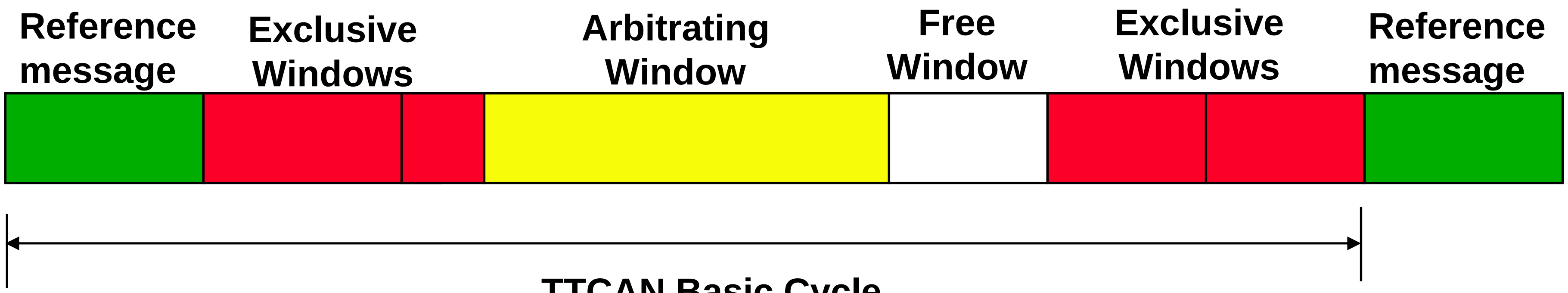
Time Triggered CAN
TTCAN

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 !!

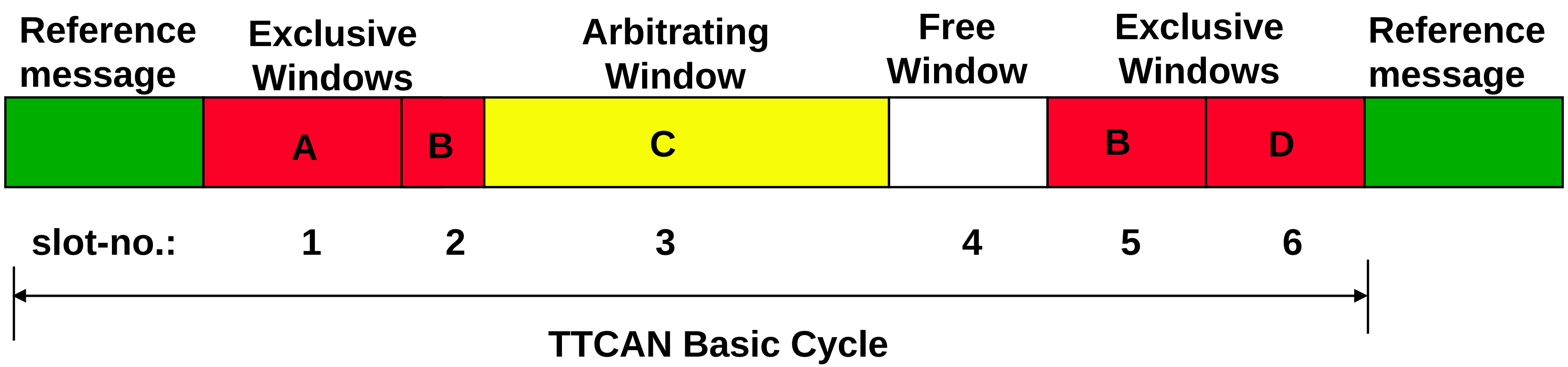
Scheduling a Basic cycle on a node

Node n

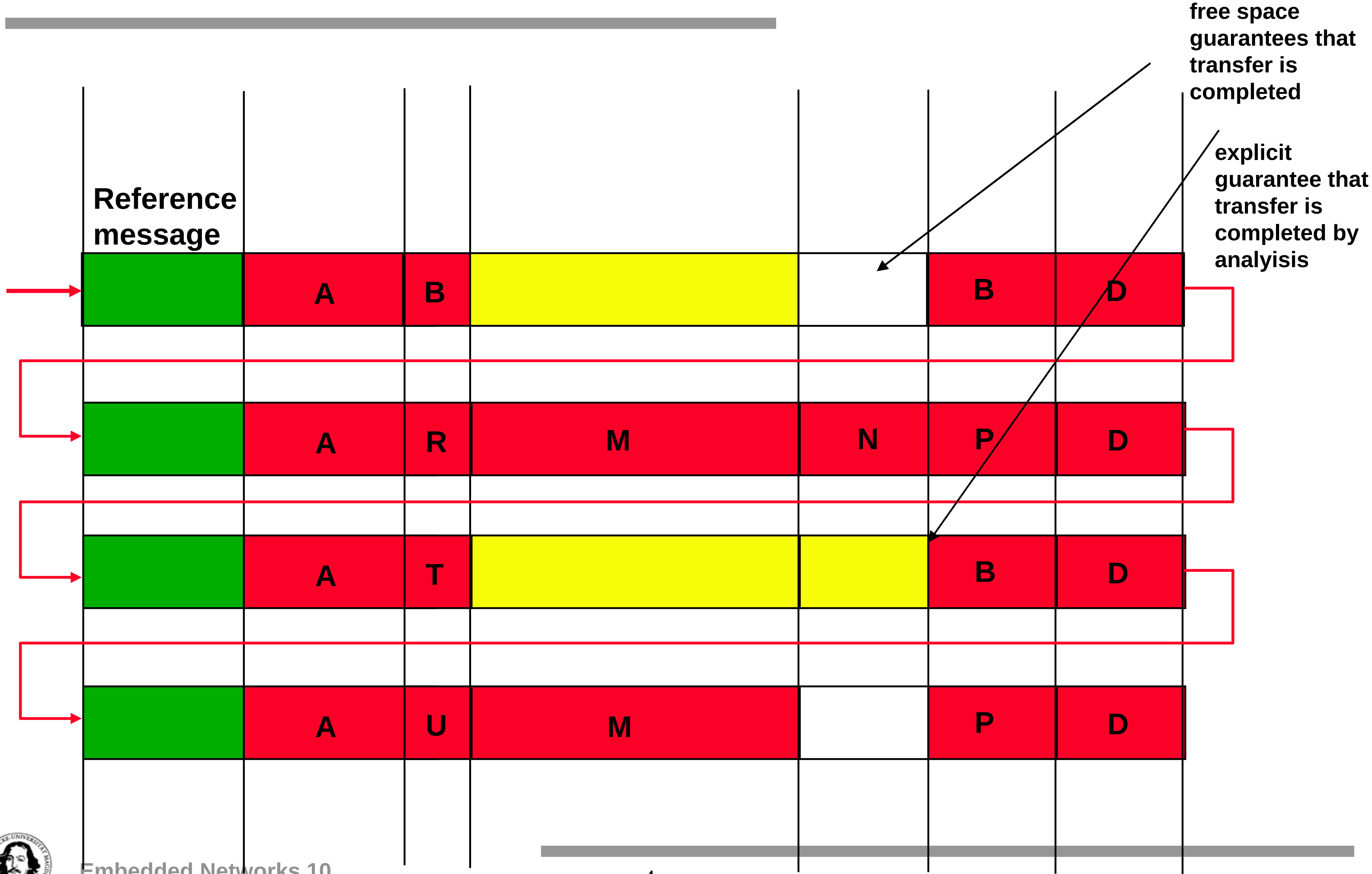
- Send msg B in slot 2 and 5
- Send msg F in slot 3
- Receive msg D in slot 6

Constraint:

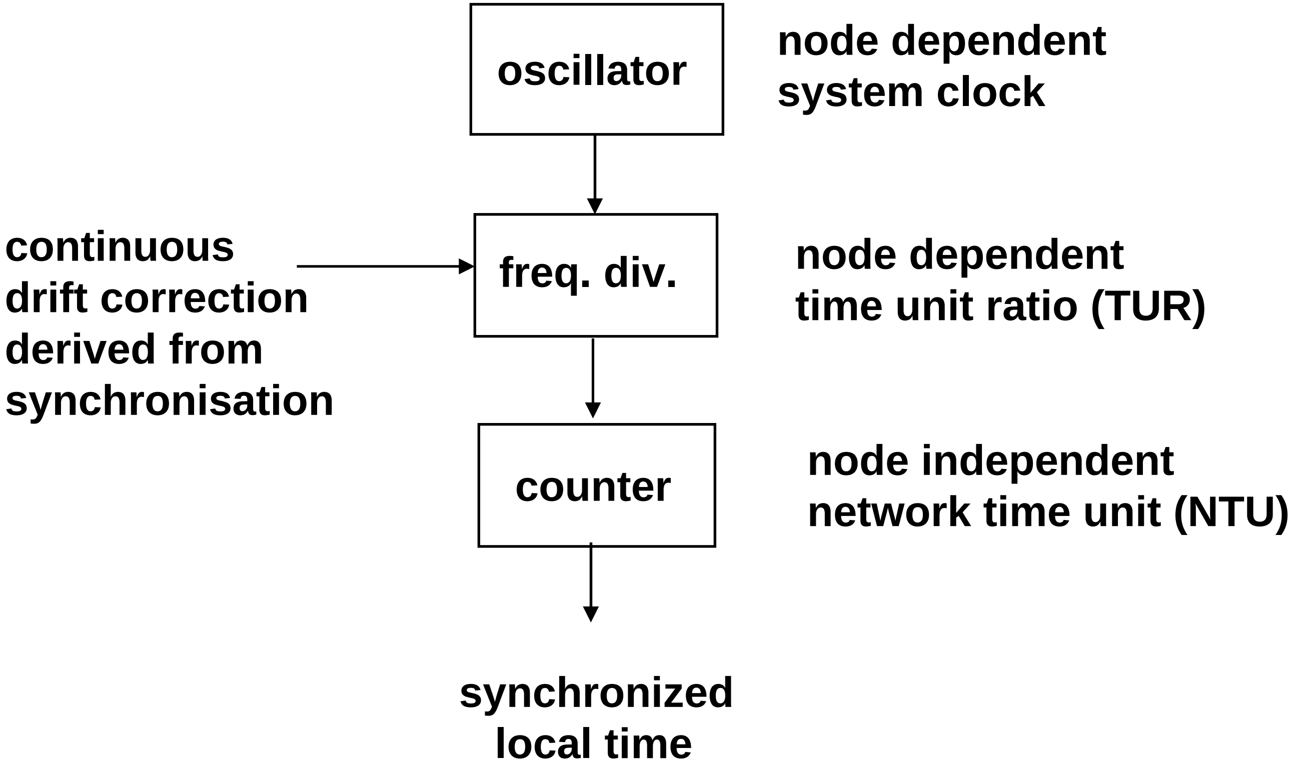
A message transfer in an arbitrating window must be successfully completed before the start of an exclusive window.



