (basis for atomic multicast)**
- **Support for fast consistent mode changes**
- **Minimal protocol overhead**
- **Flexibility without sacrifycing predictability**

Design principles

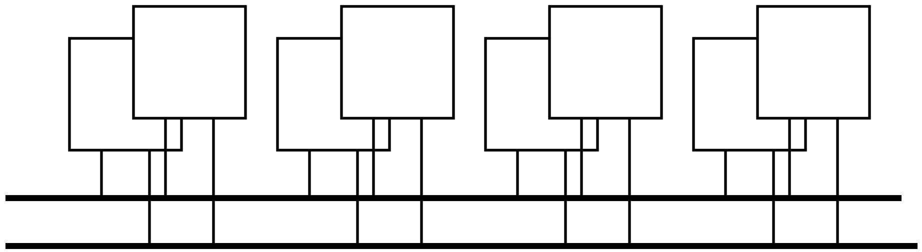
- Exploiting a priori knowledge (static message schedule)
- Implicit flow control
- Fail silence
- Continuous supervision and consistent view of system state

Fault-Tolerant Network Configurations

Class 1:
1 node/FTU
2 frames/FTU

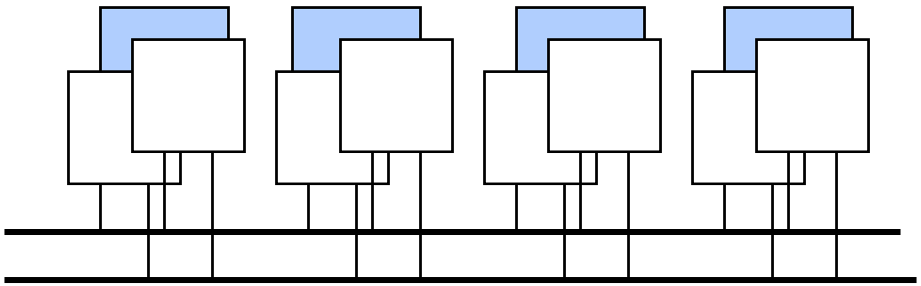


Class 2:
2 active node/FTU
2 frames/FTU



Class 3:
2 active nodes/FTU
4 frames/FTU

Class 4:
2 active nodes/FTU
+ 1 spare/FTU
4 frames/FTU



component redundancy + time redundancy

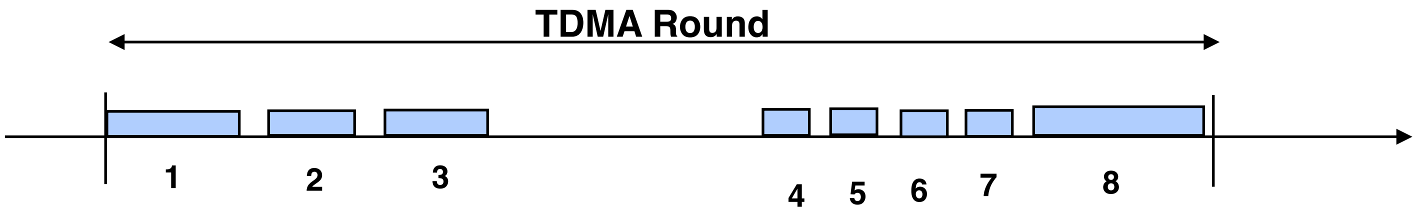
Fault-tolerance parameters

failure type	failure probability/h
permanent node failure	10^{-6}/h
permanent channel failure	10^{-5}/h
transient node failure	10^{-4}/h
transient channel failure	10^{-3}/h

what is the relation: faulty messages / overall number of messages ?

type of failures	Class 1	Class 2	Class 3	Class 4
Perm. node failure	0	1	1	2
Perm. comm. failure	1	1	1	1
Trans. node failure	0	1/Rec.interv.	1/Rec. interv.	1/TDMA-round
Trans. comm. failure	1 of 2	1 of 2	3 of 4	3 of 4

Exploit a priori knowledge: Off-line Scheduling



	time	address	Attributes			
			D	L	I	A
1						
2						
3						
4						
5						
6						
7						
8						

Message Description List

MEDL

time: defines the point in time when the message has to be transmitted

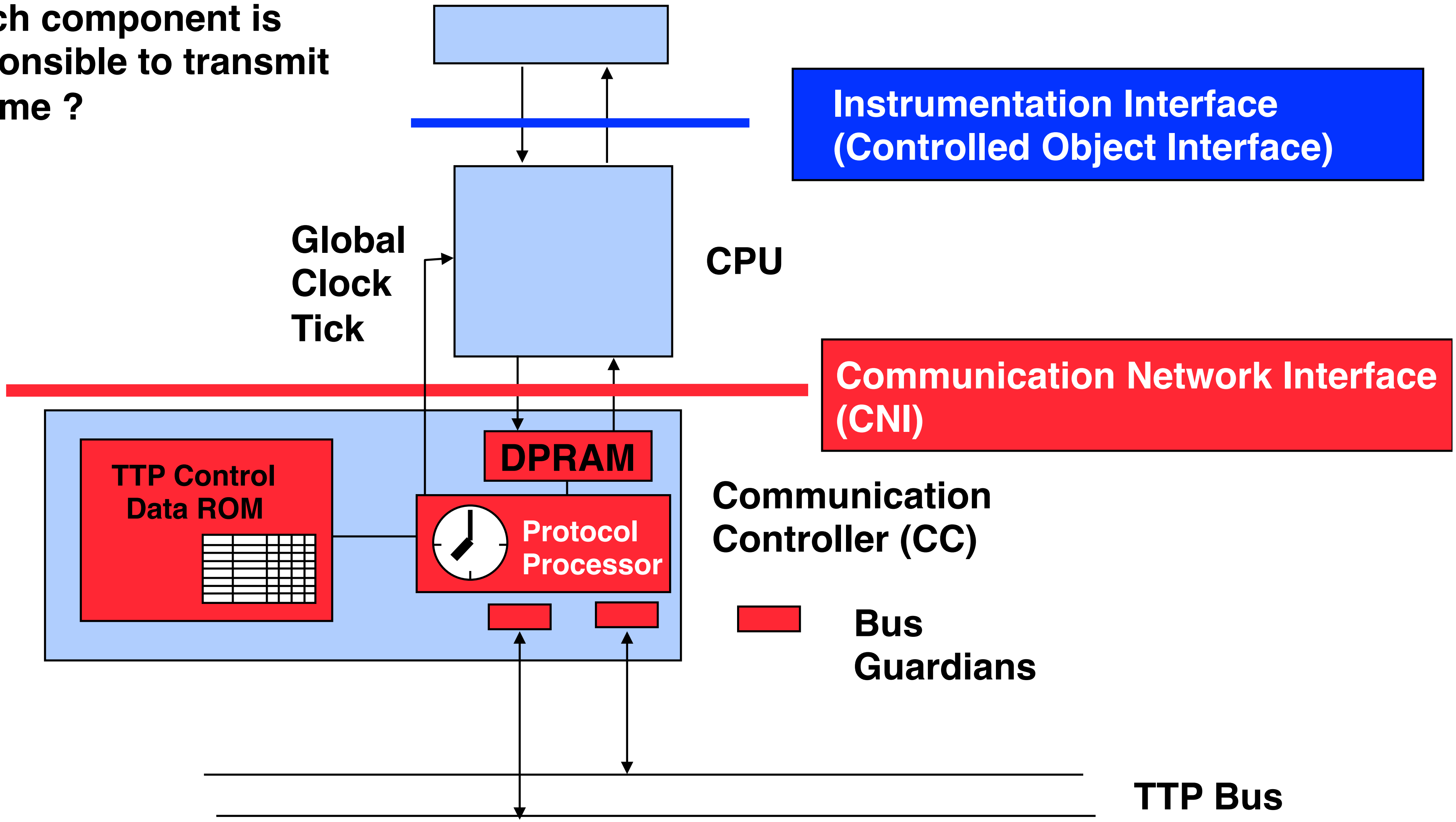
Address: Defines the local address where the messages to be transmitted/ received are stored in the node's memory

D: Direction Input or Output
L: frame length
I: Init or normal message
A: "Additional" Parameter Field

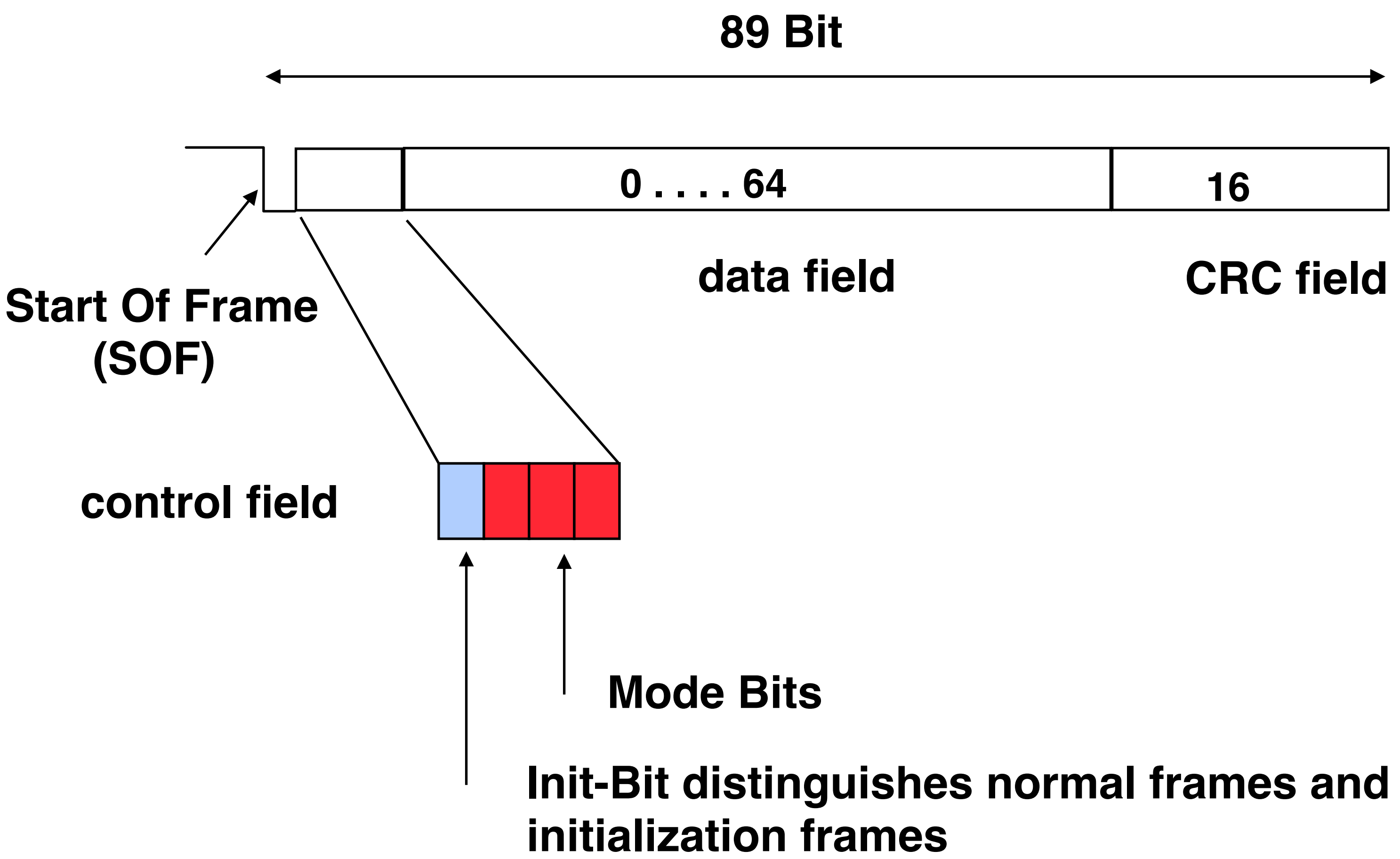
TDMA Round (Cluster Cycle): Every FTU has at least transmitted once in a round.

Fail silence und strict enforcement of transmit times

Which component is responsible to transmit a frame ?

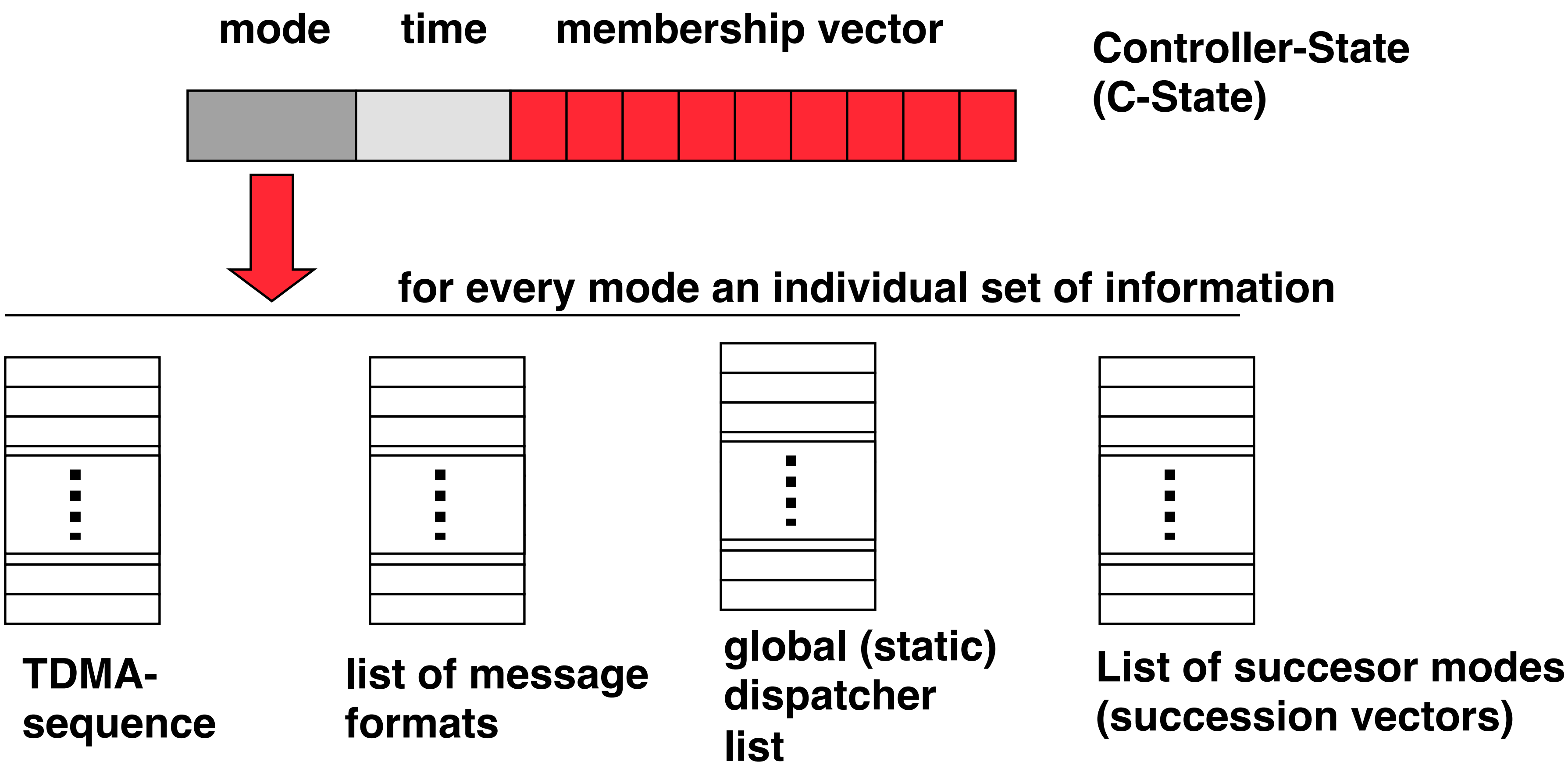


Format of a TTP frame

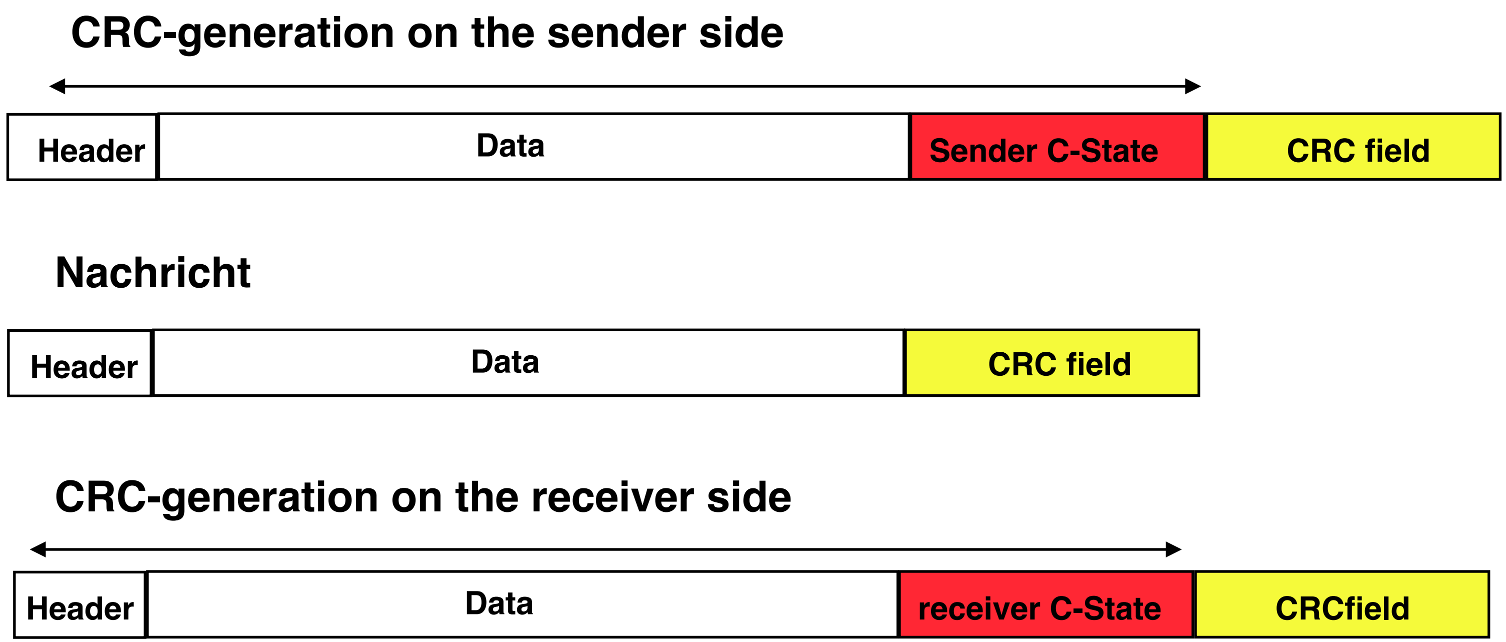


MFM Coding: Constant frame length (not data dependent)

