

LIN protocol description



LOCAL INTERCONNECT NETWORK

31 May 2011
TSPG – 8/16 bit Products Division

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2001.

LIN Consortium

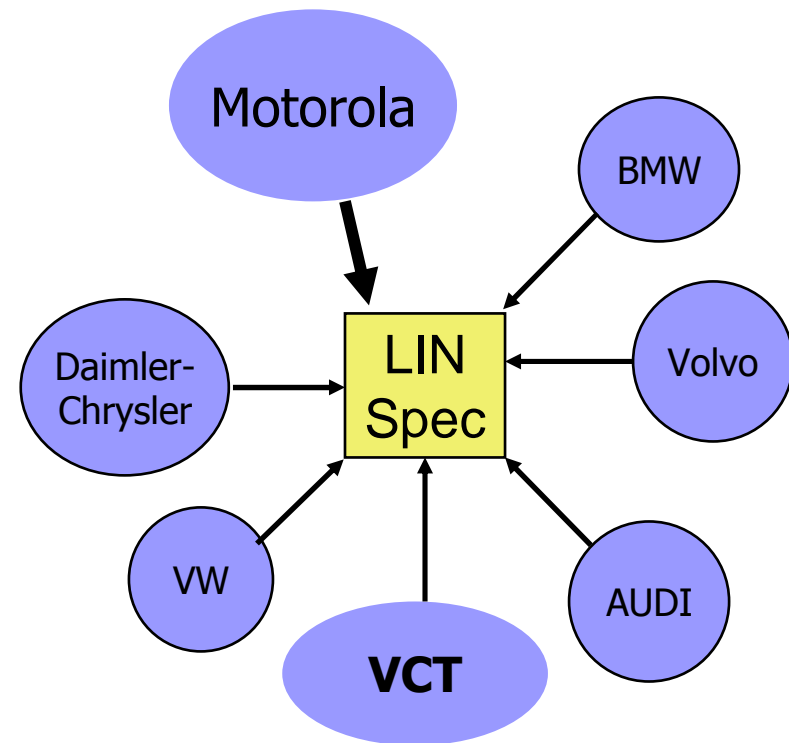
Consortium formed in 1998.

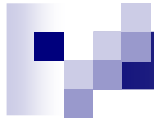
Five Car manufacturers

ONE Semiconductor Supplier (Motorola)

One tool Supplier (VCT)

LIN targets to low end applications
where the communication cost
per node must be two to three times
lower compared to CAN but
where the performance,
bandwidth, and versatility of
CAN is not required