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

???

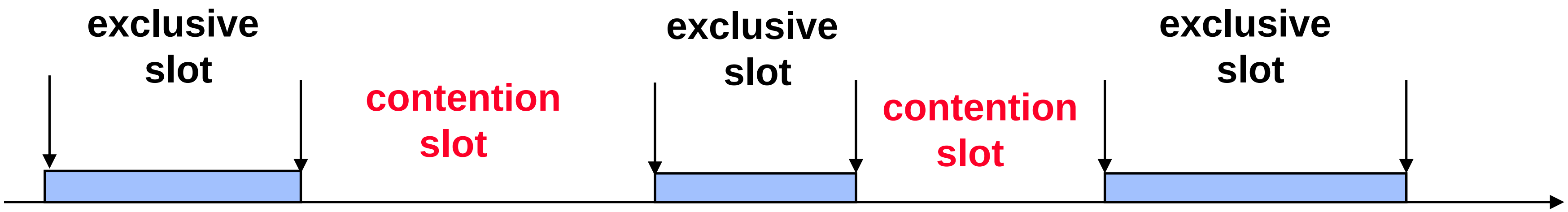
Is it possible and what are the trade-offs?

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

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

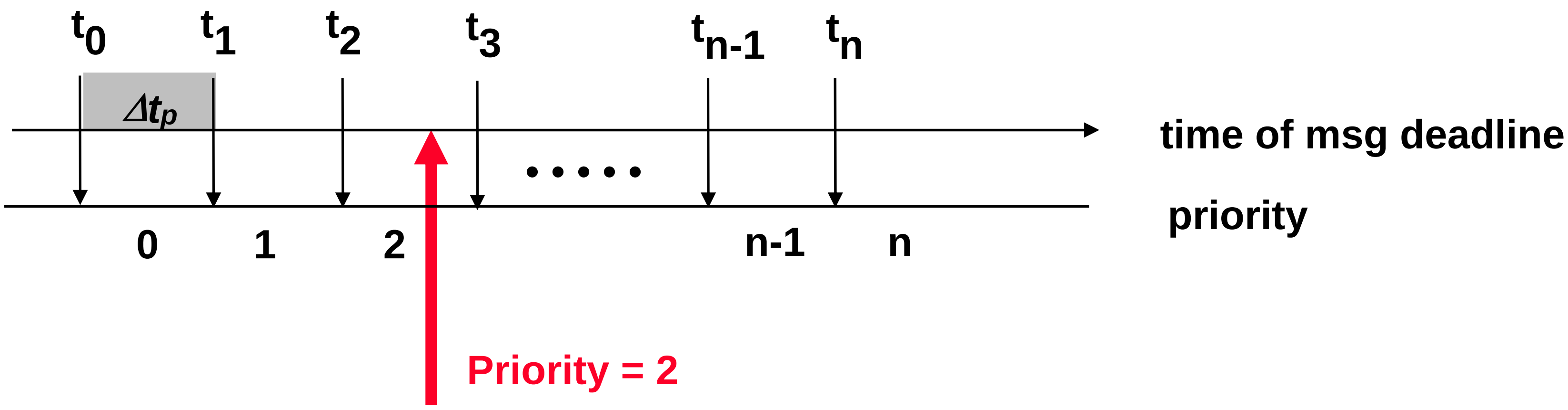


The priority scheme is used to enforce high priority message transmission in the exclusive slots.

What is the advantage over TDMA?

Mapping Deadlines to Priorities

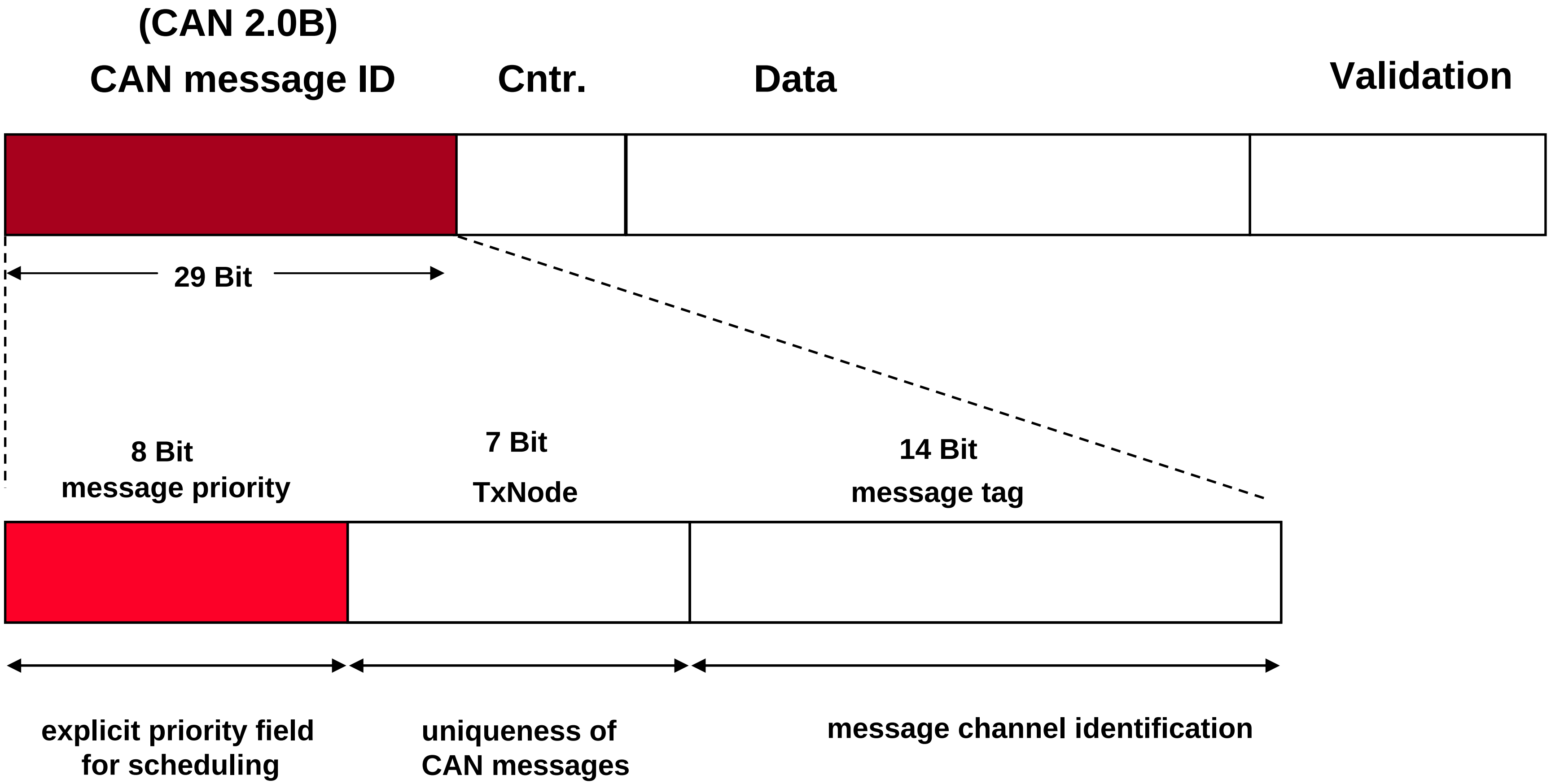
- Messages have deadlines
- Deadlines can be transformed into priorities