Continuous supervision of the global state



Continuous supervision of the global state



Handling mode changes

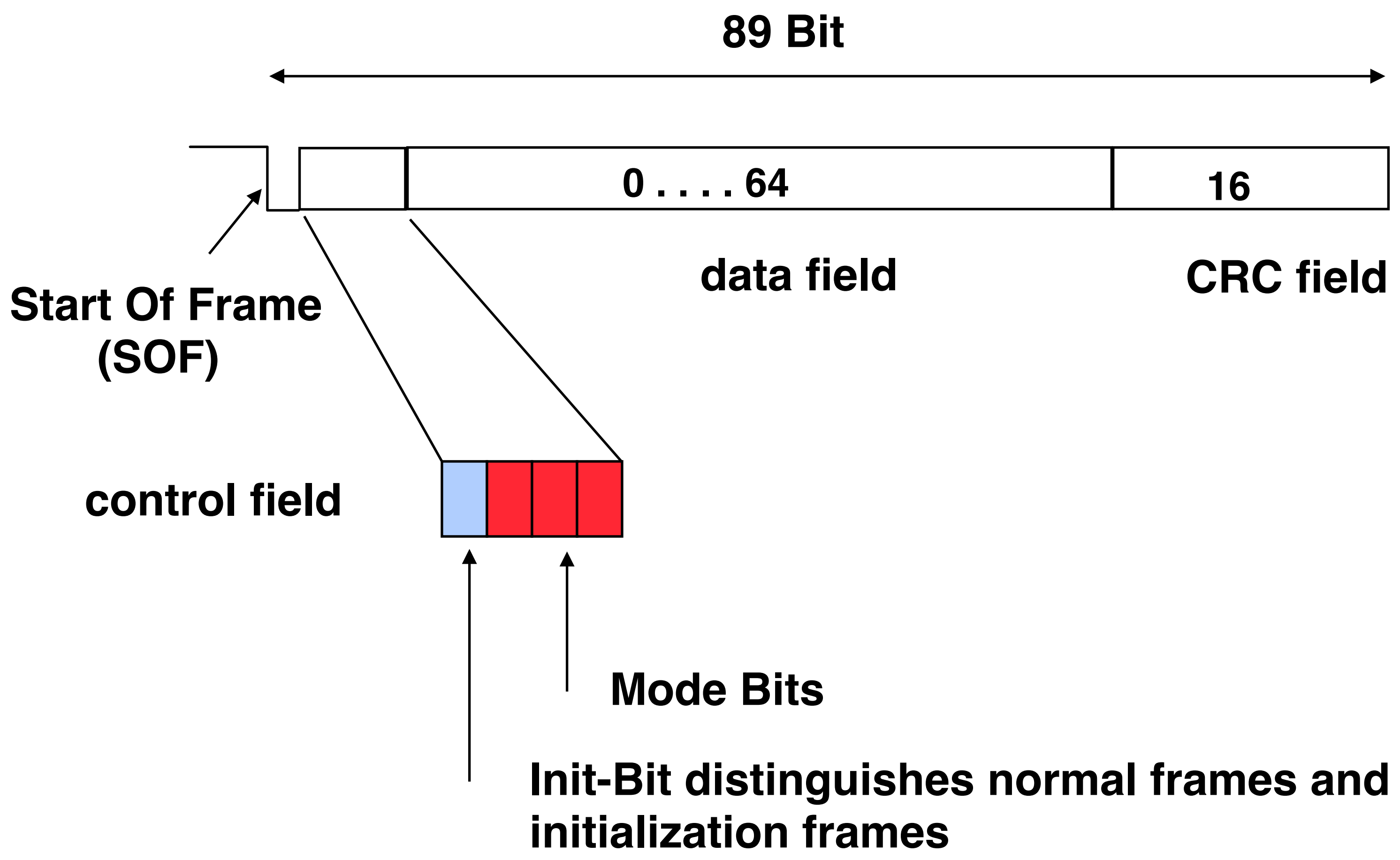
At every point in time, all nodes are in a specific mode.

→ needs consensus

Mode changes:
FTU signals mode changes in the control field by setting the position of the succession vector (index into the respective table).

→ Flexibility: Succession vector can be changed.

Format of a TTP frame



MFM Coding: Constant frame length (not data dependent)

Critical functions:

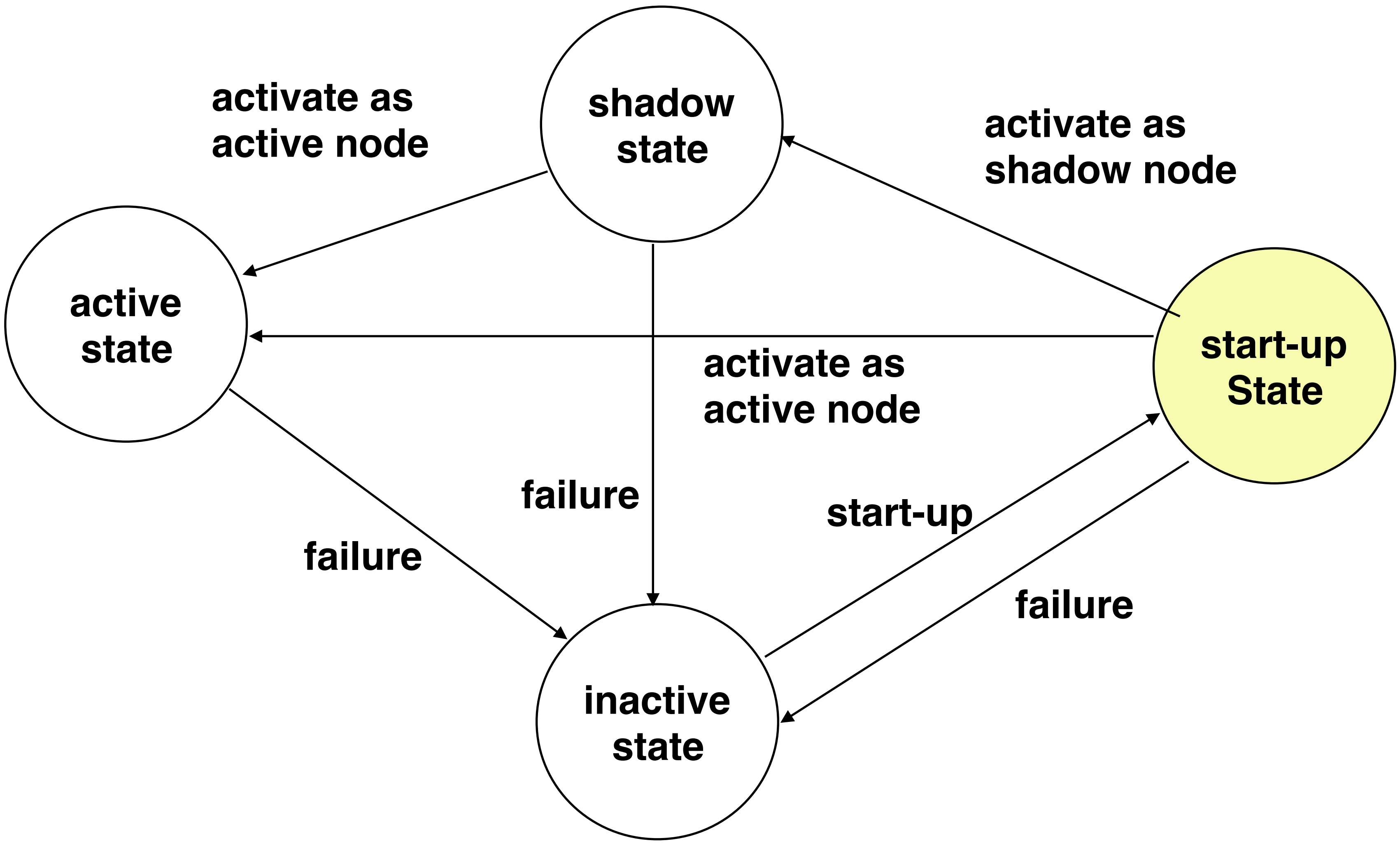
- Initialization
- Membership
- Black-out Handling

Redundancy management and initialization

- **Every node has a unique name that defines its position in the TDMA round.**
- **Some special nodes are enabled to send initialization frames (I-frame).**
- **Initialization frames comprise the complete state of the entire system.**
- **The longest interval between two I-frames determines the minimal waiting time for a node before it can be re-integrated.**

Redundancy management and initialization

Local states of an FTU:



Redundancy management and initialization

- Reset local clock.

- Monitoring the bus for I_1 (I_1 > longest TDMA round)
  An I-frame will be sent during this time if the network is initialized.

  in case of message traffic, wait for an I-frame

in case of NO message traffic, wait specified time I_2
(I_2 is a node specific delay to ommit collisions)

After I_2 send I-frame with C-state in the init-mode

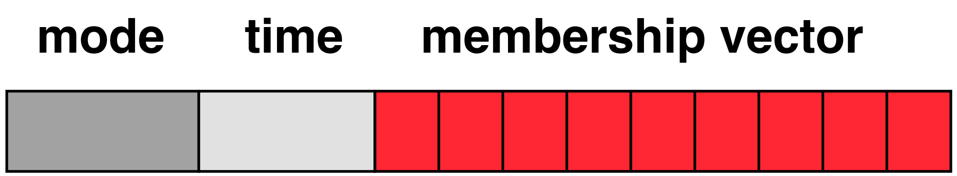
Membership Service

Sender sets membership bit (MB) to "1"

All receivers set MB to "1"

If no correct frame is received, all receivers set MB = 0 directly after the TDMA-slot

When reaching the membership-point (an a priori known point in time, when the FTU sends a message), the sender checks whether it still is member in the group.



Controller-State (C-State)

Membership Service

A node is member if:

1. the internal check is ok.

2. at least one frame which has been sent during the round has been acknowledged from one of the FTUs, i.e. the physical connection is ok.

3. the number of correct frames which were accepted by the FTU during the last TDMA round is bigger than the number of discarded frames.

If this is not the case, then the local C-state is not in compliance with the majority of other nodes and the node looses its membership. This avoids the formation of cliques, which have different views on the whole group.

Black-out handling

"Black-out" denotes a global distortion, e.g. if the physical communication channel is distorted by external electromagnetic fields.

Black-out detection:

A node continuously monitors the membership field.
If membership dramatically decreases a mode change is triggered to black-out handling.

↓

Black-out mode: nodes only send I-Frames and monitor the bus

↓

When external distortion vanishes, membership will stabilize again.

↓

Return to "normal mode"

Discussion TTP

Synchrony (Jitter, Steadyness, Thightness)

Automatic clock synchronization

Fault masking

Monopolization- (Babbling Idiot-) faults are omitted

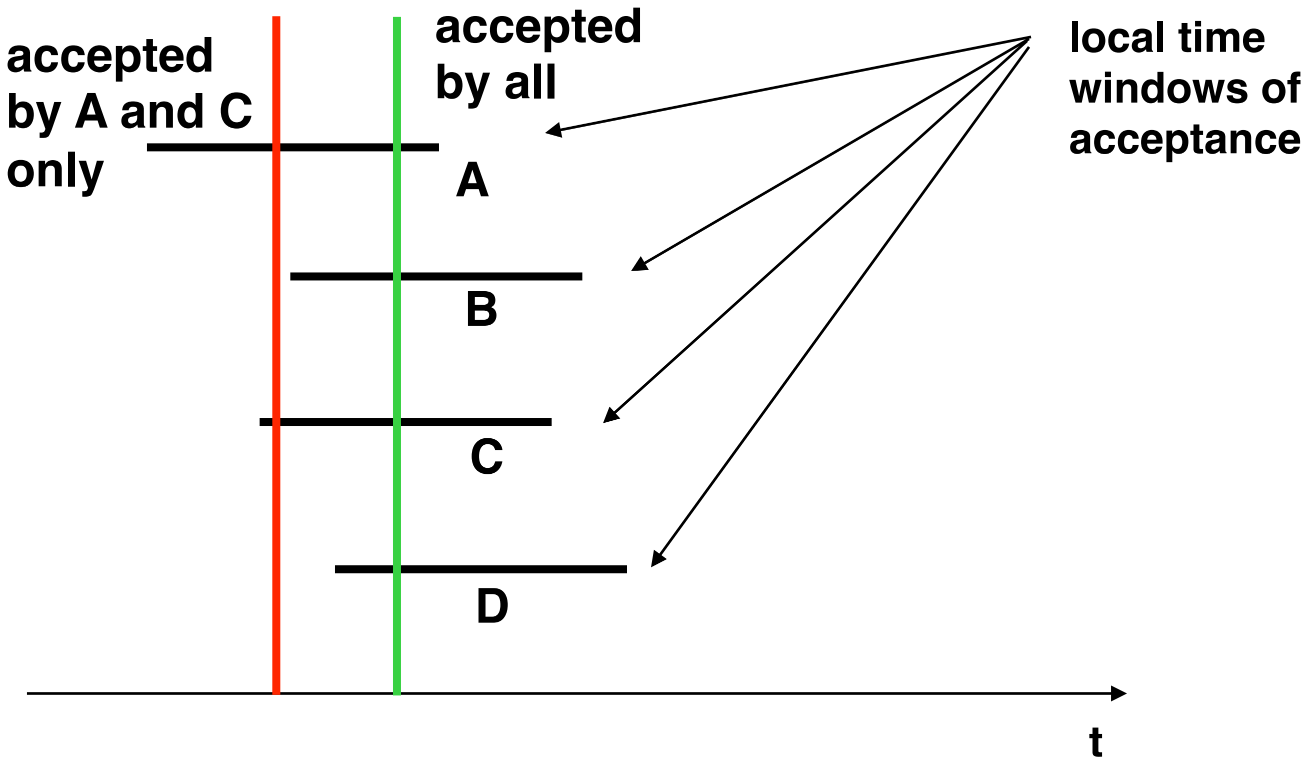
Replica Determinism

Composability and extensibility

Problems with a Bus Topology

- Monopolization failures
- Common mode & spatial proximity failures
- Synchronization between nodes and SOS failures

Slightly-off-specification failures

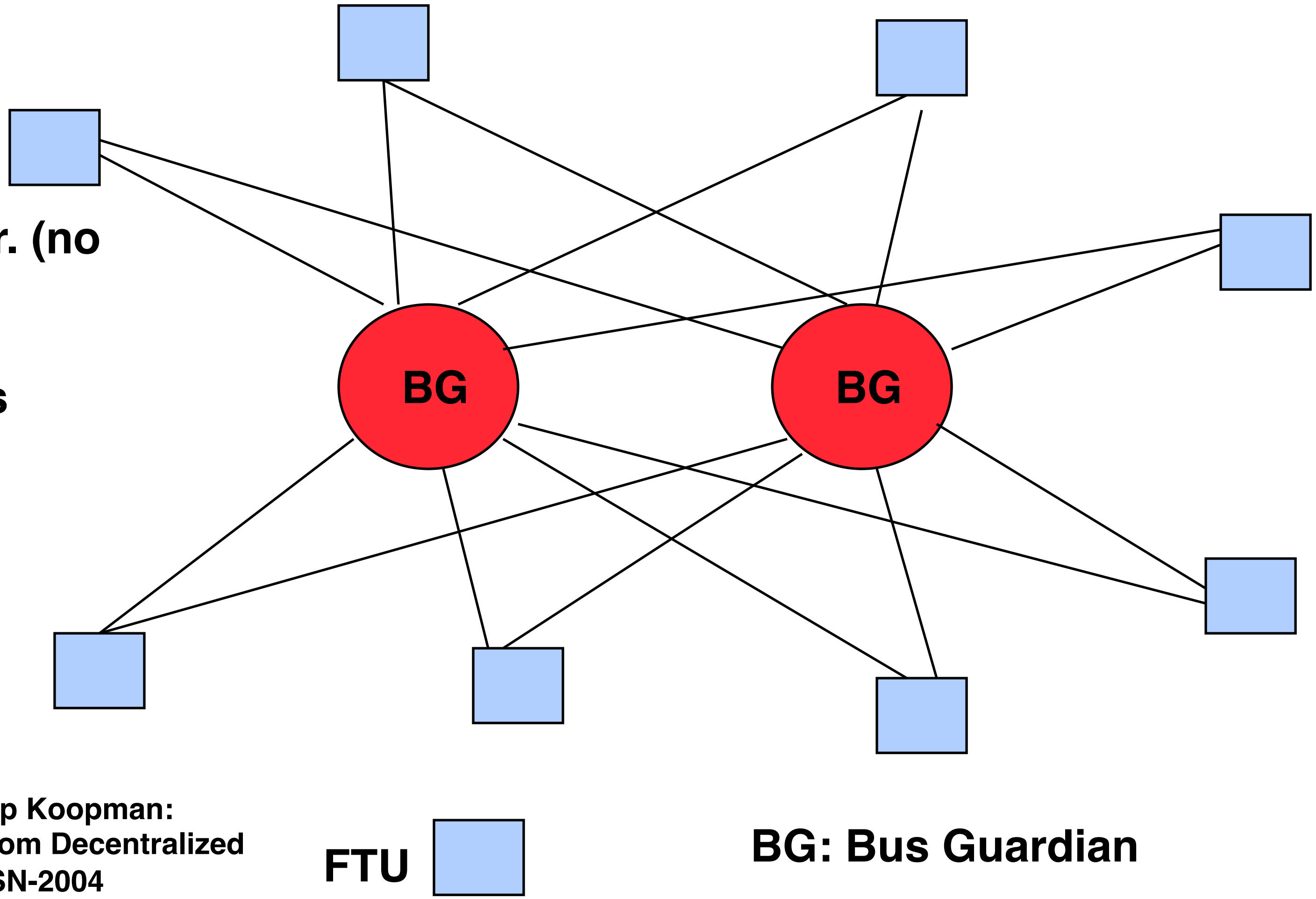


Slightly-off-specification failures can occur at the interface between the analog and the digital world. They may occur in the value (signal level) and in the time domain.