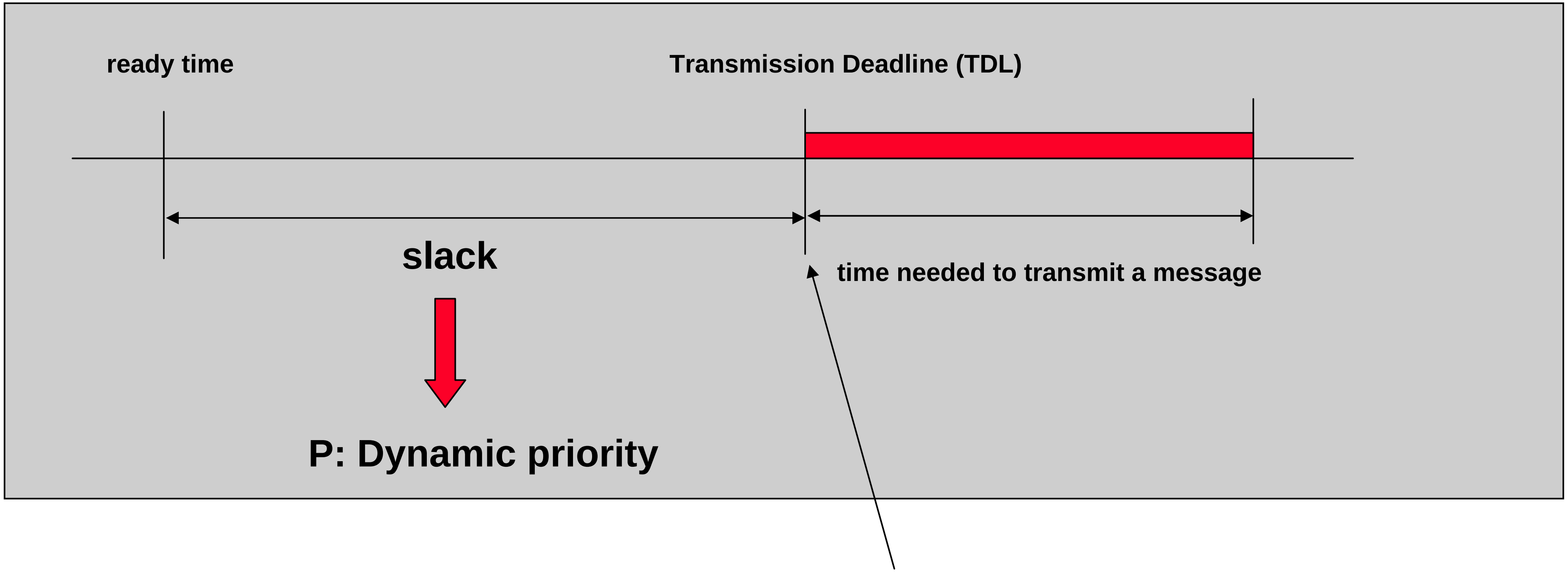
Wanted

a global priority-based message dispatcher

Structuring the CAN-ID

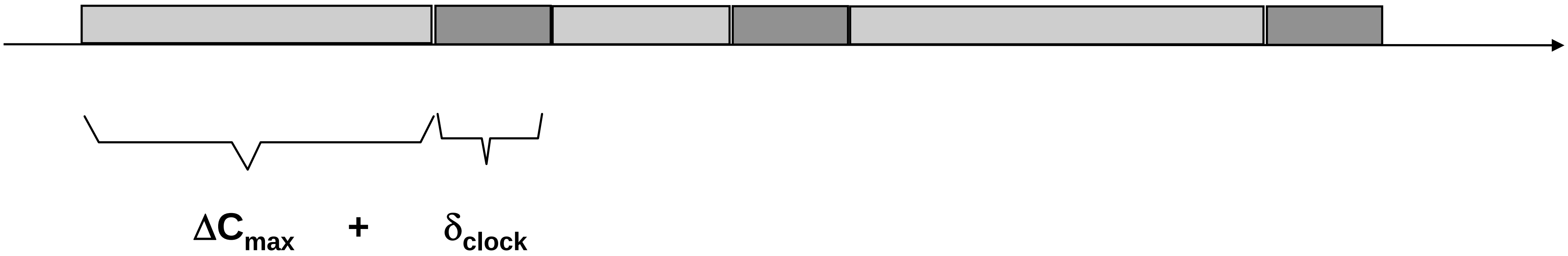


Scheduling messages with guarantees



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*

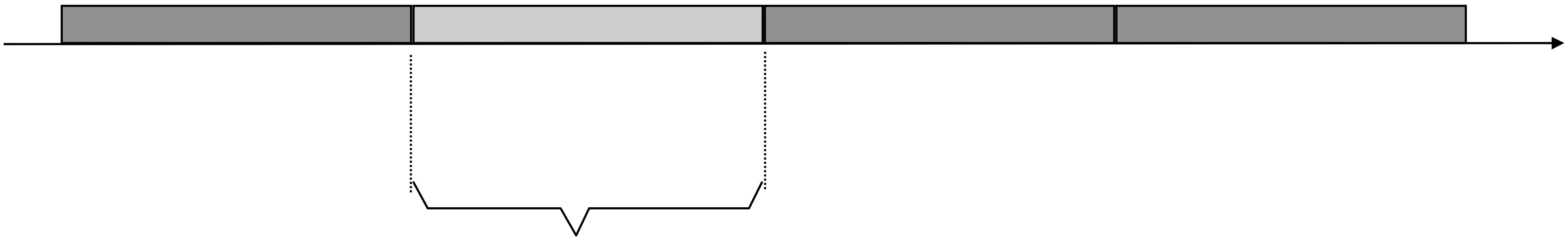
Data Rate 1 Mbps , Standard Format

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

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



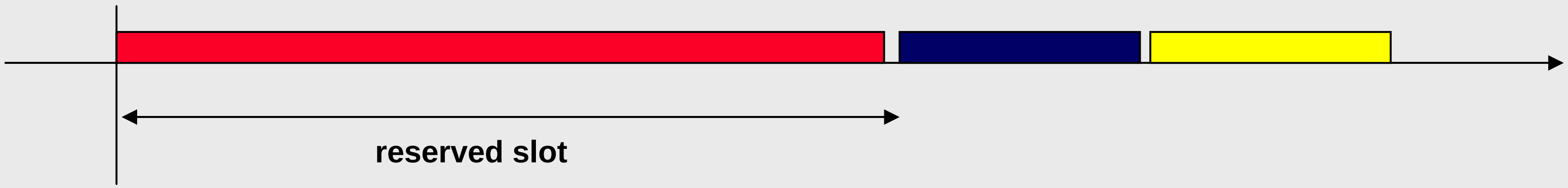
$2 \cdot t_{message} + n \cdot t_{transmitter\ failure} + m \cdot t_{message\ failure} + \delta_{clock}$

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

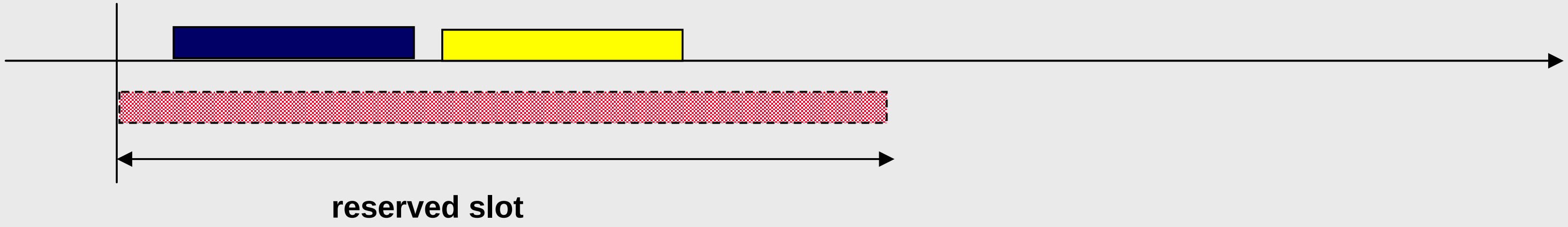
Media access controlled by global time <u>only</u> (TDMA)