Migration of Bus-Guardians: Star-Topology

Motivation:

- **Re-shaping and synchr. (no SOS failures)**
- **Isolation of faulty FTUs**
- **Physical separation of Bus guardians from hosts (less common mode failures)**



BUT: needs careful design!
Jennifer Morris, Daniel Kroening, Philip Koopman: Fault Tolerance Tradeoffs in Moving from Decentralized to Centralized Embedded Systems, DSN-2004

Summary TTP

- Protocol execution is initiated by the progression of global time.
  The sending point in time for every message is a priori know by all receivers.
- The maximum execution time corresponds to the average execution time (with a small deviation only)
- Error detection is possible for the recievers because they know when a message can be expected.
- The protocol is unidirectional. No acknowledgements are required.
- Implicit flow control is needed.
- No arbitration conflicts can occur.

Desirable Features

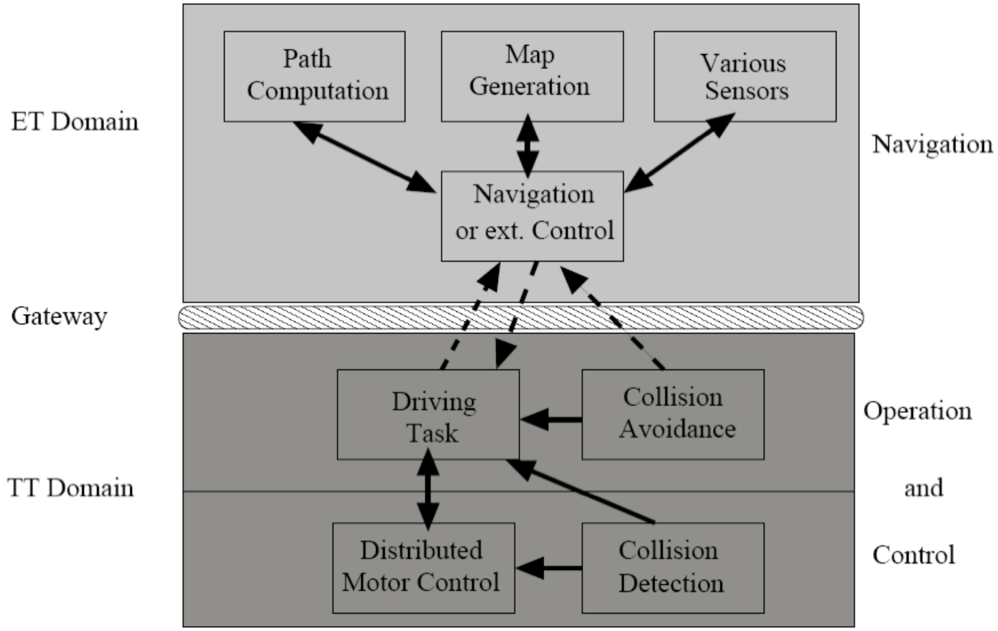
More Flexibility:

- **Accomodating a range of criticality requirements**
- **Accomodating more messages than slots**
- **Dynamic assigment of transmission slots**
- **Event-triggered message dissemination**

What will be the price to pay?

More Flexibility ?

Federating networks with different properties.



Needs a careful estimation of latencies!

Integrated Architectures

Basic Idea: Virtualization of networks

- Run multiple networks with different properties on a time-triggered network.
- Underlying TDMA provides temporal separation.
- Needs considerable bandwidth

$$\text{Total bandwidth} \geq \sum_{i=1}^{i=\text{max \# of networks}} \text{bandwidth of network } i$$

Integrated Architectures

"A further difference is that in a physical CAN network, the total bandwidth is potentially available to every node, but must be shared between them. The VN, on the other hand, assigns TDMA slots with their corresponding bandwidth exclusively to the jobs. In order to provide to each individual job the same maximum bandwidth in the prototype implementation, the entire VN is allocated the ten-fold bandwidth (1170 kbps or 4920 kbps) in case of the 10 jobs in the VN configuration. Consequently, the 117 kbps or 492 kbps are simultaneously available to all jobs. This allocation is possible, because the underlying physical Ethernet network provides a higher raw bandwidth (i.e., 100 Mbps) compared to the physical CAN networks."

R. Obermaisser: Temporal Partitioning of Communication Resources in an Integrated Architecture, Transactions on Dependable and Secure Computing byteflight –

A New High-Performance Data Bus System for Safety-Related Applications

By Josef Berwanger, Martin Peller and Robert Griessbach

BMW AG, EE-211 Development Safety Systems Electronics, Knorrstrasse 147, 80788 Munich, Germany

http://www.byteflight.com/presentations/atz_sonderausgabe.pdf

Flexible protocol supports synchronous and asynchronous messages

supports high data rates

availability of integrated communications-controller (e.g. Motorola 68HC912BD32)

integral part of FlexRay

Principles:

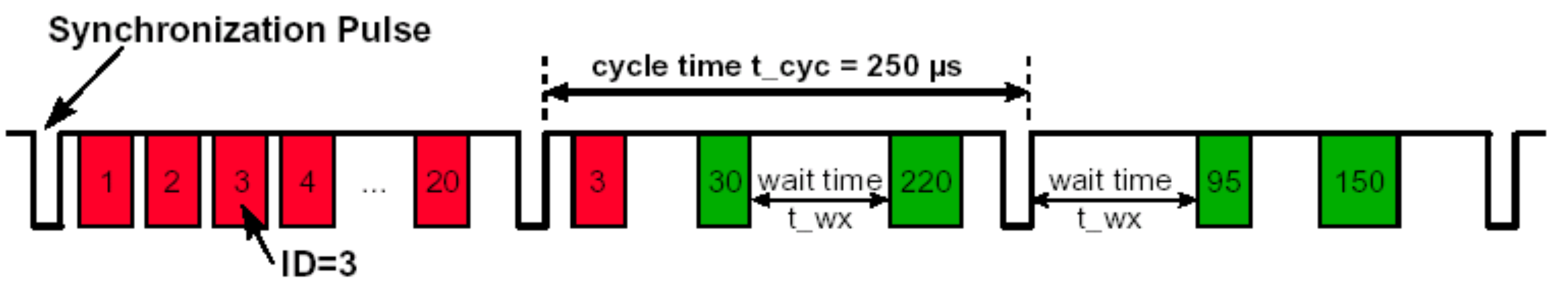
- message priorities are associated with node-IDs
- time slots, which correspond to certain priorities
- priority is enforced by waiting times

Assumptions

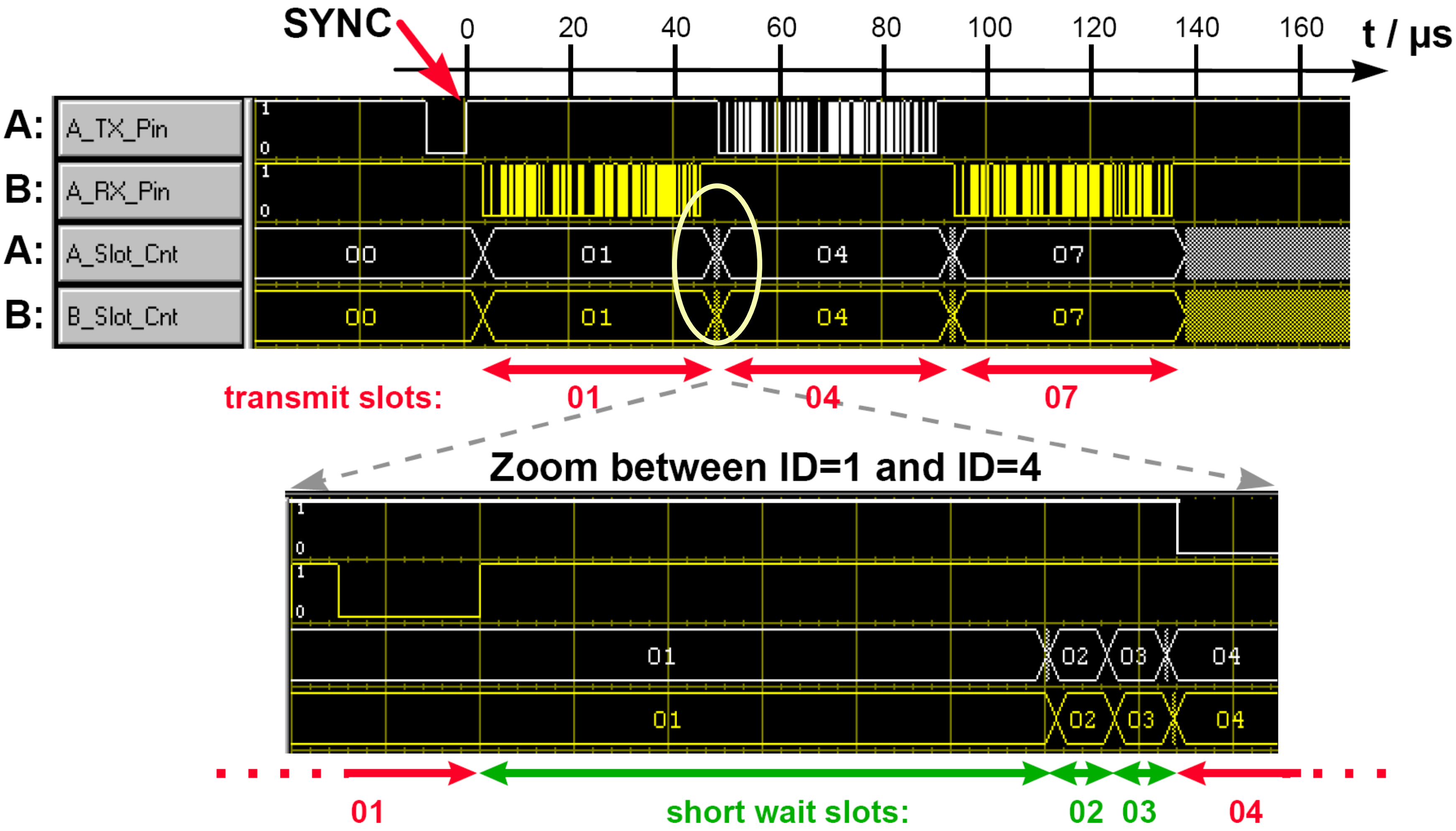
- **Communication is organized in rounds or cycles respectively.**
- **Clock synchronization between nodes is assumed to be better than 100ns.**
- **One (fault-tolerant) sync master responsible to indicate the start of a round by sending a sync pulse.**
- **The interval between two sync pulses determines the cycle time (250 μs @ 10 Mbps)**

Byteflight: Flexible TDMA

- **SyncMaster sends the synchronization pulse to init the cycle.**
- **The interval between two sync pulses determines the cycle time (250 μs @ 10 Mbps)**
- **Every node has a number of identifiers assigned that define message priorities. The system must ensure that the message IDs are unique.**
- **Every communication controller has a counter which counts message slots.**
- **The counter is stopped on an ongoing message transfer and will be started again when the transfer has completed.**
- **If the counter value corresponds to the priority of a message, this message can be transmitted.**

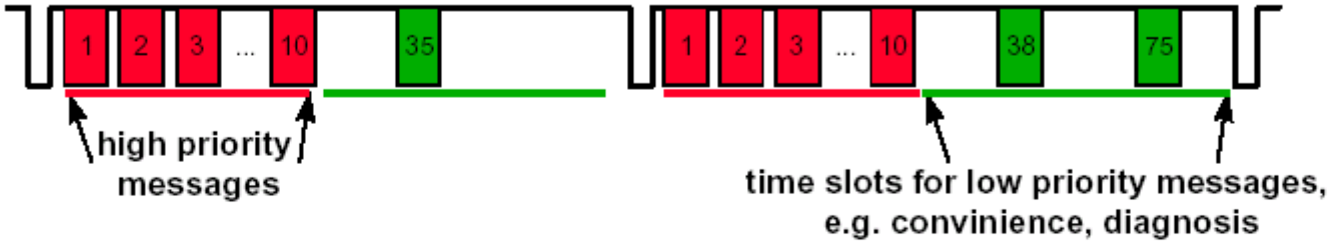


Distributed synchronized "Slot-" counter



Waiting period *t_wait = t_0 + t_delta * (ID - ID_{t-1})*

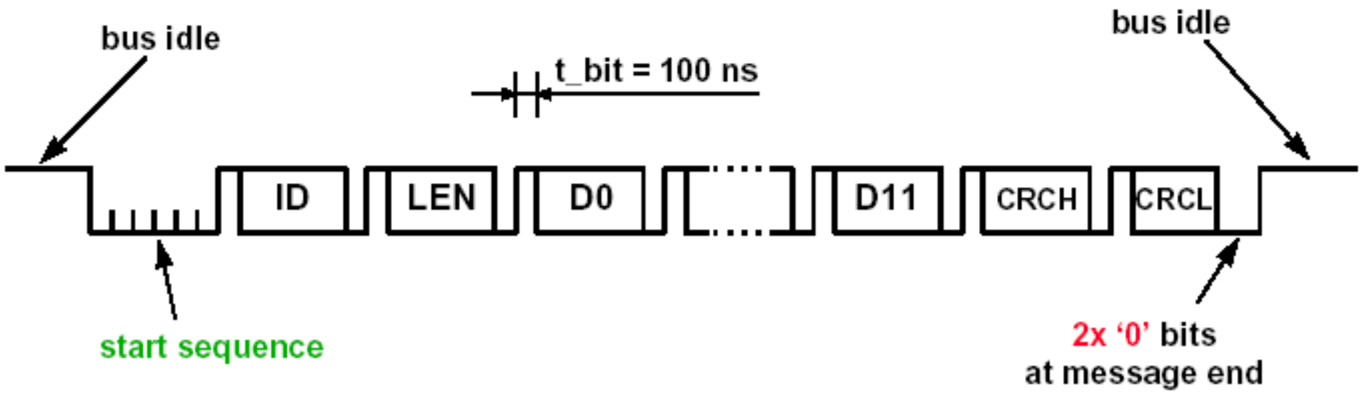
Synchronous and asynchronous data transmission



Slots with fixed priorities are reserved for synchronous messages. These slots are assigned in every cycle (1-10) and allow a deterministic analysis of message latencies.

Asynchronous messages have lower priorities. These are dynamically assigned and enforced by the waiting mechanism. To determine message latencies, only probabilistic analysis is possible.

ByteFlight message format



Start sequnence:	**6 Bits**
ID:	**8 Bits (1 Byte)**
Length:	**8 Bits (1 Byte)**
Data:	**96 Bits (12 Bytes)**
CRC:	**16 Bits (Hamming distance = 6)**

Fault handling in the Byteflight Protocol

Alarm state:

The master can send a special synchronization signal that is recognized by all stations. This signal has no influence on the protocol but the nodes can detect a specific situation locally.

Fault treatment:

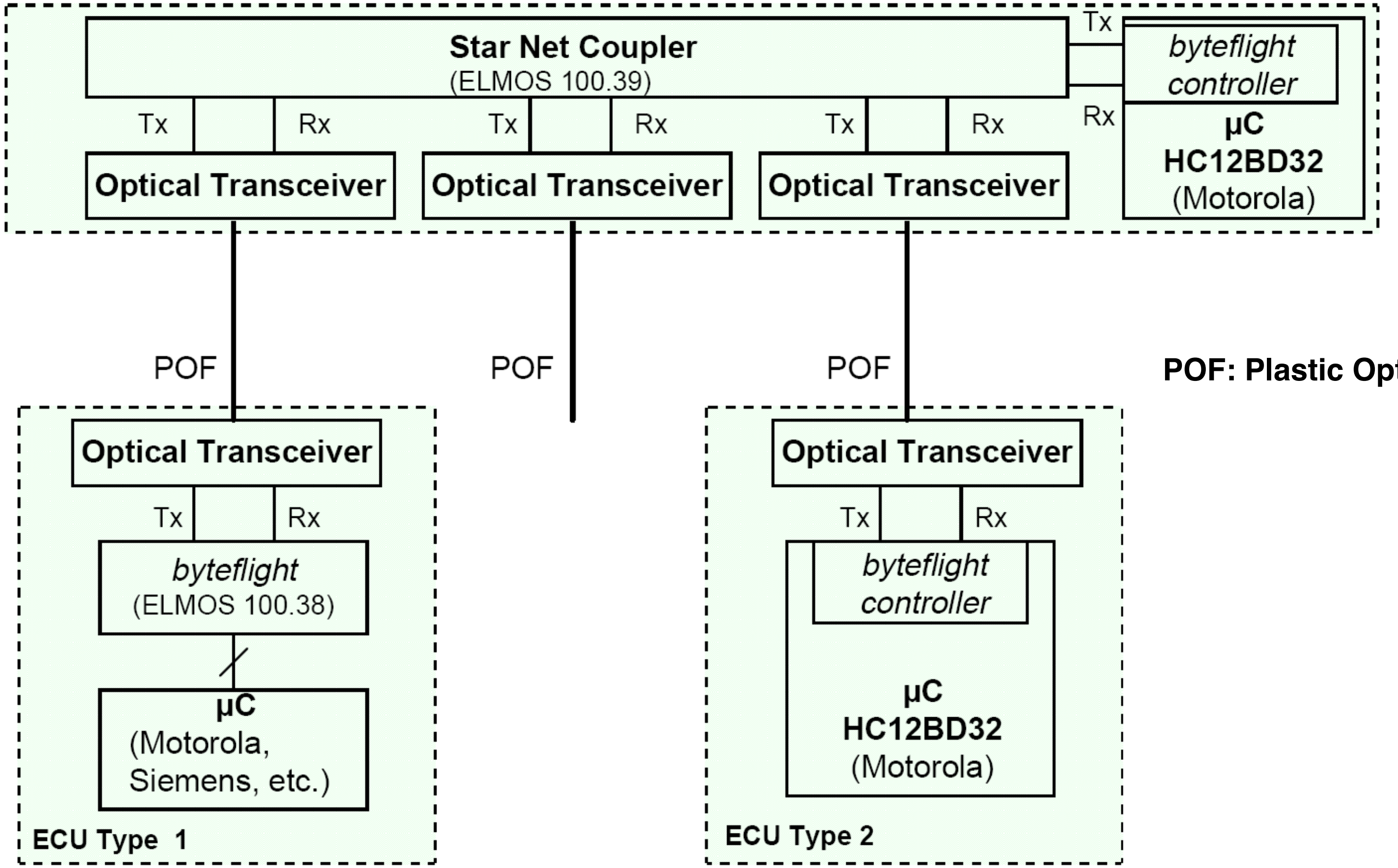
Transient transmission faults are not specially treated and no re-transmission is initiated. It is assumed that with the next cyclic transmission this fault is gone.