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

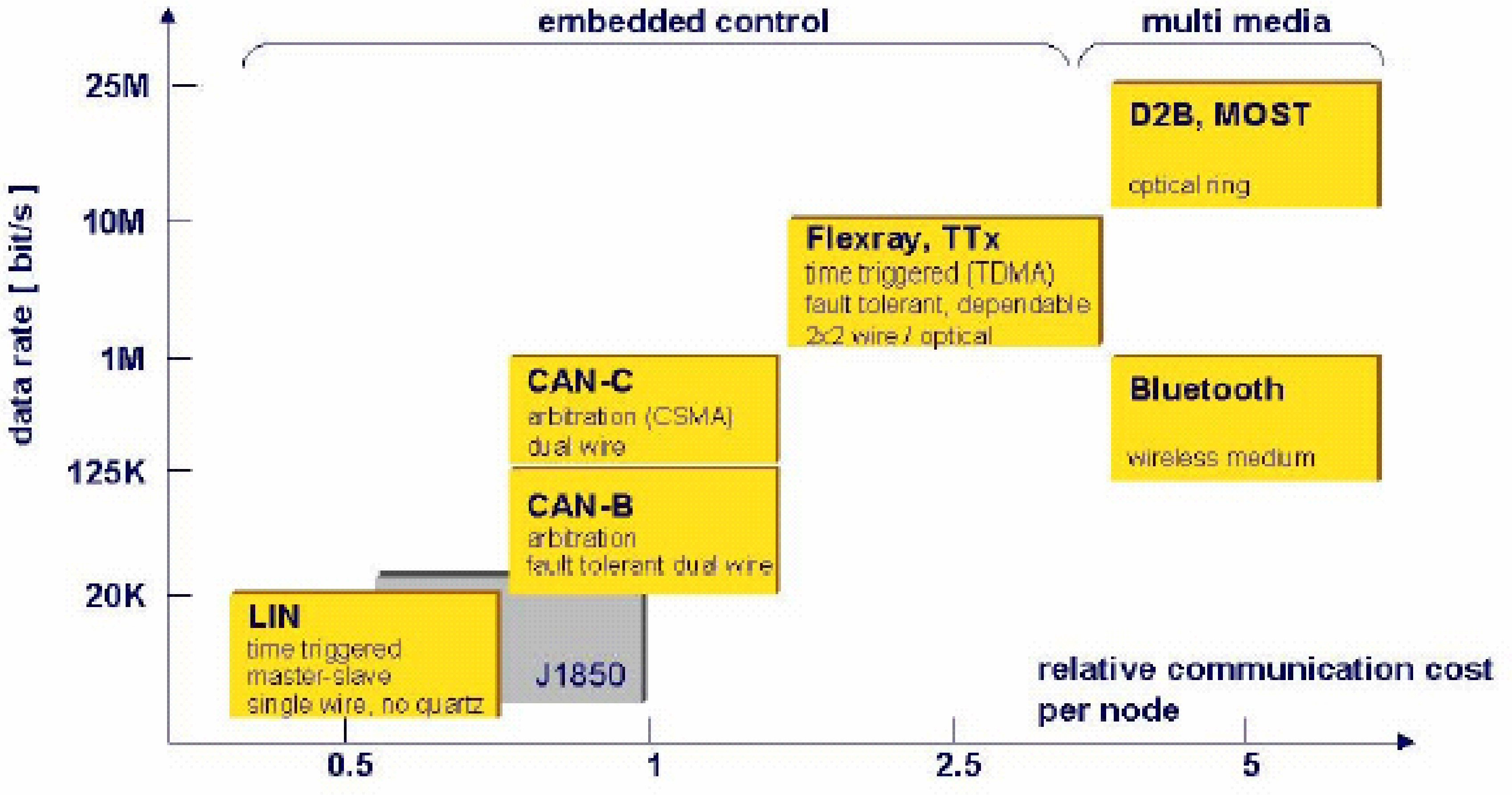


Figure 1: Major Network Protocols in Vehicles

Protocols for less critical, simple sensor-actuator networks:

TTP/A (Time Triggered Protocol for SAE class A applications)

LIN (Local Interconnect Network)

- Master/Slave protocols

- low dependability requirements

- free-runing low cost oscillators should be possible

- physical „Single-Wire-Network“ (asynch. serial interface)

- low bandwidth requirements

- low cost

Transmission speed up to	LIN	TTP/A
20 kbits/second	ISO 9141 (ISO-K)	ISO 9141 (ISO-K)
1 Mbit/second	not specified	RS 485 or CAN
above 1 Mbit/second	not specified	fiber optics

Table 4: Transmission speed of LIN and TTP/A

TTP/A

H. Kopetz: Lit. Einführung,
H. Kopetz, W. Elemenreich, C. Mack: A Comparison of LIN and TTP/A, Research report 4/2000,
Institut für Technische Informatik, TU Wien

- **real time data**
- **round: up to 64 byte**
- **broadcast**

- **managment and configuration data**
- **diagnostic interface**
- **point-to-point**

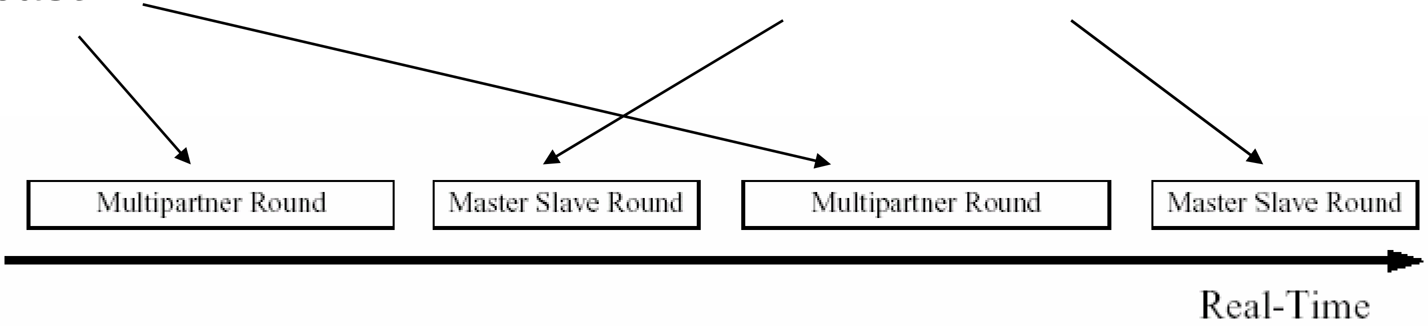
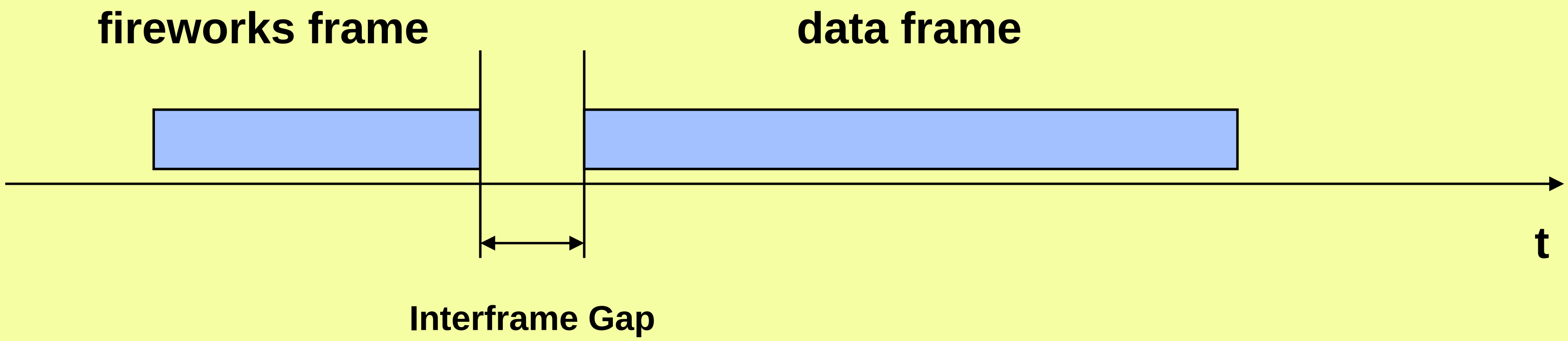


Figure 3: Traffic on the TTP/A Bus

3 different interfaces for slaves:

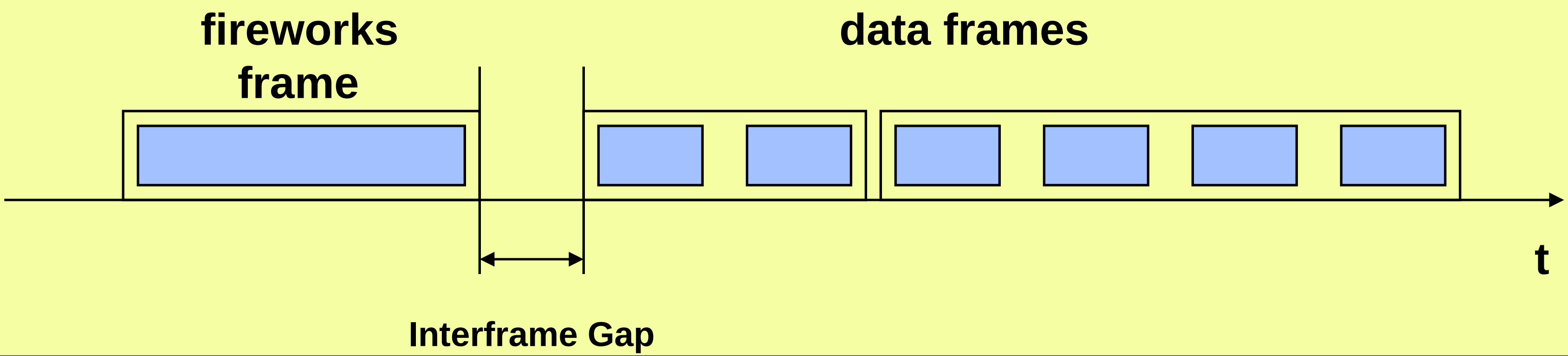
- **RMI : Real-Time message Interface**
- **DMI: Diagnostic message Interface**
- **CMI: Configuration Message Interface**

master-slave dialogue



<master-slave ("fireworks"), file op and identifier, record number, logical node name, check byte>

multi partner round



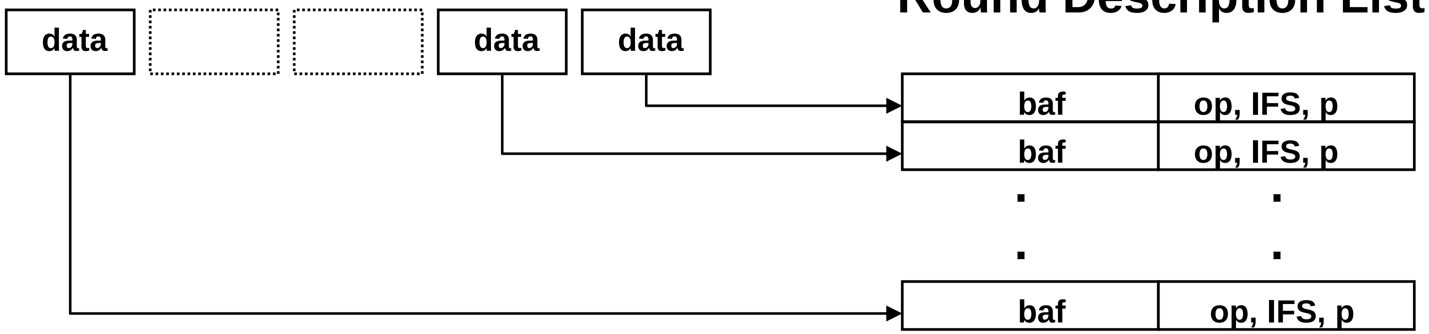
data centric communication model

- real time frames contain data only!
- all data is stored in the Interface File System (IFS).
- addresses to data are specified as IFS addresses.
- addresses are specified in the round description list (RODL), i.e. the time slot in which the message is transmitted is fixed according to the TT model.