Timing errors are handled by the star coupler.

In a bus structured network, bus guardians are used to enforce a fail silent behaviour. Here the protocol exploits the strict timing discipline.

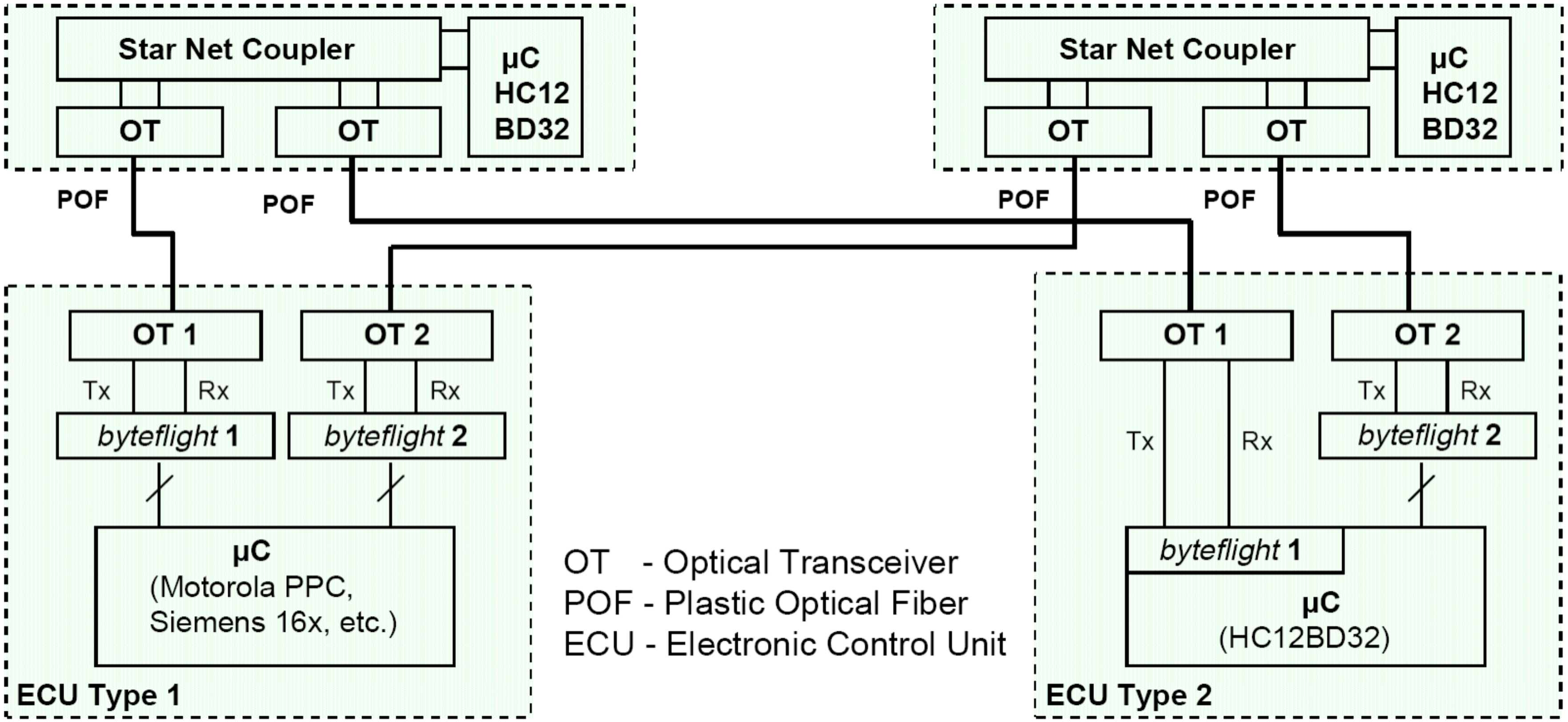
Replacements for a failing sync master are determined a priori.

Example of a Byteflight topology



POF: Plastic Optical Fiber

Byteflight star topology & redundancy concept



OT - Optical Transceiver
POF - Plastic Optical Fiber
ECU - Electronic Control Unit

Comparison between Byteflight and TTP

byteflight: a new high-performce data bus system for safety related applications,
J. Berwanger, M. Peller, J. Griessbach, BMW-AG, EE211 Development Safety Systems Electronic

Feature	CAN	TTP [10]	*byteflight*
Message transmission	asynchronous	synchronous	asynchronous and synchronous
Message identification	message identifier	time slot	message identifier
Data rate	1 Mbps gross	2 Mbps gross	10 Mbps gross
Bit encoding	NRZ with bit stuffing	modified frequency modulation (MFM)	NRZ with start/stop bits
Physical layer	transceivers up to 1 Mbps	not defined	optical transceiver up to 10 Mbps
Latency jitter	bus load dependent	constant for all messages	constant for high priority messages according t_cyc
Clock synchronization	not provided	distributed, in µs range	by master, in 100 ns range
Temporal composability	not supported	supported	supported for high priority messages
Error containment (physical layer)	partially provided	provided with special physical transceiver	provided by optical fiber and transceiver chip
Babbling idiot avoidance	not provided	possible by independent bus guardian	provided via star coupler
Extensibility	excellent	only if extension planned in original design	extension possible for high priority messages with affect on asynchronous bandwidth
Flexibility	flexible bandwidth for each node	only one message per node and TDMA cycle	flexible bandwidth for each node
Availability of components	several µC families and transceiver chips	microcoded RISC chip available, physical transceiver and independent bus guardian not available	HC12BD32, E100.38 *byteflight* standalone controller, E100.39 star coupler ASIC, optical transceiver available

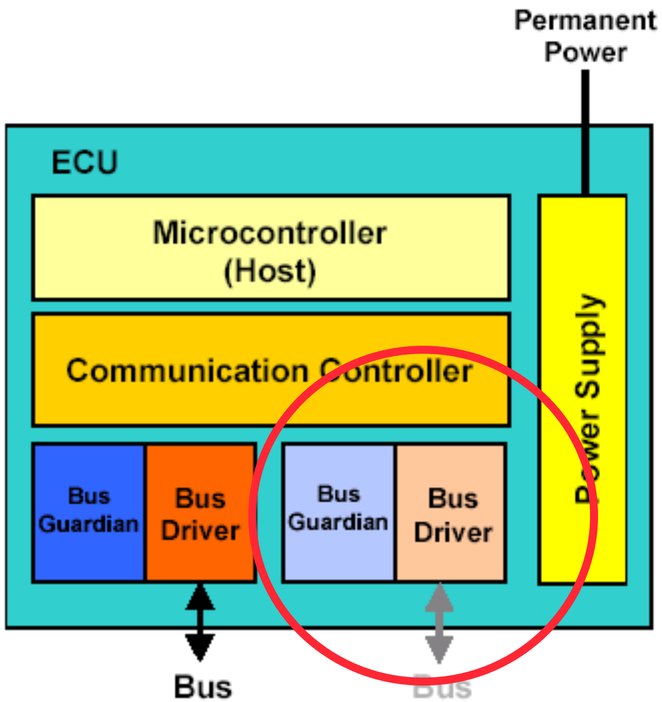
Combination of TDMA and Byteflight

Belschner et al. : Anforderungen an ein zukünftiges Bussystem für fehlertolerante Anwendungen aus Sicht Kfz-Hersteller

Requirements of the Protocol

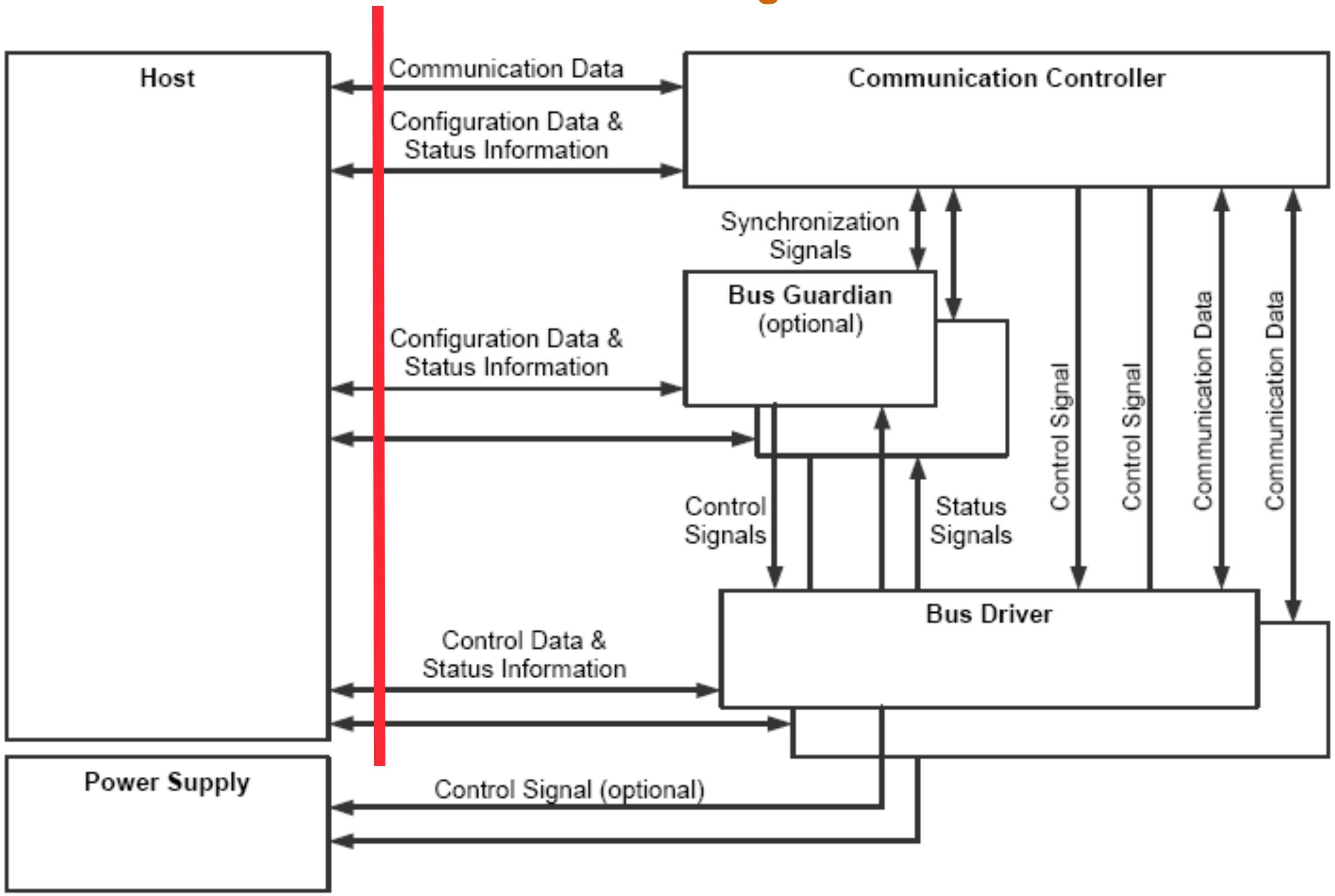
- Synchronous and asynchronous data transmission (scalable)
- Deterministic data transmission, guaranteed message latency
- Fault-tolerant, synchronized global time
- Redundant transmission channels (configurable)
- Flexibility (expandability, bandwidth usage, ...)
- Different topologies (bus, star and multi-star)
- Electrical and optical physical layer
- Communication protocol independent of the baud rate

Architecture of a FlexRay node (ECU: Electronic Control Unit)



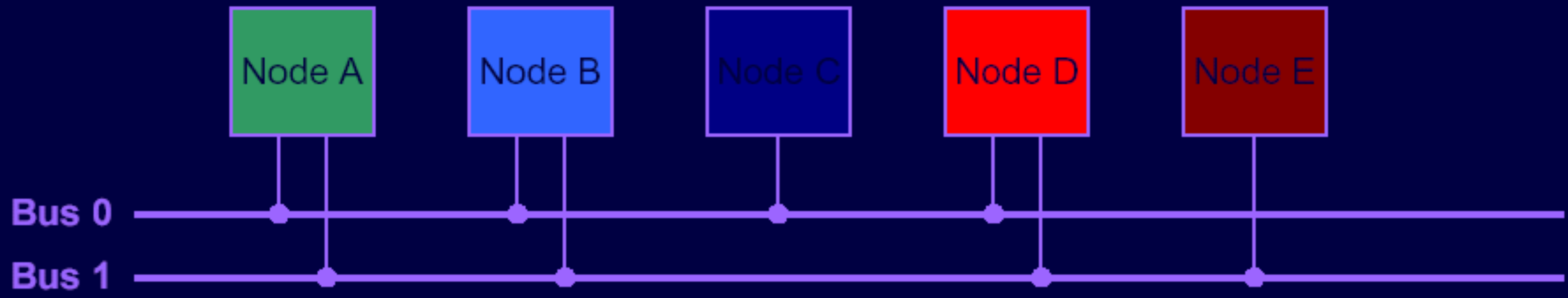
Interfacing the communication controller

CNI: no control signals



Data- und control flow between Host and CC

FlexRay Basic Concepts



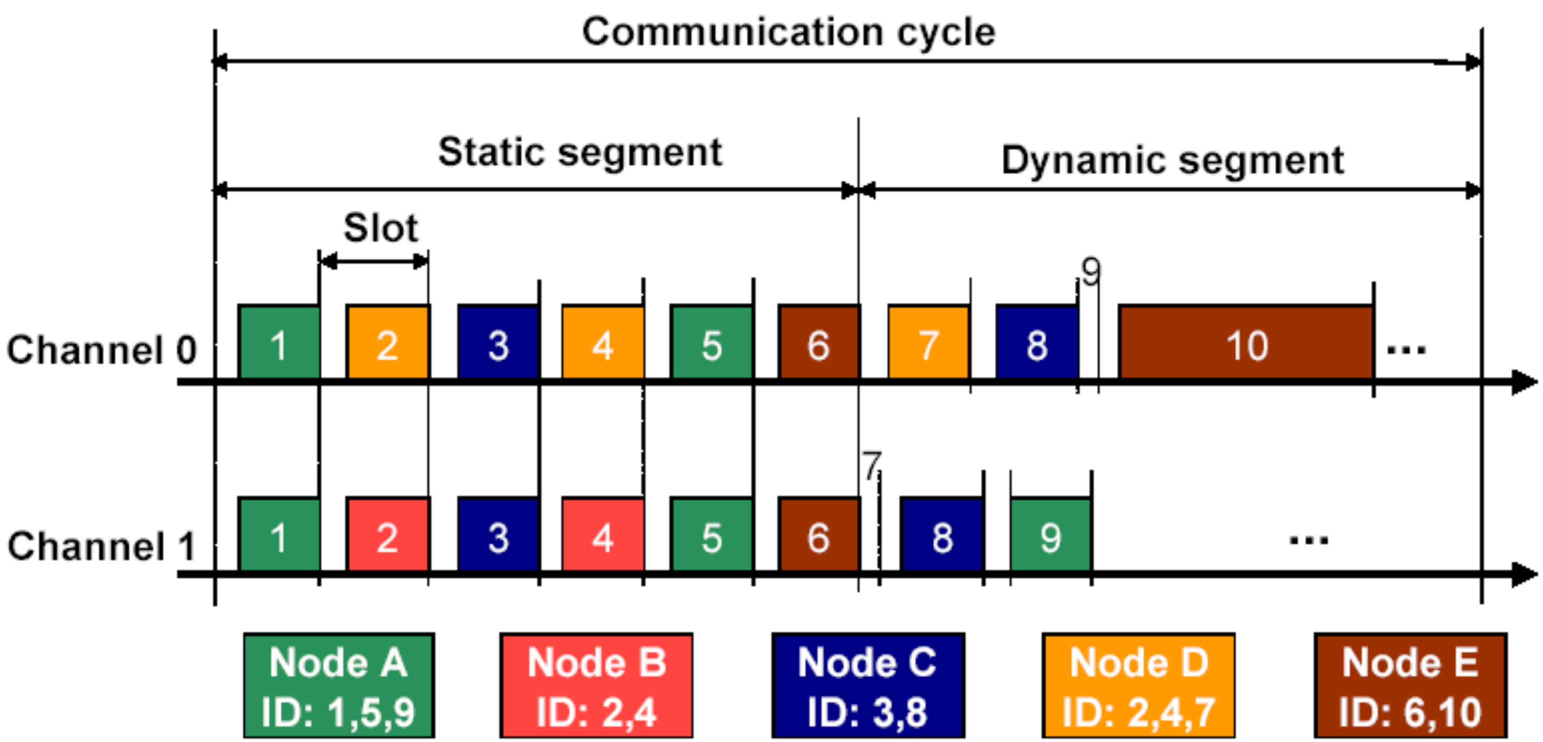
Redundancy

- The protocol supports two serial busses
- A node can either be connected to both or only one of the busses

PHY Bit Coding

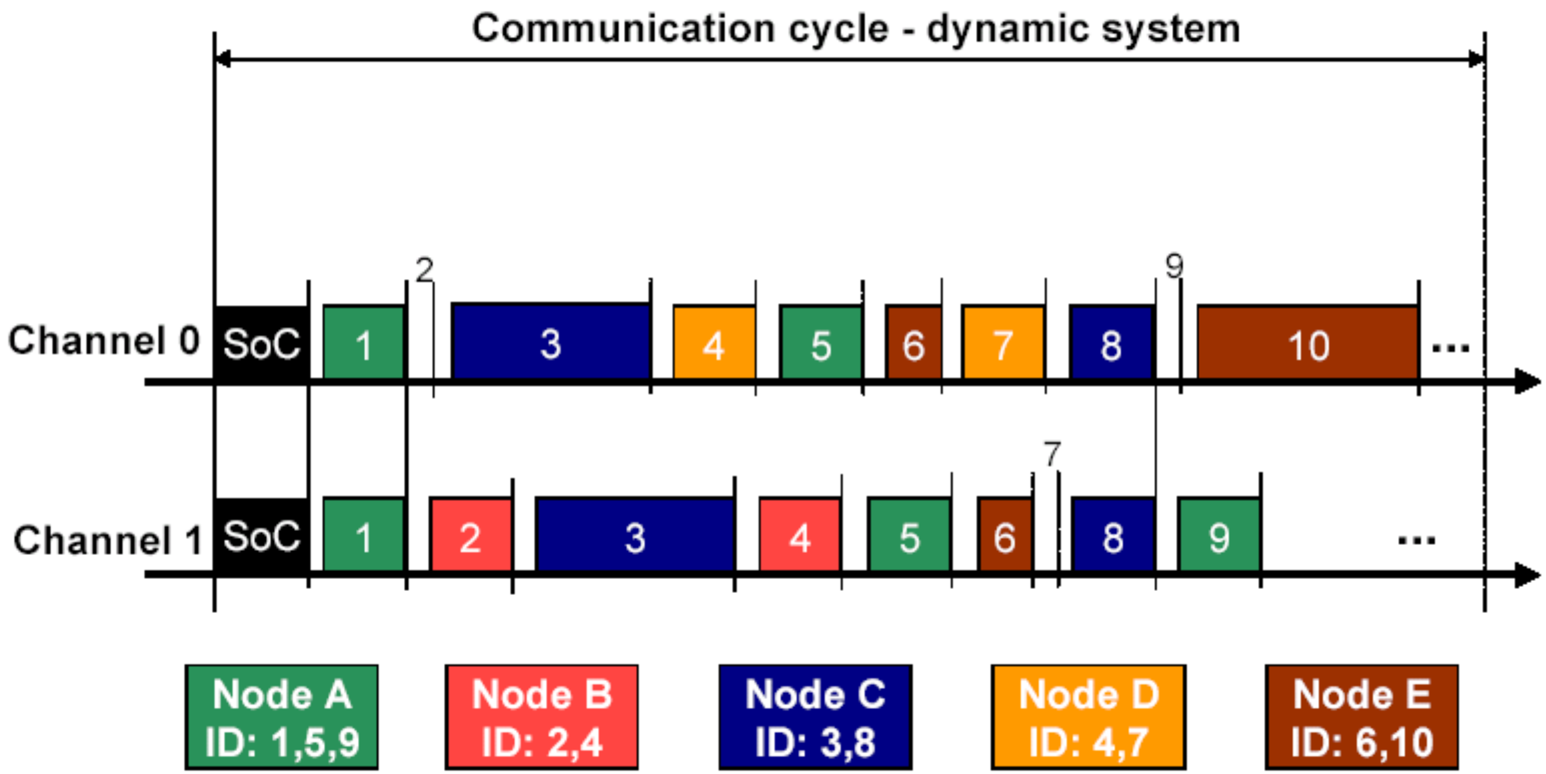
- transmission speed up to 10 Mbit/s (gross, optical)
- NRZ 8N1 for optical transmission
- Xerxes (MFM extension) coding for electrical transmission

The FlexRay Communication Cycle



Cycle with static and dynamic segment

The FlexRay Communication Cycle



Cycle with dynamic segment only

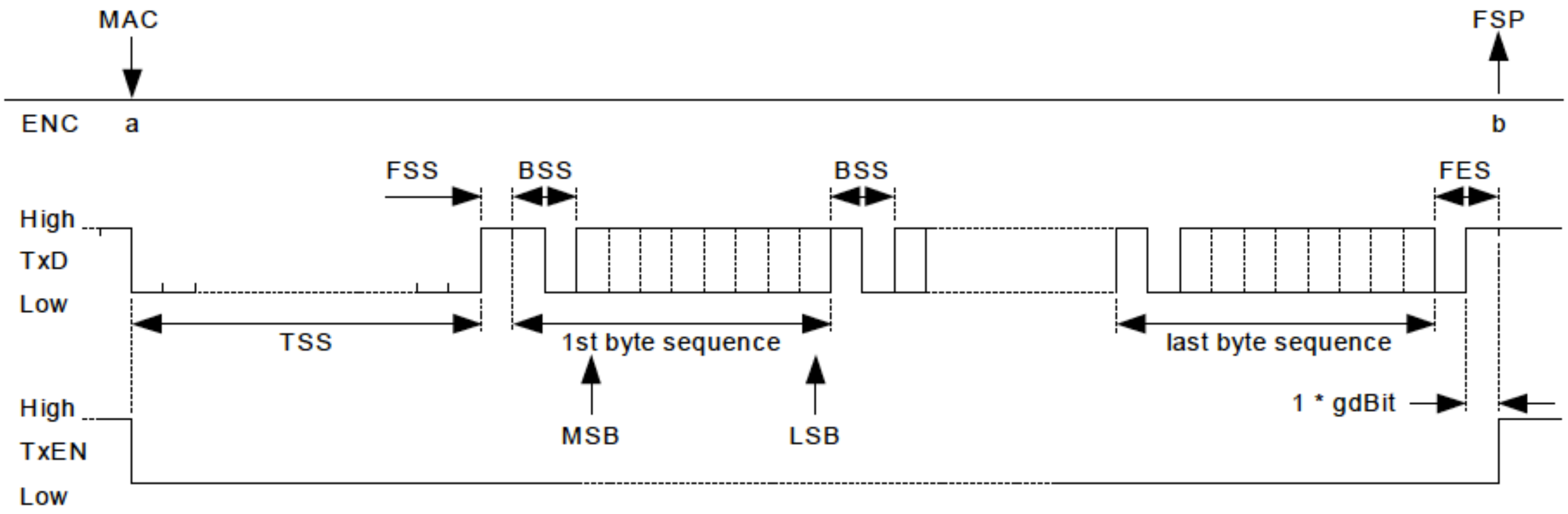


Figure 3-2: Frame encoding in the static segment.

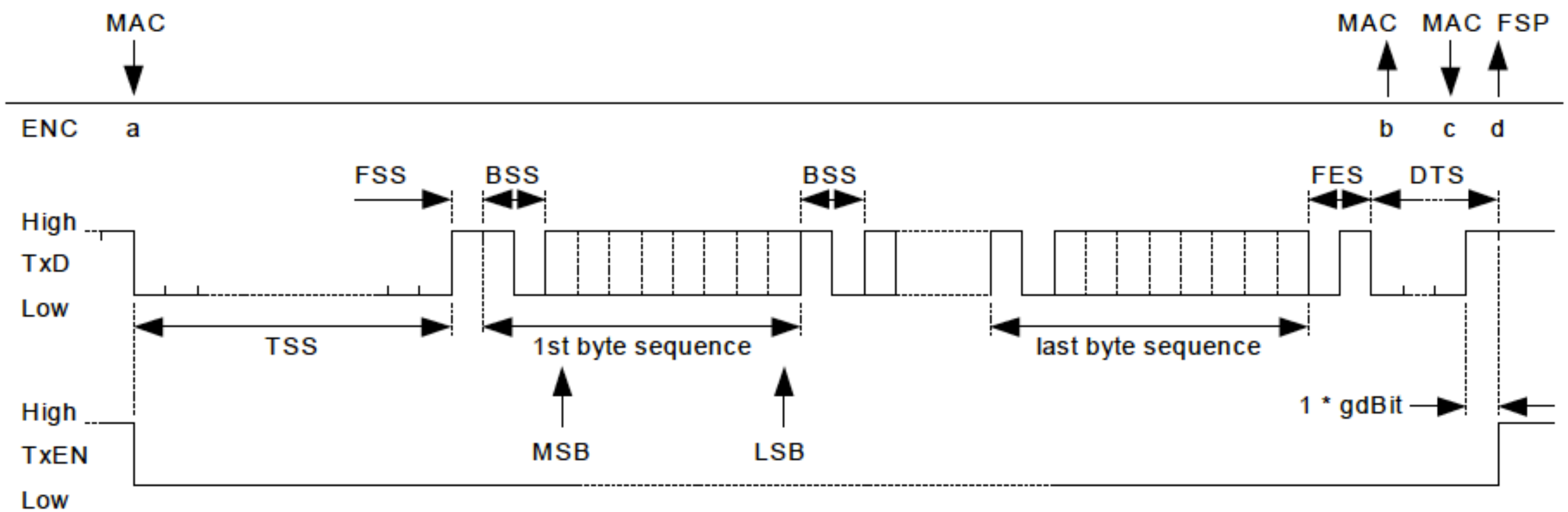


Figure 3-3: Frame encoding in the dynamic segment.

Format of a FlexRay frame

ID	MUX	SYNC	LEN	CYCLE	DATA	CRC	0
10 bit	1 bit	1 bit	4 bit	8 bit	0...11 byte	15 bit	1 bit

ID: Identifier, 10 Bit, value range: (1 ... 1023), defines the slot position in the static segment and the priority in the dynamic segment. A low ID defines a high priority. ID = 0 is reserved for the SYNC-symbol. An identifier must be unique in the network, i.e. two identical IDs would lead to a collision. Every node may use one or more identifiers in the static and the dynamic segment.

MUX: Multiplex-field, 1 Bit. This bit enables to send multiple data under the same ID..

SYNC: SYNC-field, 1 Bit. This bit indicates whether the message is used for clock synchronization and whether the first byte contains the sync counter (SYNC = “1”: message with Frame-Counter and clock synchronization, SYNC = “0”: message without counter)

LEN: Length field, 4 Bit, number of data bytes (0 ... 12). Any value > 12 will be interpreted as LEN=12. If the cycle counter (in the first byte) is used (SYNC=1) any value >11 is set to LEN=11.

CYCLE: The CYCLE-Field can be used to transmit the cycle counter or data. The cycle counter is synchronously incremented at the start of every communication cycle by all communication controllers.

D0-11: Data bytes, 0 – 12 bytes

CRC: 15 Bit Cyclic Redundancy Check.

Topology Options

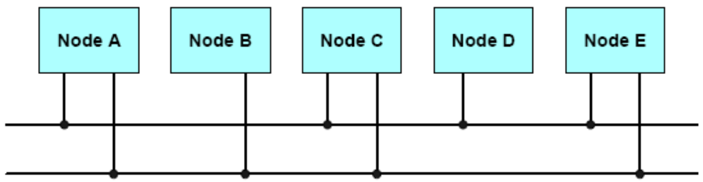


Figure 1-1: Dual channel bus configuration.

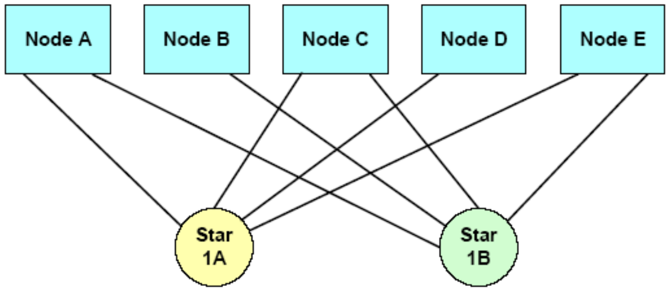


Figure 1-2: Dual channel single star configuration.

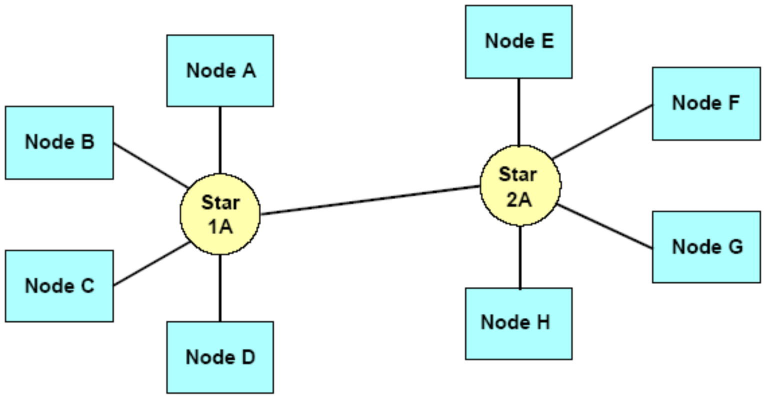


Figure 1-3: Single channel cascaded star configuration.

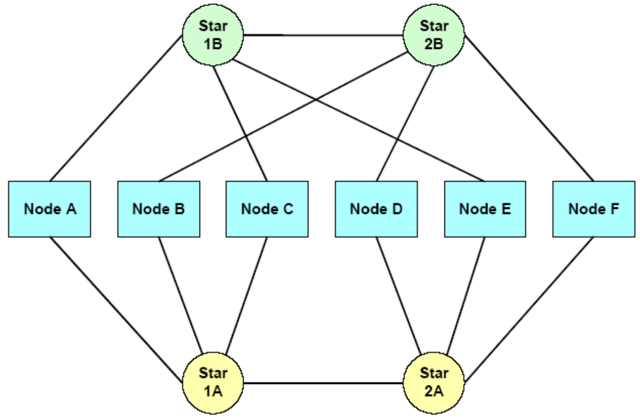


Figure 1-4: Dual channel cascaded star configuration.

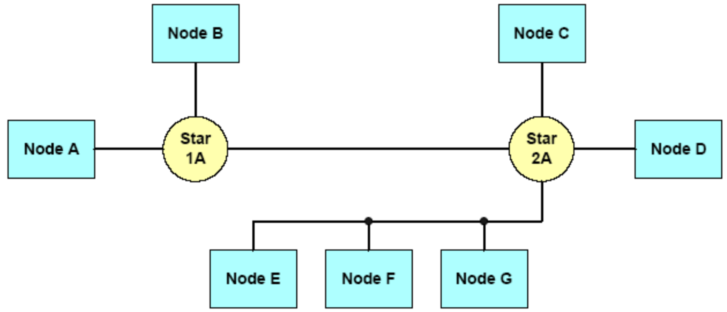


Figure 1-5: Single channel hybrid example.

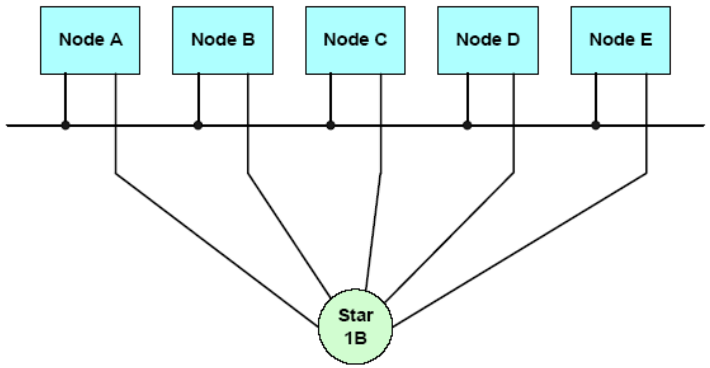


Figure 1-6: Dual channel hybrid example.

Comparison

H. Kopetz

A Comparison of TTP/C and FlexRay
Research Report 10/2001

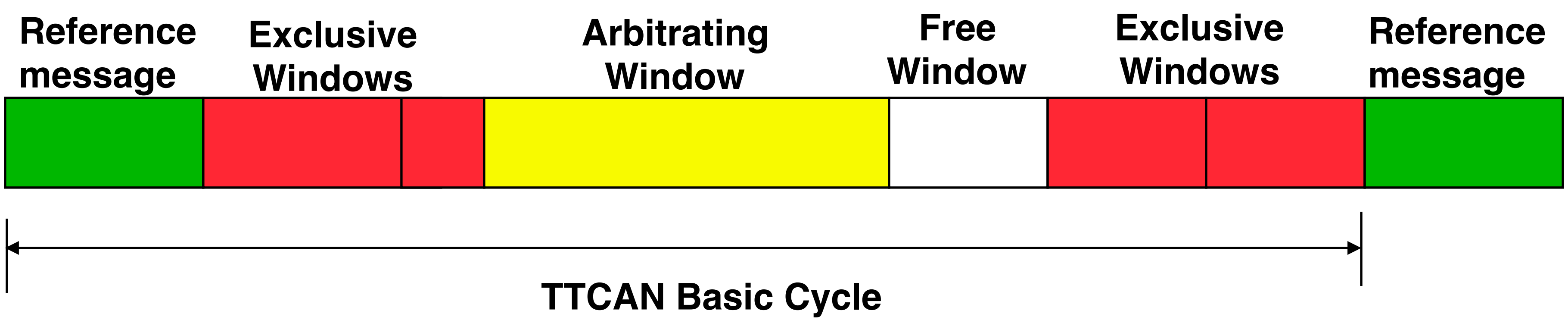
hk@vmars.tuwien.ac.at
Institut für Technische Informatik
Technische Universität Wien, Austria
May 9, 2001

Characteristic	TTP/C	FlexRay
Designed to meet automotive requirements	yes	yes
Priority in the "safety versus flexibility" conflict	safety	flexibility
Specification in the public domain	yes	no
Composability (precise interface specification in the value domain and in the temporal domain)	yes	no
Fault-tolerant clock synchronization	yes	yes
Replicated communication channels	yes	yes
Time-triggered message channels	yes	yes
Bus guardians to avoid babbling idiots	yes	yes
Bus guardian and protected node in different fault-containment regions	yes	no
Dynamic asynchronous message channels	yes, local	yes, global
Membership service	yes	no
Fault-hypothesis specified	yes	no
Never-give-up (NGU) strategy specified	yes	no
Critical algorithms formally analyzed	yes	no
Handling of outgoing link failures	yes	?
Handling of SOS failures	yes	?
Handling of Spatial Proximity failures	yes	?
Handling of Masquerading failures	yes	?
Handling of babbling idiot failures	yes	?
Transmission speed planned up to	25 Mbits/sec	10 Mbits/sec
Message data field length up to	236 bytes	12 bytes
Physical layer	copper/fiber	copper/fiber
CRC field length	3 bytes	2 bytes
Maximum achievable data efficiency for time-triggered messages in a 10Mbit/second system, interframe gap 5 microseconds.	95.8 %	45.7 %
Scalability: Maximum achievable data efficiency for time-triggered messages in a 100Mbit/second system, interframe gap 5 microseconds.	78 %	14.5%
Number of oscillators in a system with 10 ECUs	12	30
First system available on the market	1998	planned 2002
Architecture validated by fault injection	yes	no
Architecture viable for aerospace applications	yes	?