baf: byte after fireworks
op: operation
IFS: IFS-Adresse
p: protection (checksum)

Multipartner Round

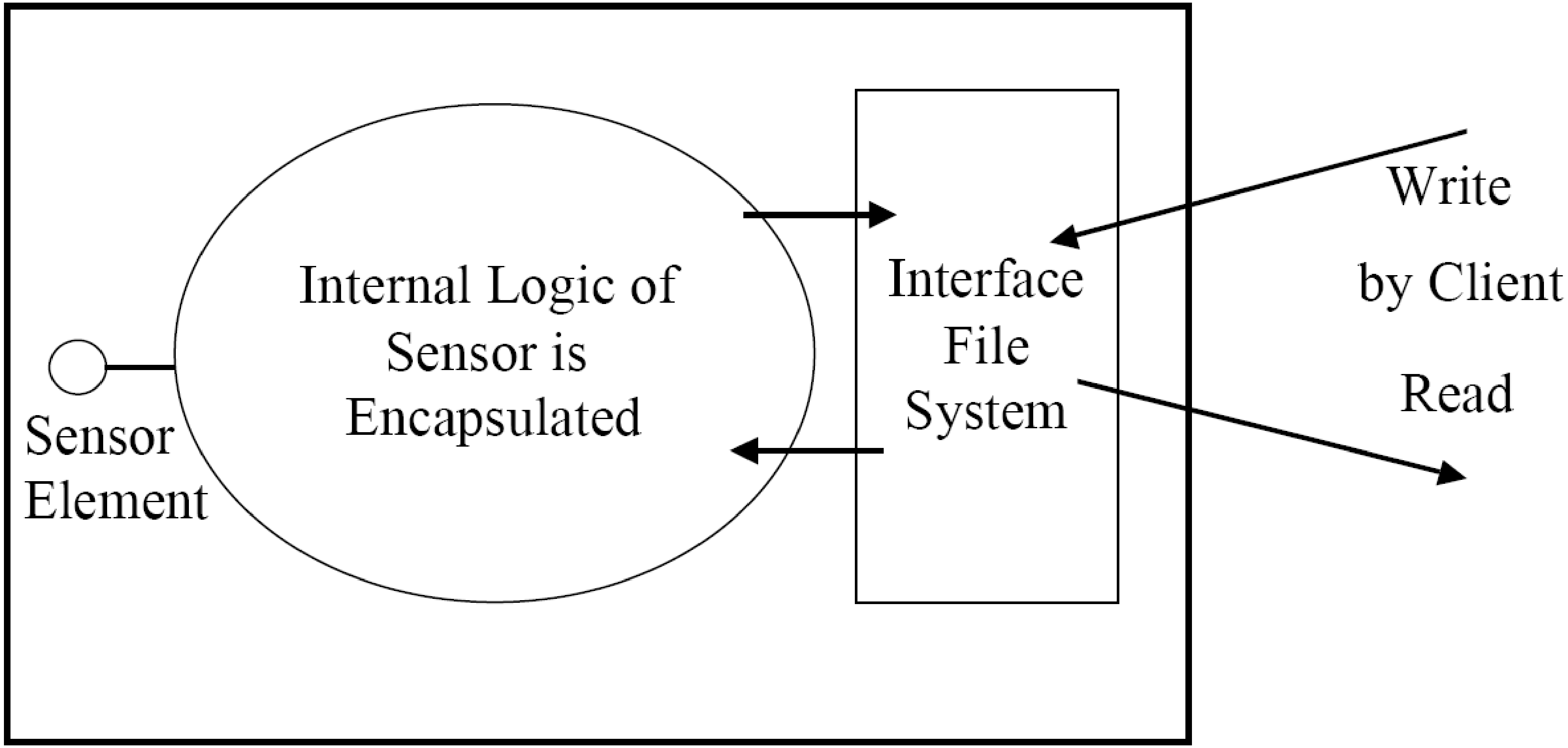
Round Description List: RODL



The RODL is also stored in the IFS and can be configured via the CMI.

There are max. 8 RODLs. RODL# is transmitted with a Hamming Distance of 4 (high protection against failures).

Programming model for smart transducers in the IFS



Address contains: < file, record, byte, checksum>

2^6 2^8 2^2

every node in the IFS supports:

up to	64	files
up to	256	records
with	4	bytes each

i.e. an address space of 2^{16} bytes/node.

and how to address the nodes ?

Every Smart Transducer has a unique physical name (8 bytes) consisting of:
- a node type name (series number)
- a node name within series (serial number)

During operation a node is addressed by a one-byte logical name that is unique within a cluster (i.e. up to 256 nodes/cluster).

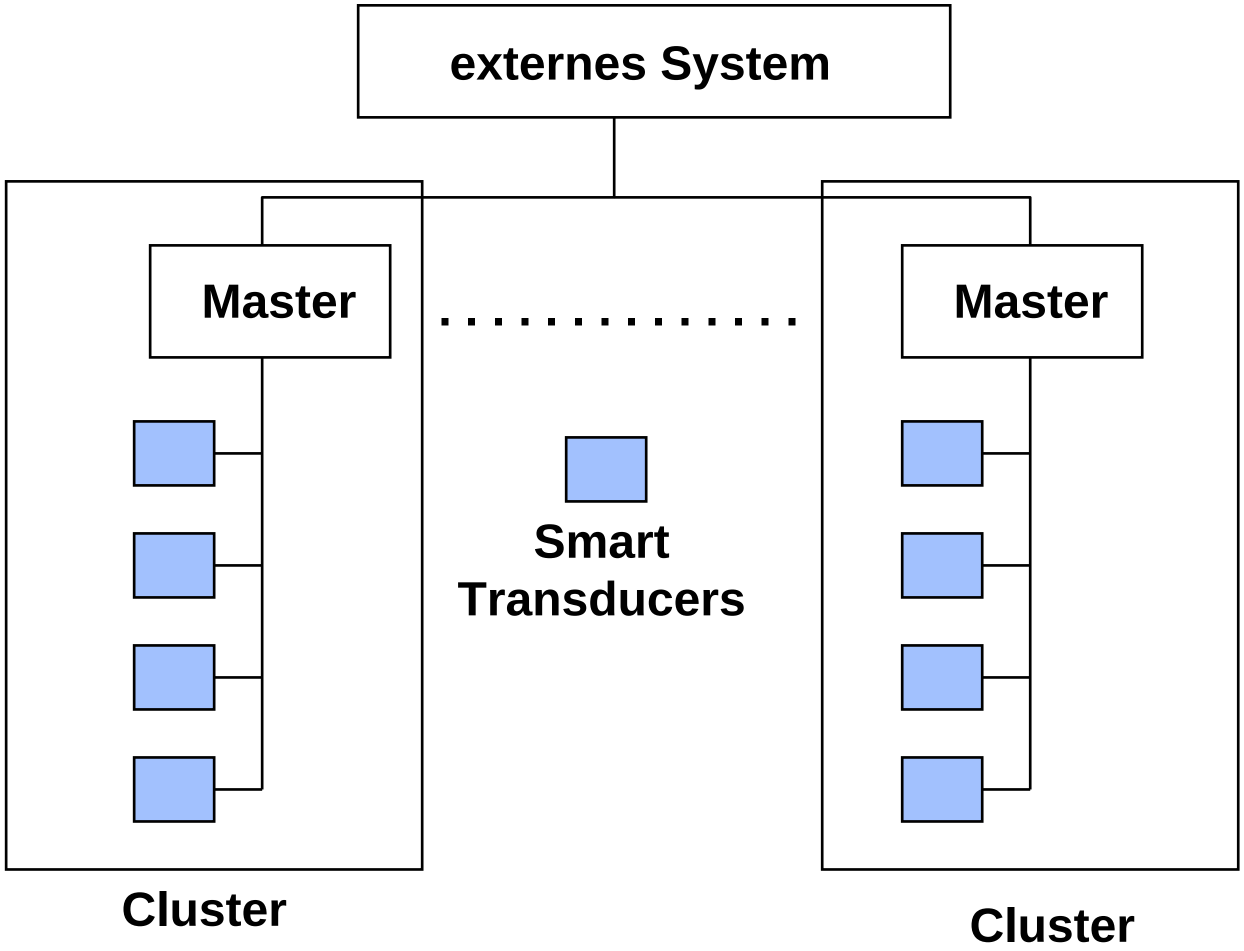
The assignment of a logical name to a node is called baptizing and can be performed on-line. Low cost nodes can have preprogrammed logical names.