Time Triggered CAN
TTCAN

Time Triggered CAN: TTCAN (Führer, Müller, Dieterle, Hartwich, Hugel, Walther,(Bosch))

Basic Cycle and Time Windows



reference message: indicates the start of a cycle,
exclusive window : used for critical periodic state messages,
arbitrating window: used for spontaneous state and event messages,
free window : window for further extensions and gap to the next exclusive window.

RETRANSMISSIONS ARE GENERALLY NOT ALLOWED IN TTCAN !!

Scheduling a Basic cycle on a node

Node n

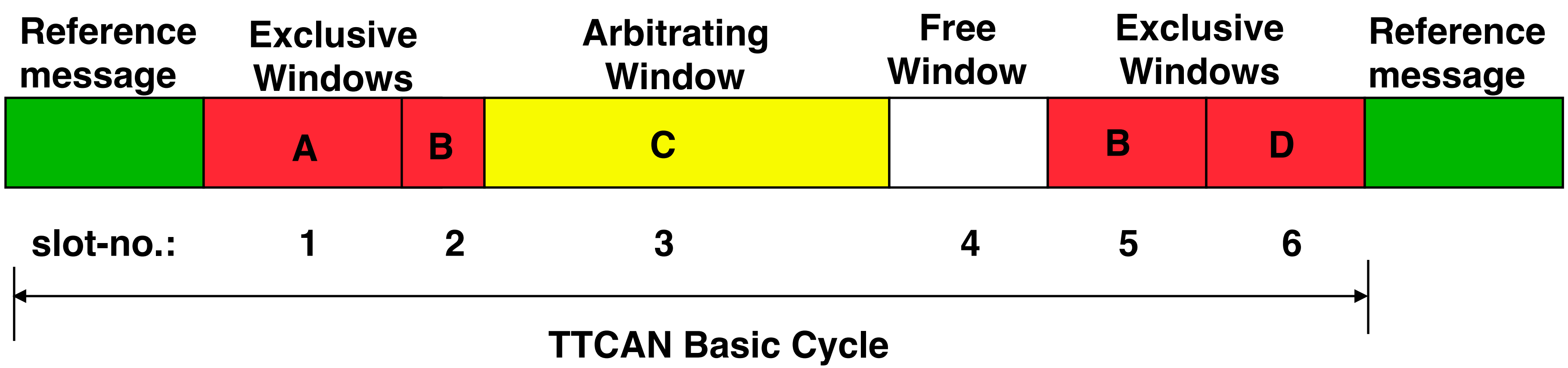
Send msg B in slot 2 and 5

Send msg F in slot 3

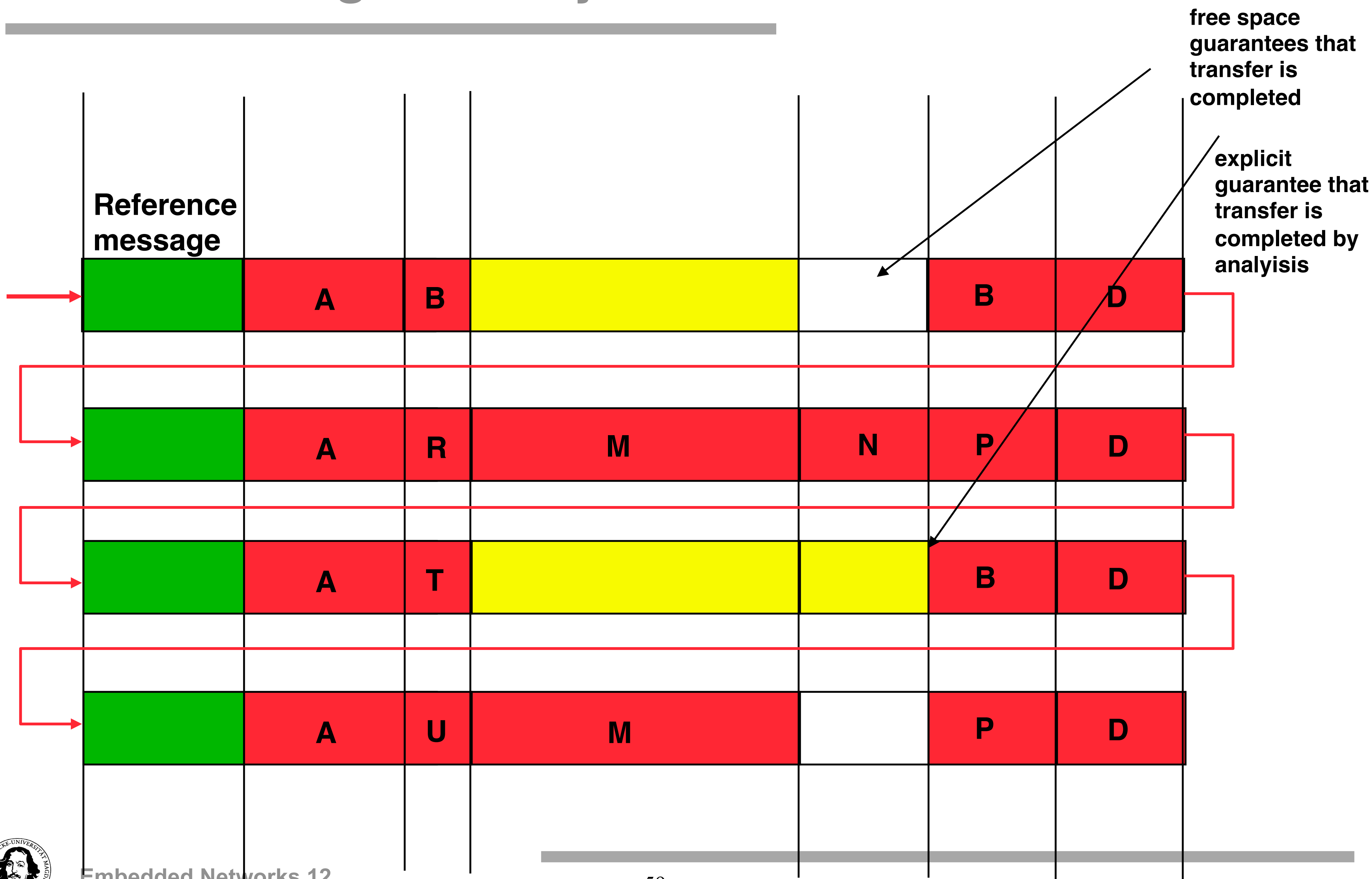
Receive msg D in slot 6

Constraint:

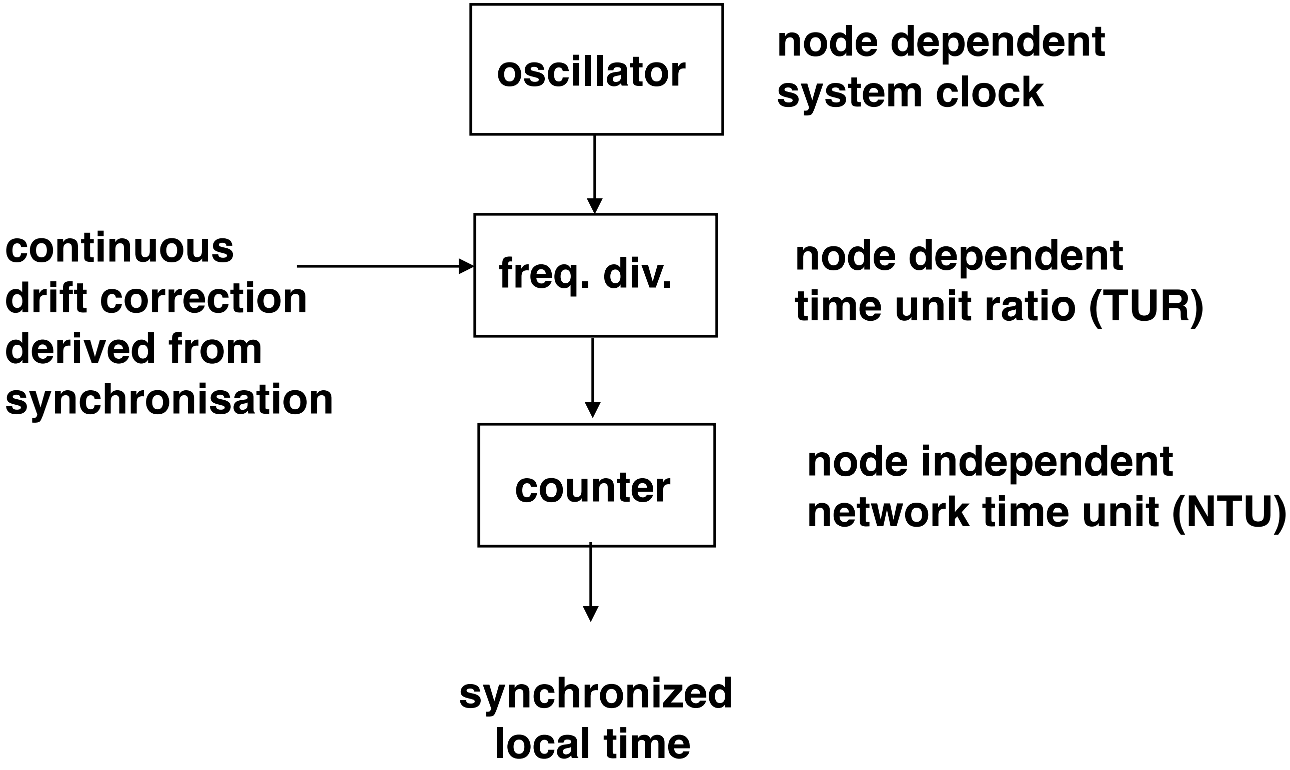
A message transfer in an arbitrating window must be successfully completed before the start of an exclusive window.



Concatenating Basic Cycles to a MATRIX CYCLE



Time and clock synchronization in TTCAN



Synchronization based on the existence of a Time Master.

All nodes take a snapshot of their local time at the SoF (Start of Frame) bit of the reference message.

Because of dependability reasons, TTCAN supports redundant Time Masters.

Arbitration among Time Masters is based on the priority scheme of CAN.

Conclusion

TT-CAN adds predictability to CAN

TT-CAN considers periodic message transfer

Fault handling differs substantially from Standard CAN

Clock synchronization is supported by hardware

Hybrid approaches are available in the scientific community

Coexistence of time-triggered and event-triggered mechanisms on the CAN-Bus

???

Is it possible and what are the trade-offs?

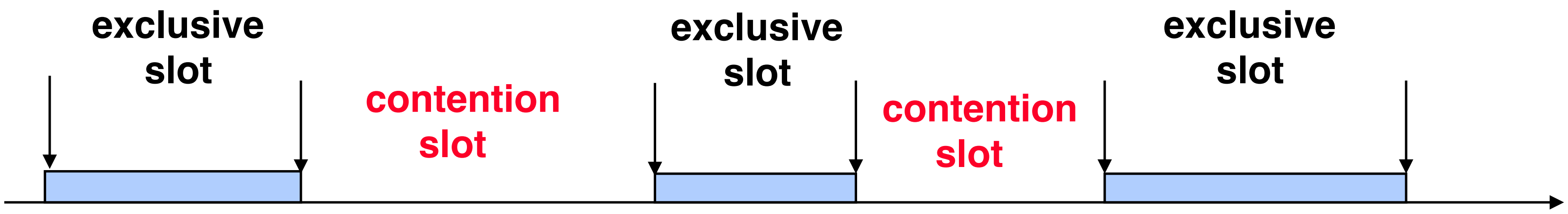
1. **Time Triggered CAN: TTCAN (Führer, Müller, Dieterle, Hartwich, Hugel, Walther,(Bosch))**
2. **Dynamic Priorities (Kaiser, Livani)**

Integration of TT- and ET- communication

by dynamic priorities

M.A. Livani, J. Kaiser, W.J. Jia. Scheduling Hard and Soft Real-Time Communication in the Controller Area Network (CAN), *23rd IFAC/IFIP Workshop on Real Time Programming*, Shantou, China, June 1998.

Basic Idea: Reserve slots for hard real-time traffic and schedule soft real-time traffic in the remaining slots

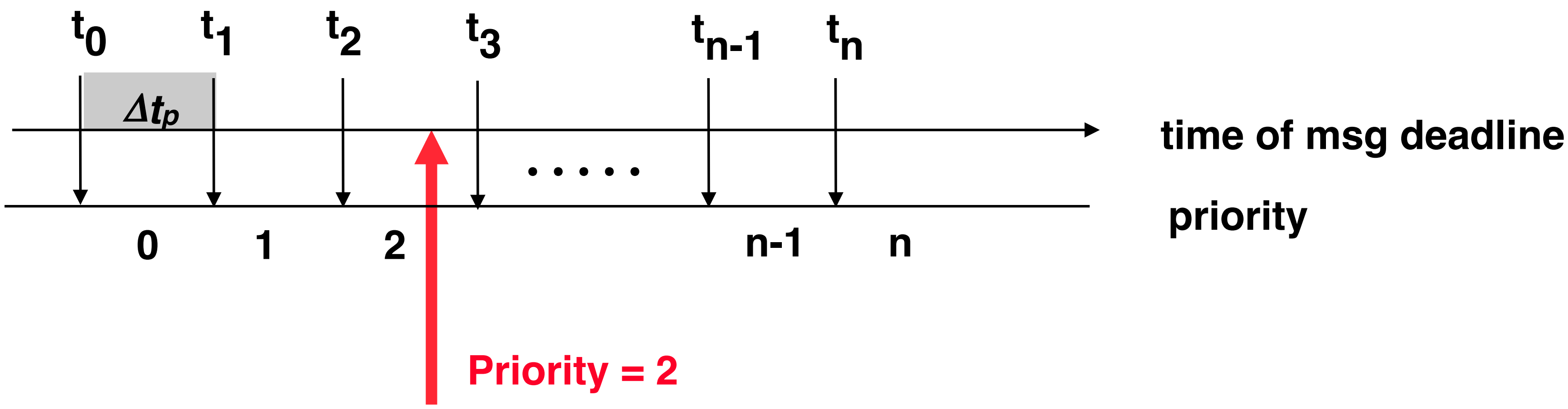


The priority scheme is used to enforce high priority message transmission in the exclusive slots.

What is the advantage over TDMA?

Mapping Deadlines to Priorities

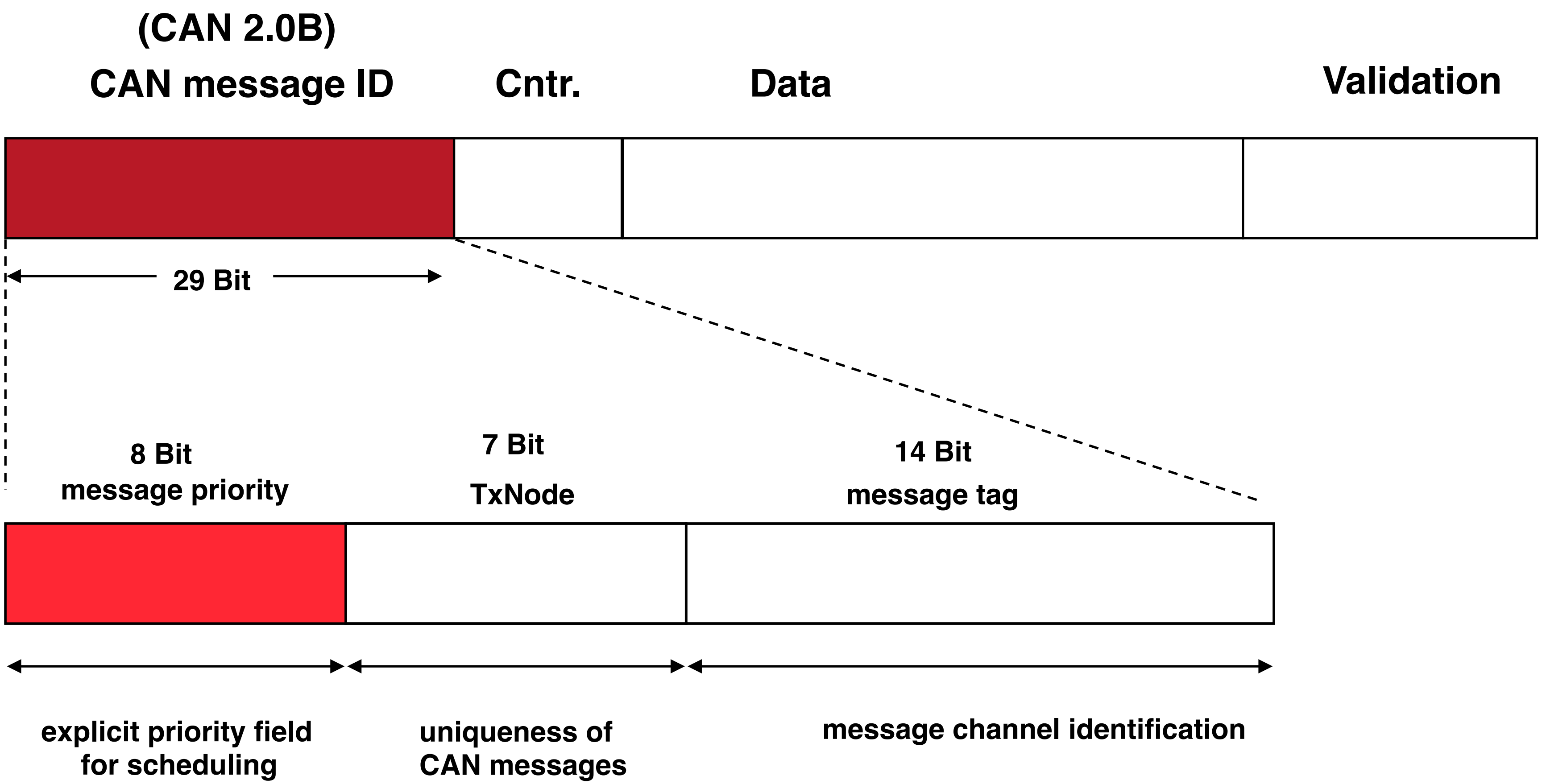
- Messages have deadlines
- Deadlines can be transformed into priorities