During operation a node is addressed by:

<Cluster Name, Node Name, File Name, Record Name>

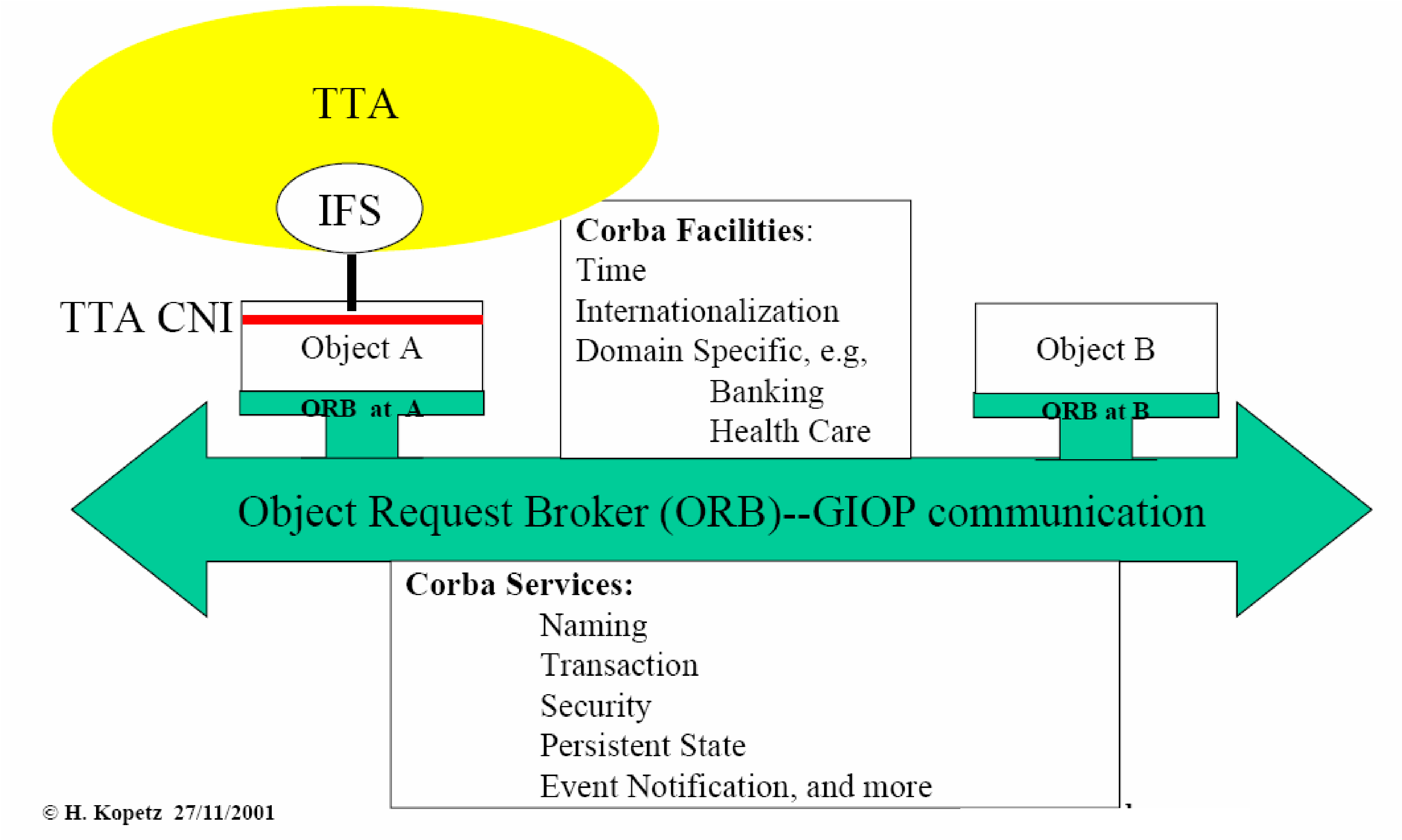
General architecture of a TTP/A system

global name of a data item: **<cluster name, node name, file name, record name>**



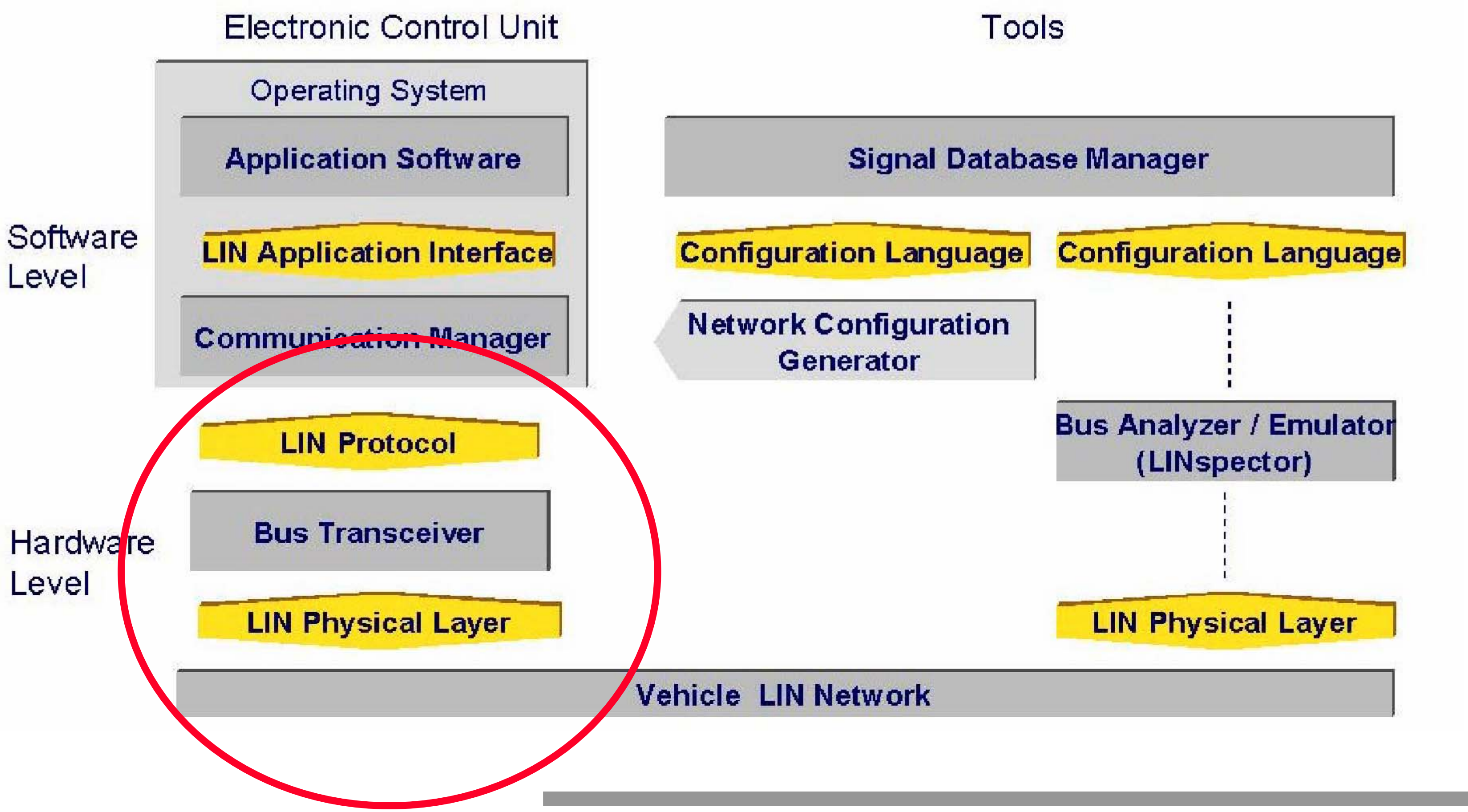
http://www.omg.org/docs/formal/03-01-01.pdf

Integrating a TTP/A system in CORBA



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.

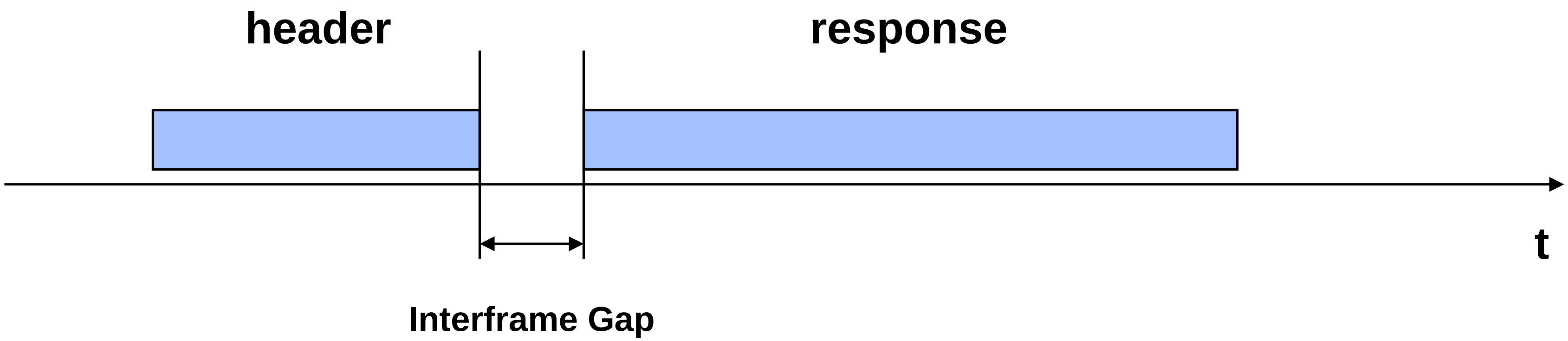
. self synchronization without quartz or ceramics resonator in the slave nodes

. guarantee of latency times for signal transmission

. low cost single-wire implementation

. speed up to 20kbit/s.

Master-Slave communication in LIN

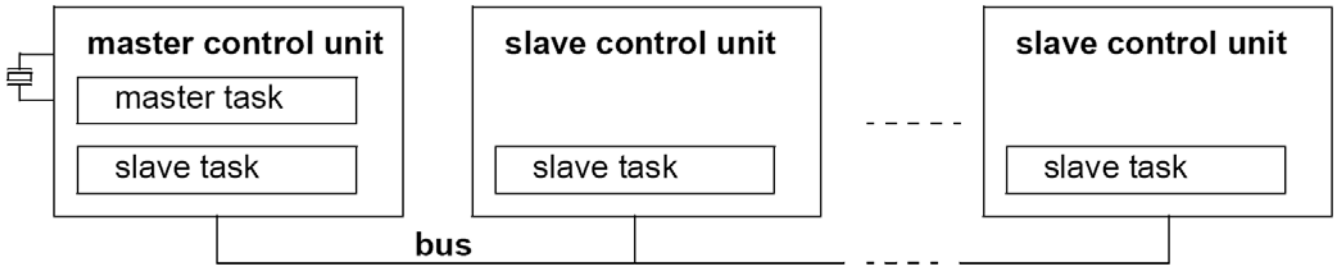


Header:
- serves for the synchronisation of slaves
- specifies the sequence and length of the fields in the data frame

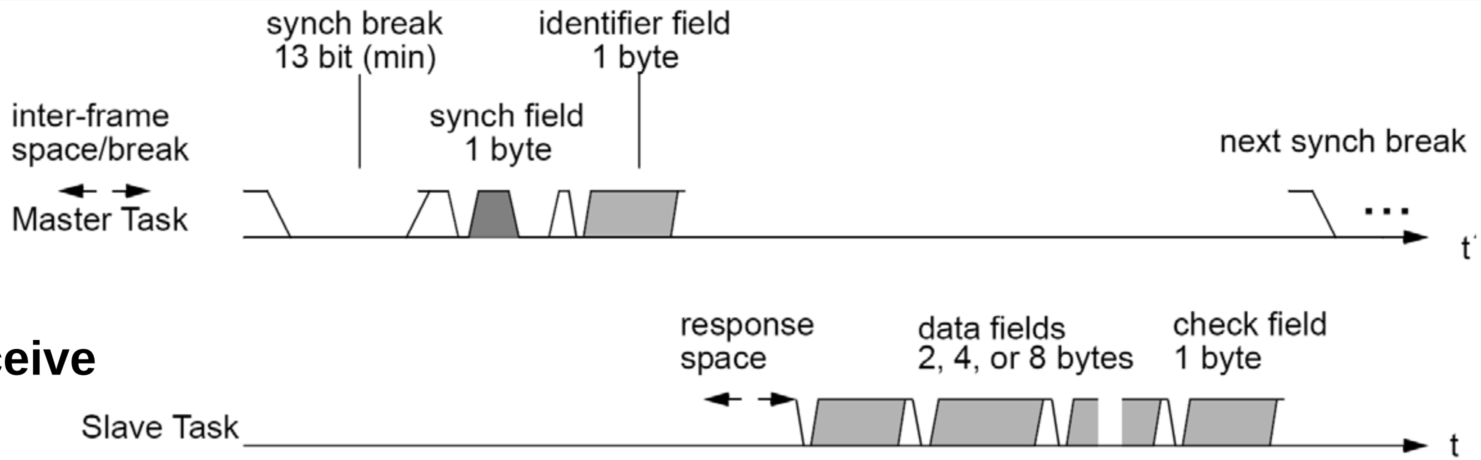
LIN (Local Interconnect Network)