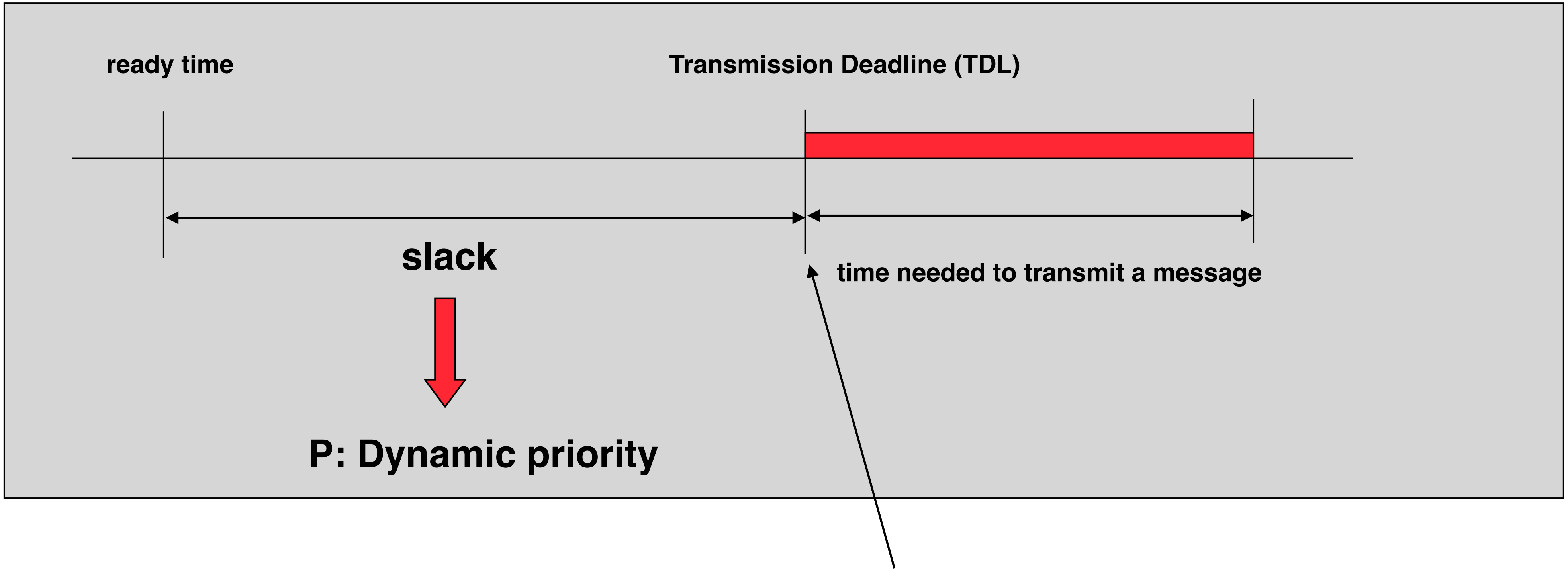
Wanted

a global priority-based message dispatcher

Structuring the CAN-ID

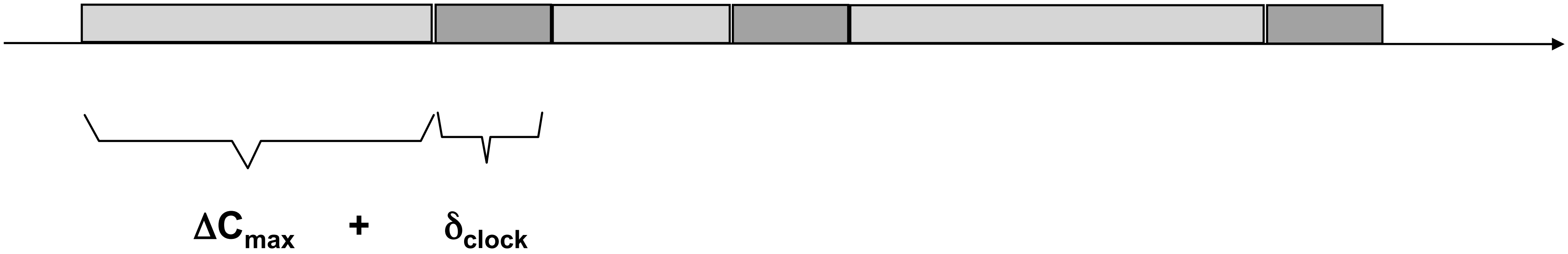


Scheduling messages with guarantees



At TDL: $P_{HRTM} > P_{SRTM} > P_{NRTM}$

How many HRT-slots can be guaranteed ?



ΔC_{max} max. time interval (possibly under failure assumptions), which is neccessary to safely transmit a message to the destination

ΔC_{max} is a worst case assumption under all anticipated load and failure conditions

δ_{clock} max. offset, i.e. the difference between any two local clocks

CAN Inaccessibility Times*

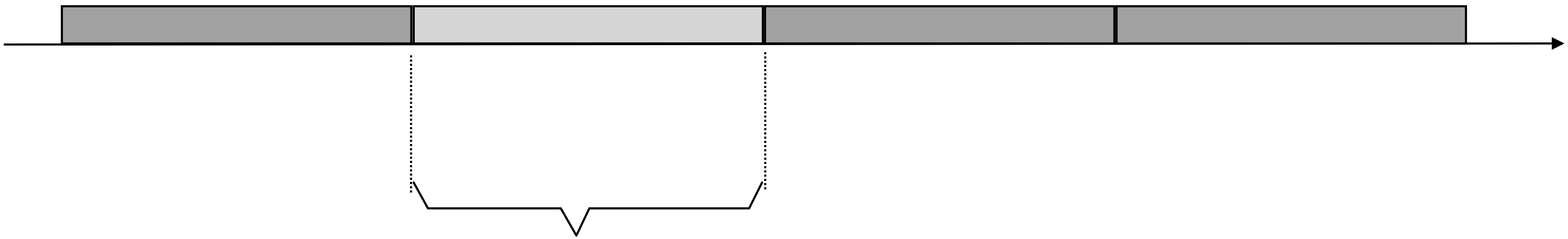
Data Rate 1 Mbps , Standard Format

Scenario	t_{inacc} (µs)	
Bit Errors	155.0	← worst case single
Bit Stuffing Errors	145.0	
CRC Errors	148.0	
Form Errors	154.0	
Ack. Errors	147.0	
Overload Errors	40.0	
Reactive Overload Errors	23.0	
Overload Form Errors	60.0	
Multiple Consecutive Errors (n=3)	195.0	
Multiple Successive Errors (n=3)	465.0	
Transmitter Failure	2480.0	← worst case multiple
Receiver Failure	2325.0	

P. Verissimo, J. Ruffino, L. Ming:" How hard is hard real-time communication on field-busses?"

Utilization of CAN for HRT-messages

$\Delta C_{max} + \delta_{clock}$



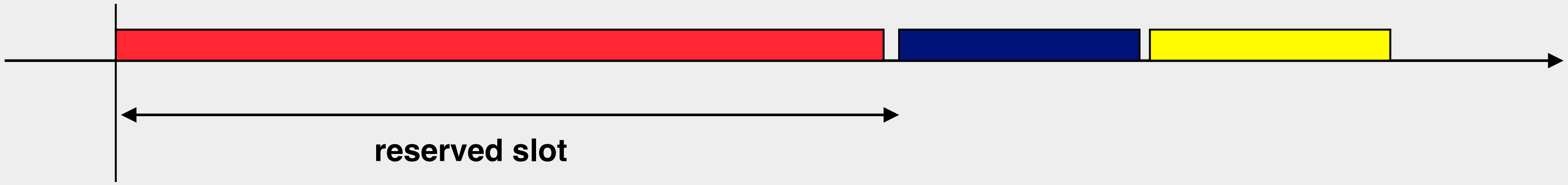
$$2 \bullet t_{message} + n \bullet t_{transmitter\ failure} + m \bullet t_{message\ failure} + \delta_{clock}$$

fault assumption n	m	ΔC_{max} + 50 µs δ_{clock} (µs)	HRT messages / sec. #
0	0	358	2793
0	1	532	1880
0	3	880	1136
1	0	2988	335
1	3	3664	273

Benefits of the approach

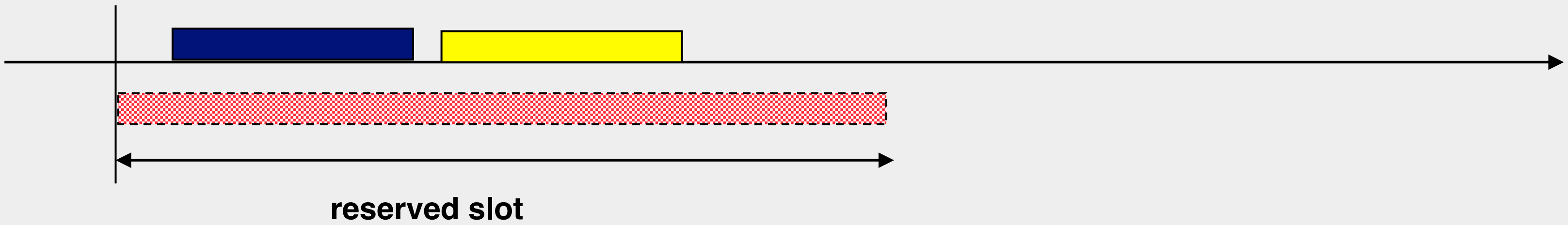
Media access controlled by global time only (TDMA)

All nodes need global time
Unused slots remain unused



Media access in a system controlled by our priority scheme

Only nodes with HRT-msg need global time
Unused slots can be used by msg which are ready to be transmitted



Cost-Performance Trade-off

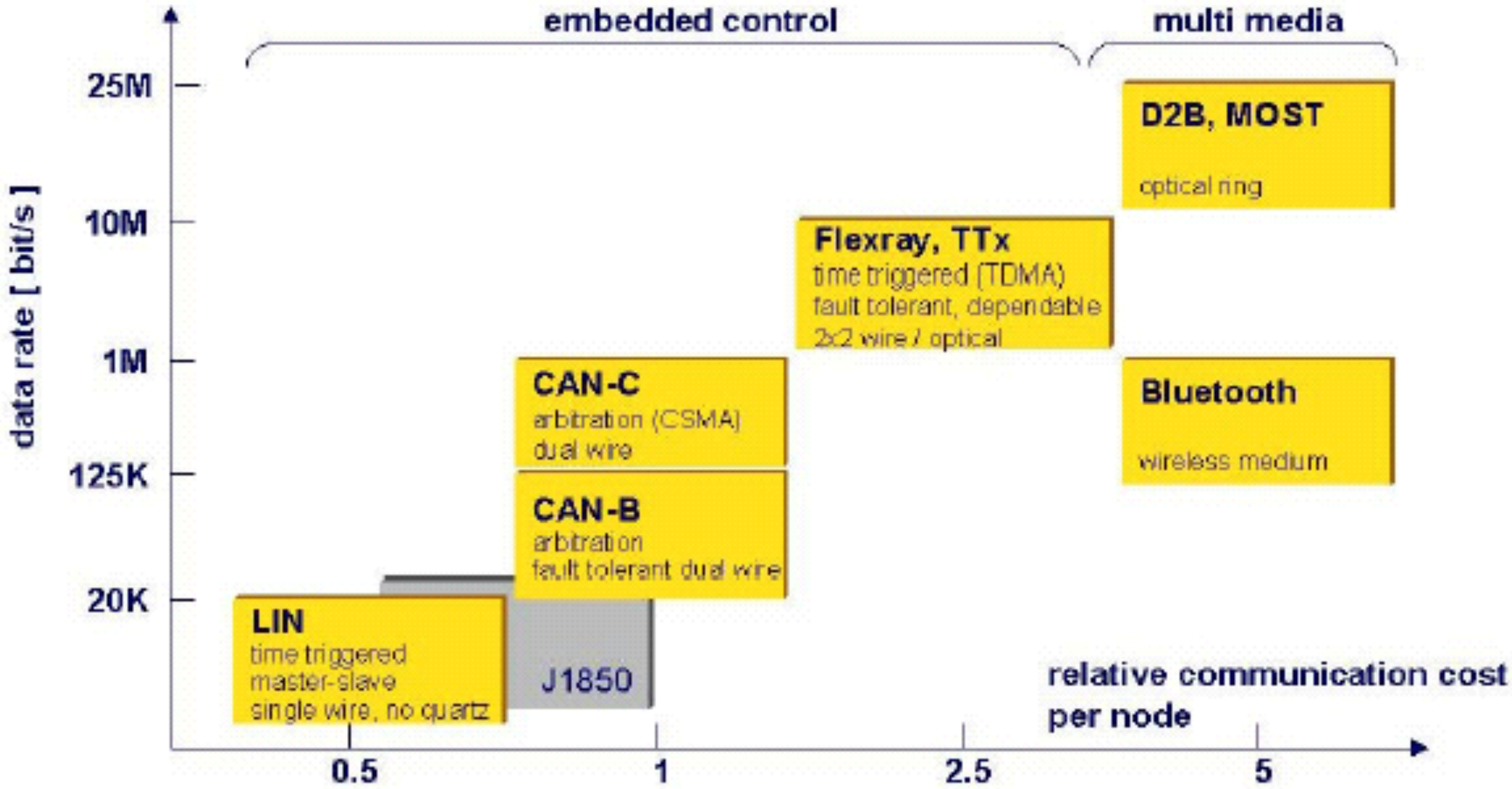
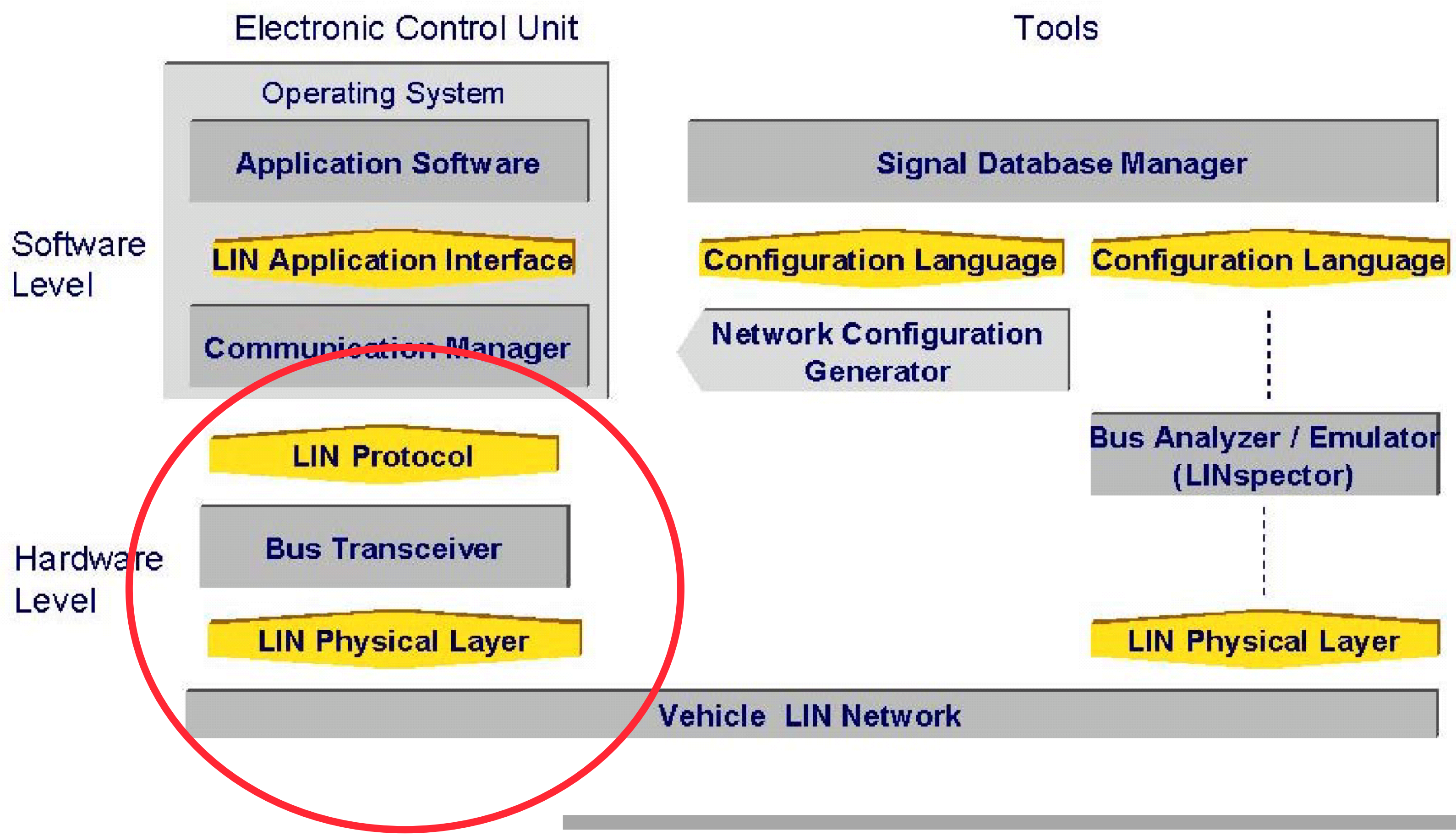


Figure 1: Major Network Protocols in Vehicles

LIN (Local Interconnect Network)

LIN Specification Package, Revision 1.2, Nov. 17, 2000



Properties of LIN

. single-master / multiple-slave concept

. low cost silicon implementation based on common UART/SCI interface hardware, an equivalent in software, or as pure state machine.

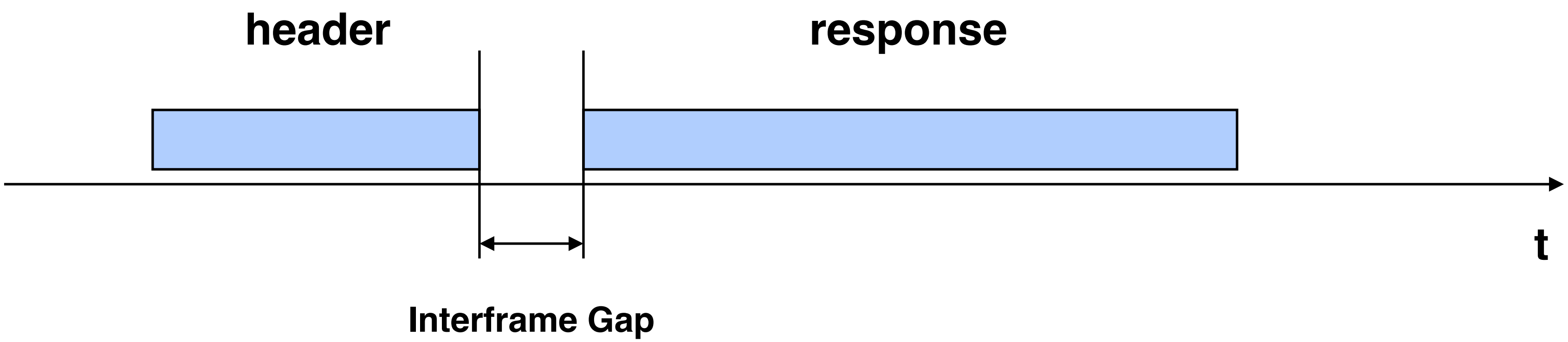
. self synchronization without quartz or ceramics resonator in the slave nodes

. guarantee of latency times for signal transmission

. low cost single-wire implementation

. speed up to 20kbit/s.

Master-Slave communication in LIN

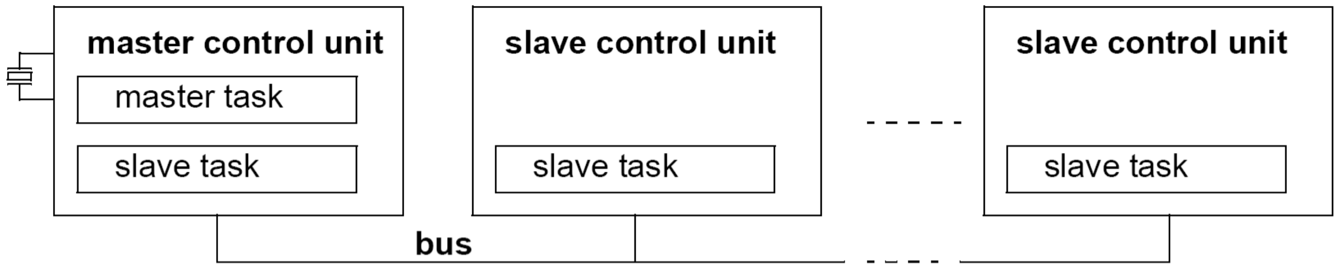


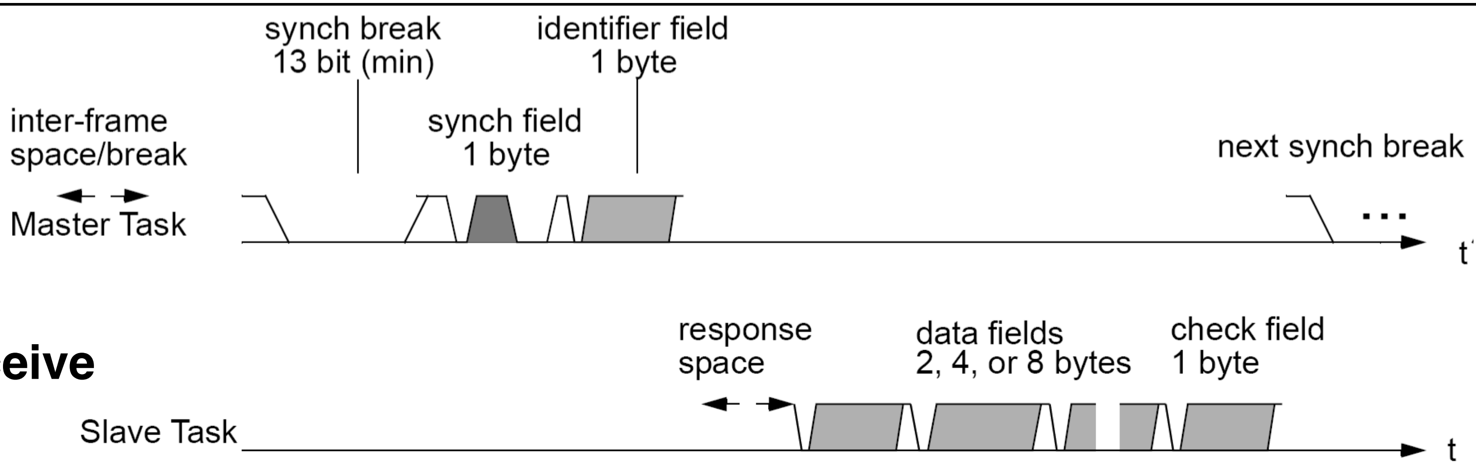
Header:
- serves for the synchronisation of slaves
- specifies the sequence and length of the fields in the data frame

LIN (Local Interconnect Network)

LIN Specification Package, Revision 1.2, Nov. 17, 2000

time dependent master/slave protocol





only 1 slave is allowed to respond!
But all slaves receive the response.

LIN Specification Package, Revision 1.2, Nov. 17, 2000

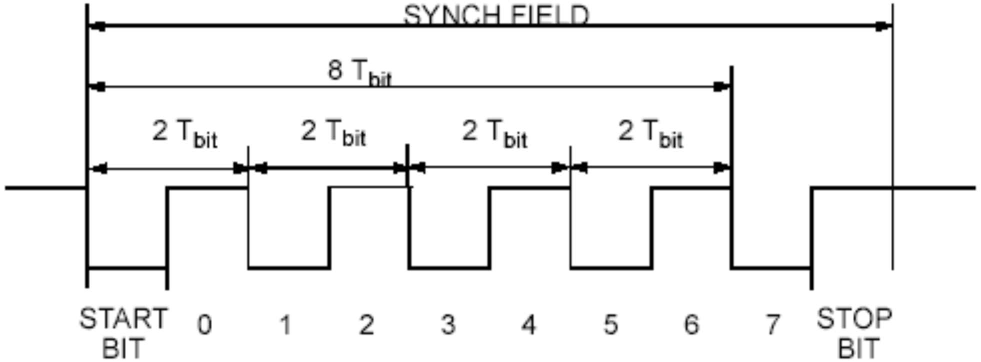


Figure 9.1: SYNCHRONIZATION FIELD

Synch. Feld 0x55

clock tolerance	Name	$\Delta F / F_{Master}$
master node	$F_{TOL\_RES\_MASTER}$	$< \pm 0.5\%$
slave node with quartz or ceramic resonator (without the need to synchronize)	$F_{TOL\_RES\_SLAVE}$	$< \pm 1.5\%$
slave without resonator, lost synchronization	$F_{TOL\_UNSYNCH}$	$< \pm 15\%$
slave without resonator, synchronized and for a complete message	F_{TOL_SYNCH}	$< \pm 2\%$

Table 8.1: Oscillator Tolerance

LIN Specification Package, Revision 1.2, Nov. 17, 2000

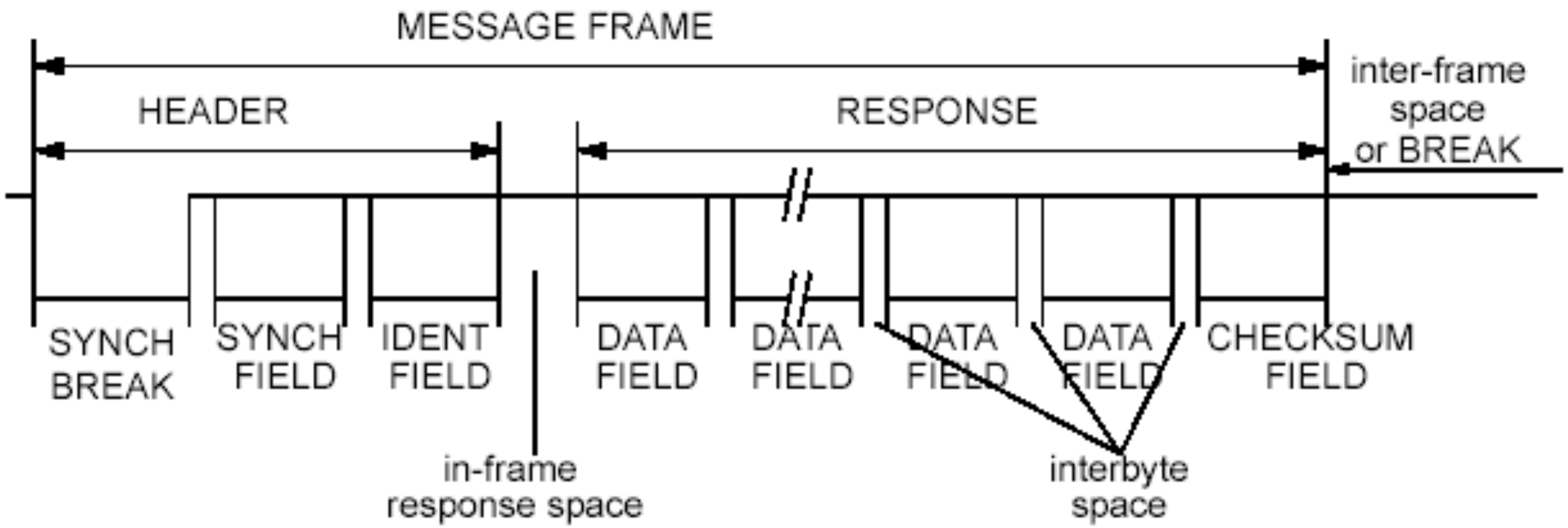


Figure 3.1: LIN MESSAGE FRAME

LIN Specification Package, Revision 1.2, Nov. 17, 2000

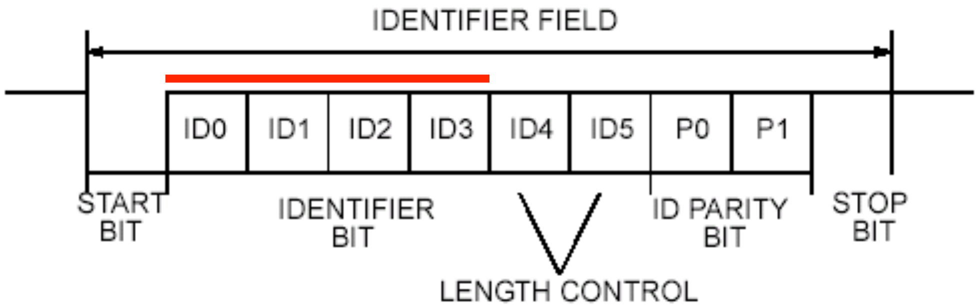


Figure 3.5: IDENTIFIER FIELD

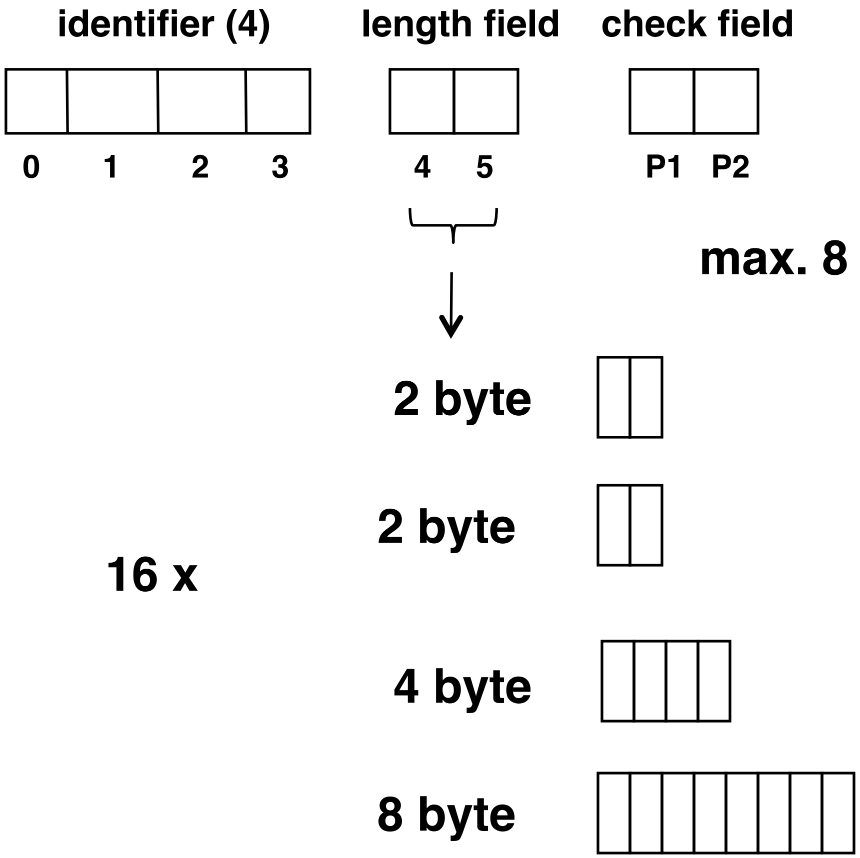
64 identifiers

divided in 4 groups of length: 2,4, and 8 bytes

An ID identifies the content of a message, not the sender or receiver !

Slaves can be added or removed without changing any software in the other slaves.

LIN frame format



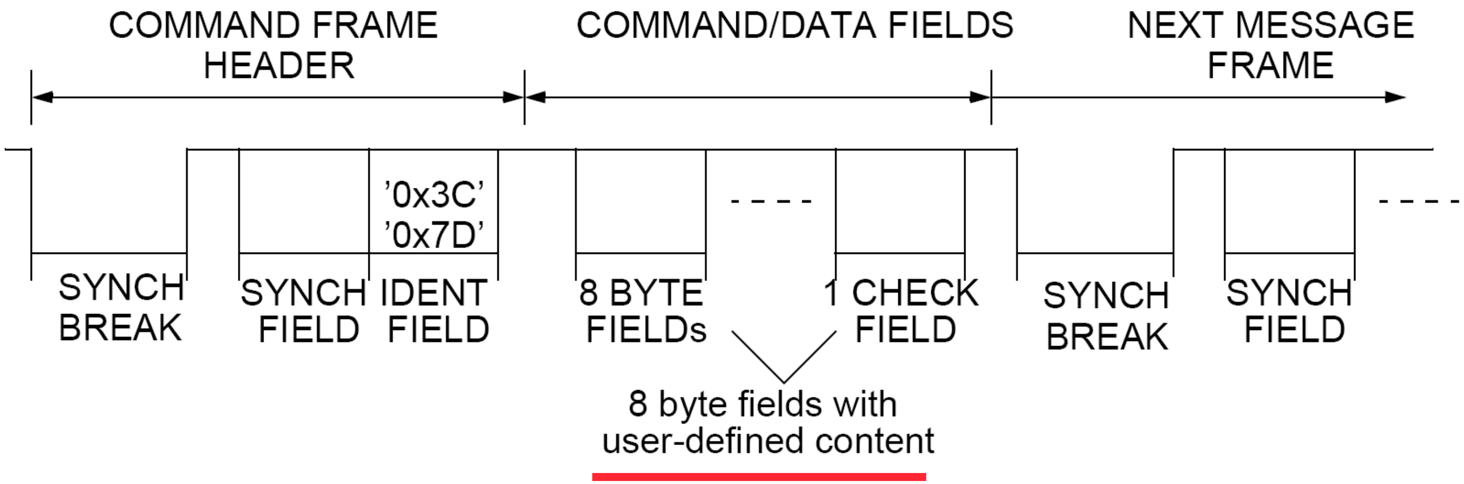
content-based addressing

max. 8 Byte response frame

ID5	ID4	N_{DATA} (number of data fields) [byte]
0	0	2
0	1	2
1	0	4
1	1	8

reserved IDs: Master request Frame (0x3C), Slave Response Frame (0x3D)
Extended Frames (User 0x3E, Reserved 0x3F)

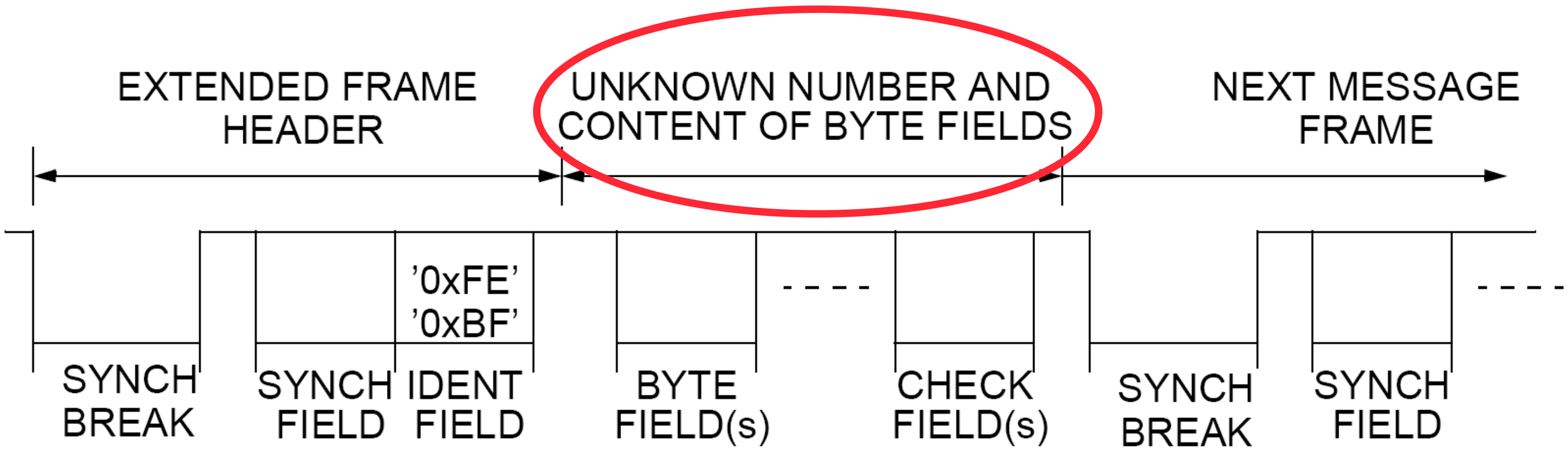
LIN Master Request Frame



Download of data to the slave.
Request of data from the slave.

Multiple 8 byte fields possible!
Slave address is part of the command fields.

LIN Extended Frame



slaves, whiche are not addressed (interested resp.) wait until the next SyncBreak!

Error detection capabilties of LIN:

Bit-Error

Checksum-Error

Identifier-Parity-Error

Slave-Not-Responding-Error

Inconsistent-Synch-Field-Error

No-Bus-Activity

Automotive and highly dependable Networks

TTP/C
Byteflight
FlexRay
Time Triggered CAN (TTCAN)
LIN