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

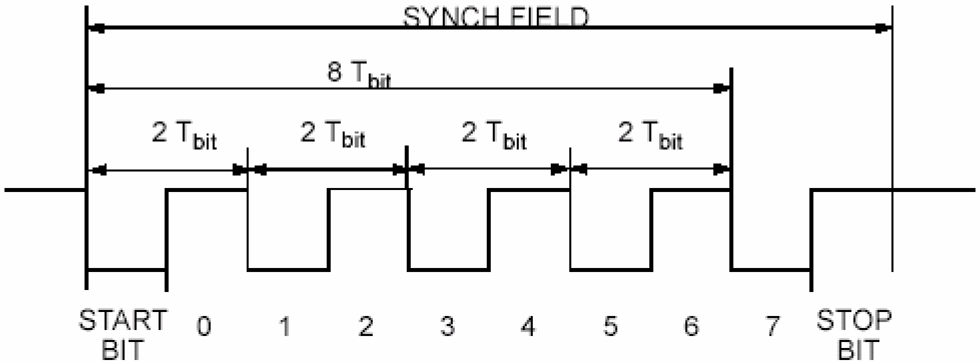


Figure 9.1: SYNCHRONIZATION FIELD

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

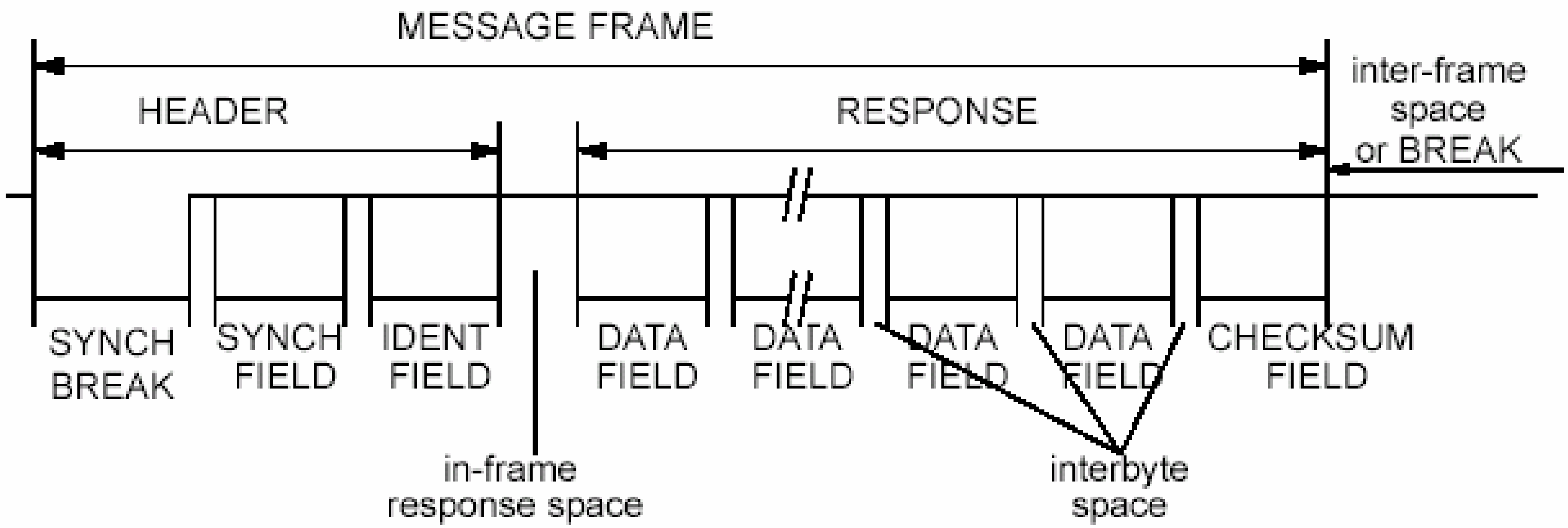


Figure 3.1: LIN MESSAGE FRAME

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

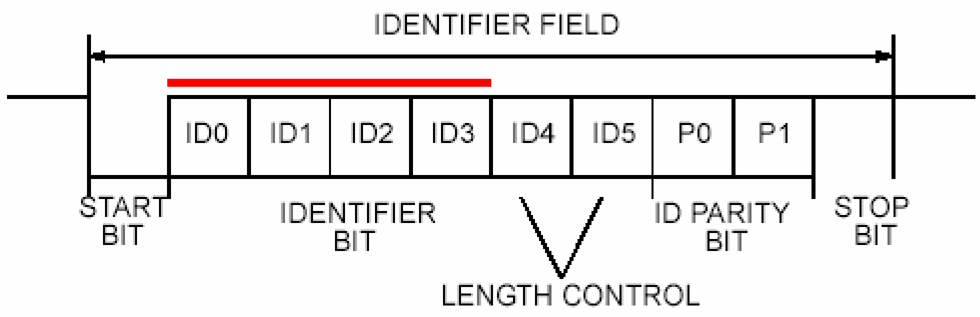


Figure 3.5: IDENTIFIER FIELD

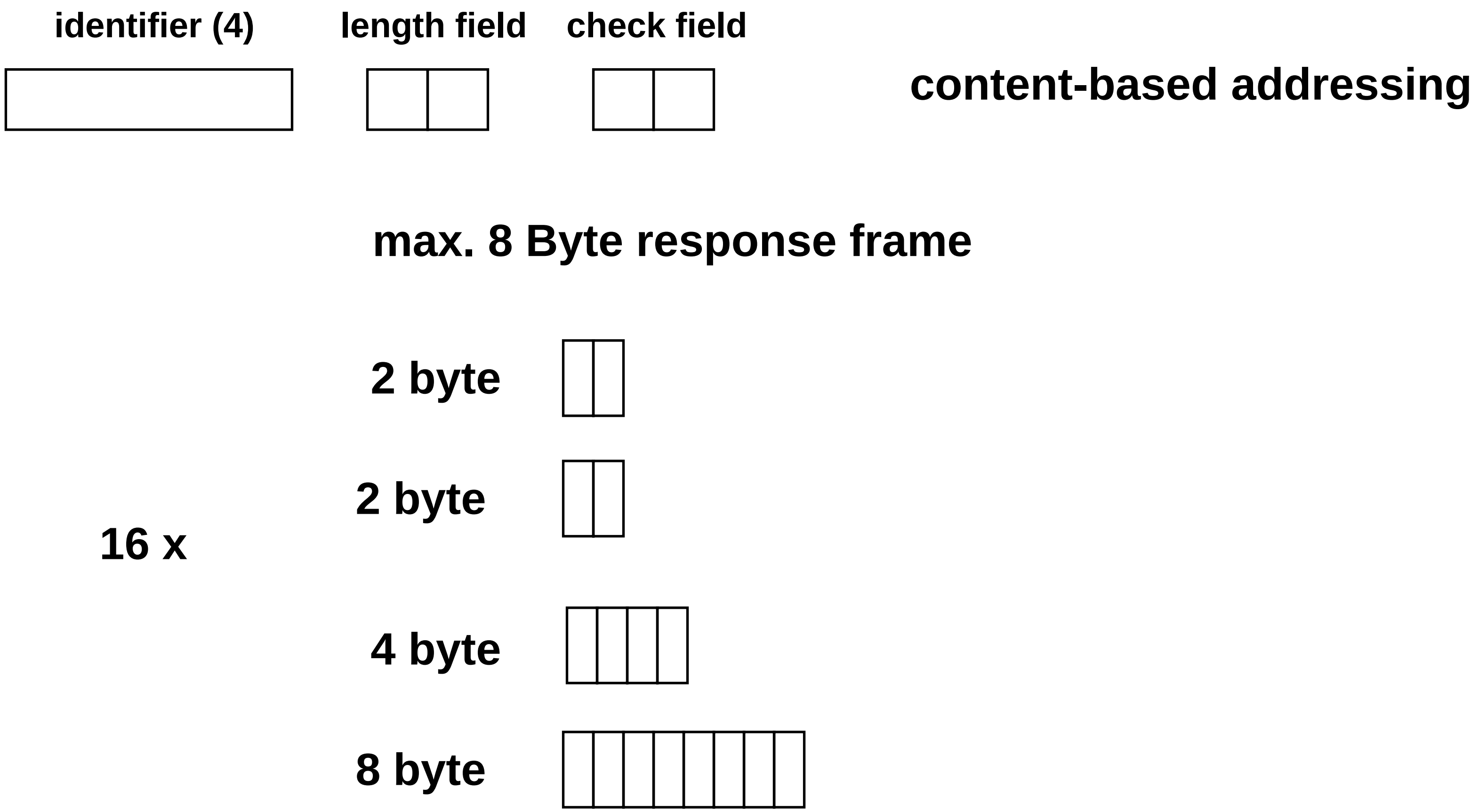
64 identifiers

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

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

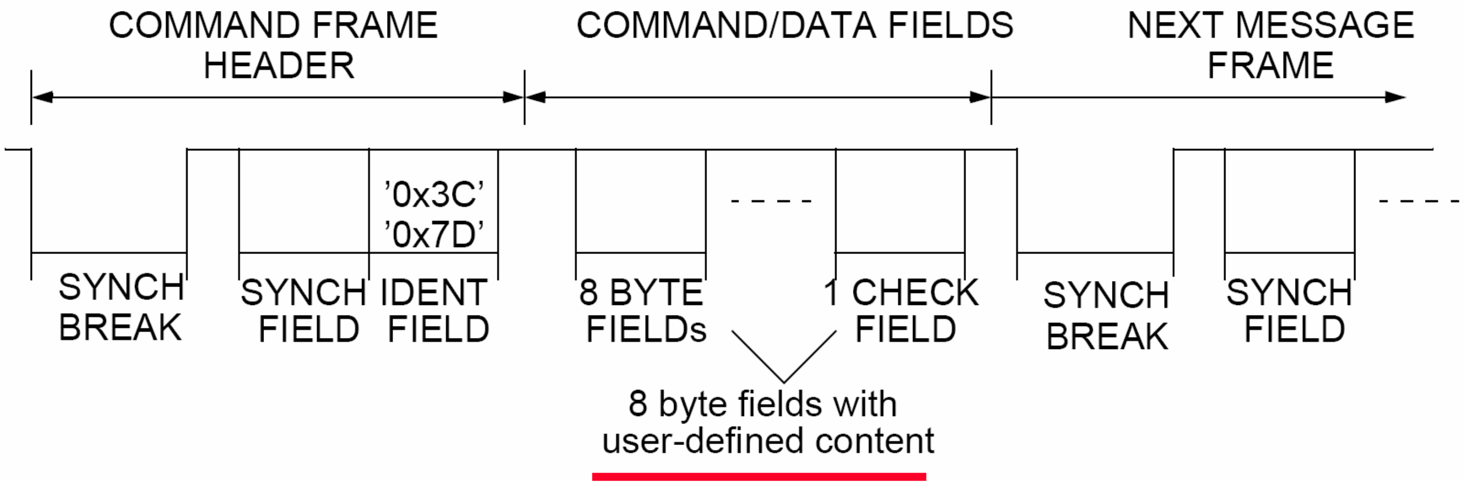
Slaves can be added or removed without changing any software in the other slaves.

LIN frame format



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

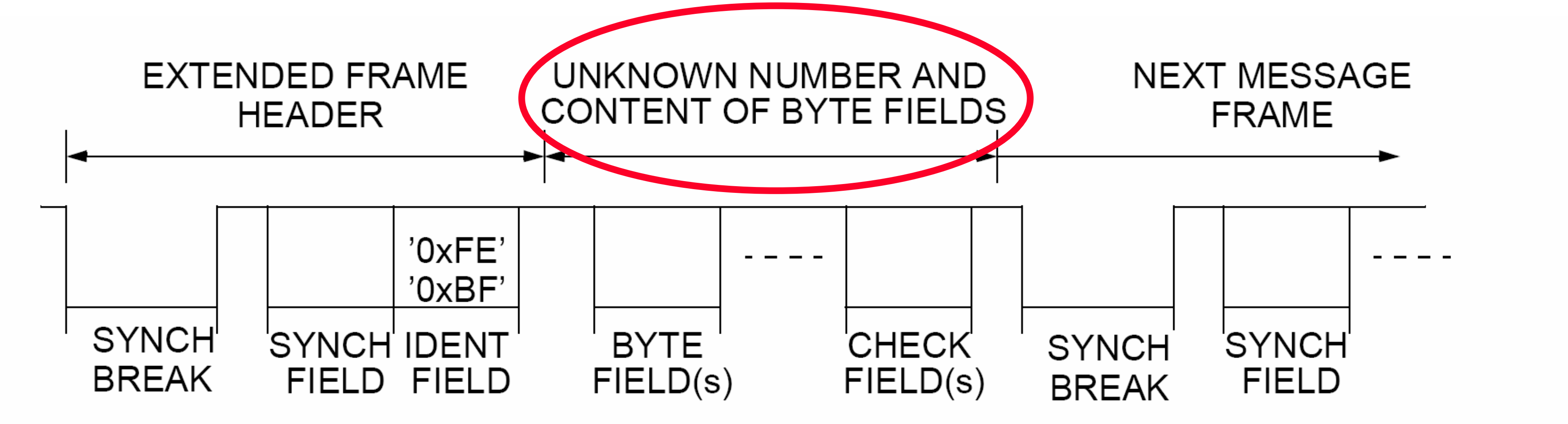
LIN Master Request Frame



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

Comparison LIN und TTP/A (response time and protocol efficiency)

Kopetz, Elmenreich, Mack, TU Wien, 2000

response time

10 nodes, response time in milliseconds on a 20 kbit bus	Minimum LIN	Maximum LIN	Minimum TTP/A	Maximum TTP/A
Every nodes sends four bytes of data	46.75 msec	65.4 msec	35.4 msec	35.6 msec
Every nodes sends two bytes of data	35.75 msec	50.05 msec	22.2 msec	22.3 msec
Every node sends one byte of data	35.75 msec	50.05 msec	15.6 msec	15.7 msec
Every node sends four bits of data	35.75 msec	50.05 msec	9 msec	9.1 msec
Every node sends four bits of data, additional master-slave round for DM service between any two multipartner rounds in TTP/A	not supported	not supported	16.8 msec	16.9 msec

Table 2: Achievable response times of LIN and TTP/A

protocol overhead

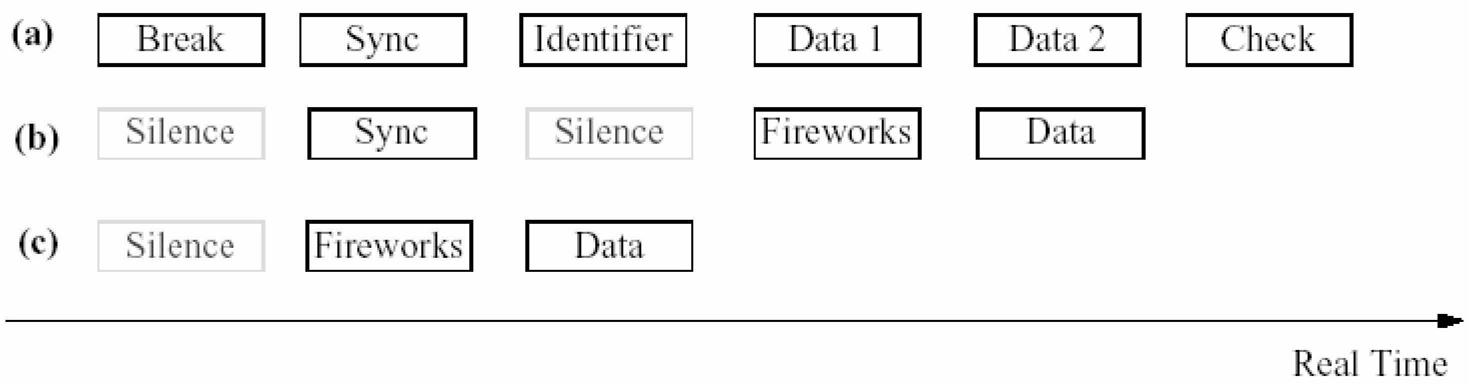


Figure 5: Byte Sequence of the simplest message in LIN (a), in TTP/A with start-up synchronization (b) and in TTP/A without start-up synchronization (c).

Automotive and highly dependable Networks

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