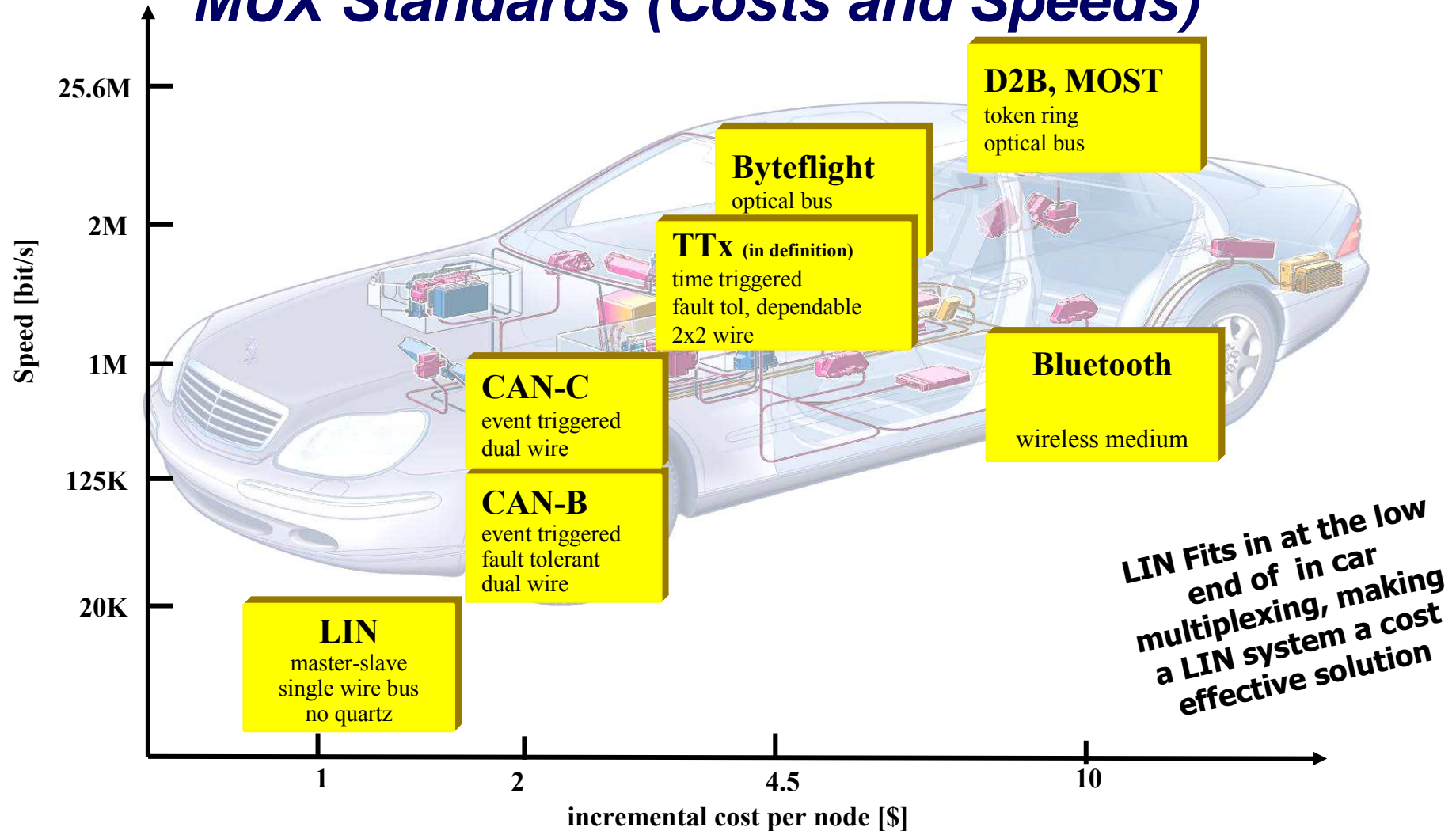




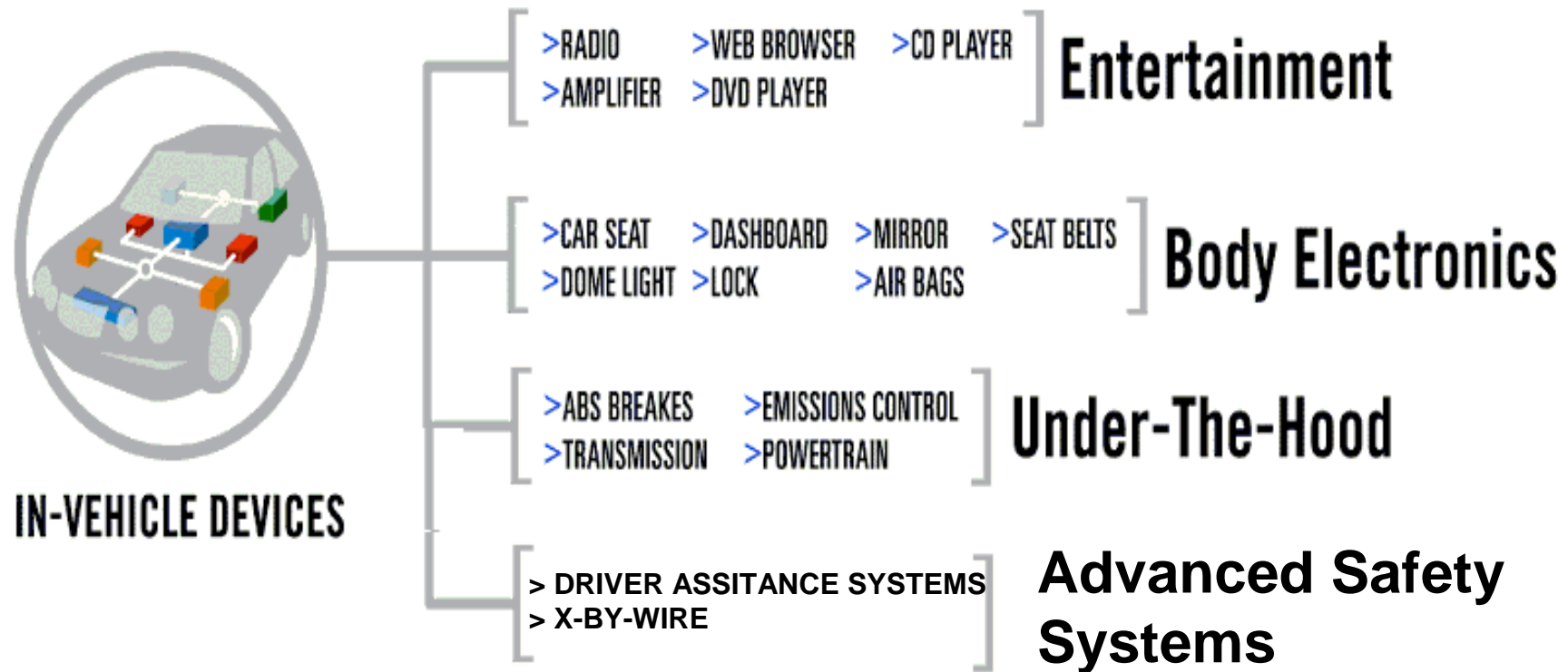
The Need for LIN

- (a) Multimedia applications, calling for protocols providing high speed, high bandwidth, and even wireless interconnection, like MOST, D2B, or Bluetooth;
- (b) Emerging safety critical applications in chassis and power train (x-by wire) calling for a fault tolerant, dependable protocol, like TTP/C, byteflight, TT-CAN or others;
- (c) Conventional body and power train applications, mainly using CAN;
- (d) Mechatronic type applications such as smart sensors and actuator, or even complex ECUs with simple communications needs, being addressed by low-end protocols like LIN, TTP/A, J-1850, and quite a few other OEM or Tier-I in-house protocols.

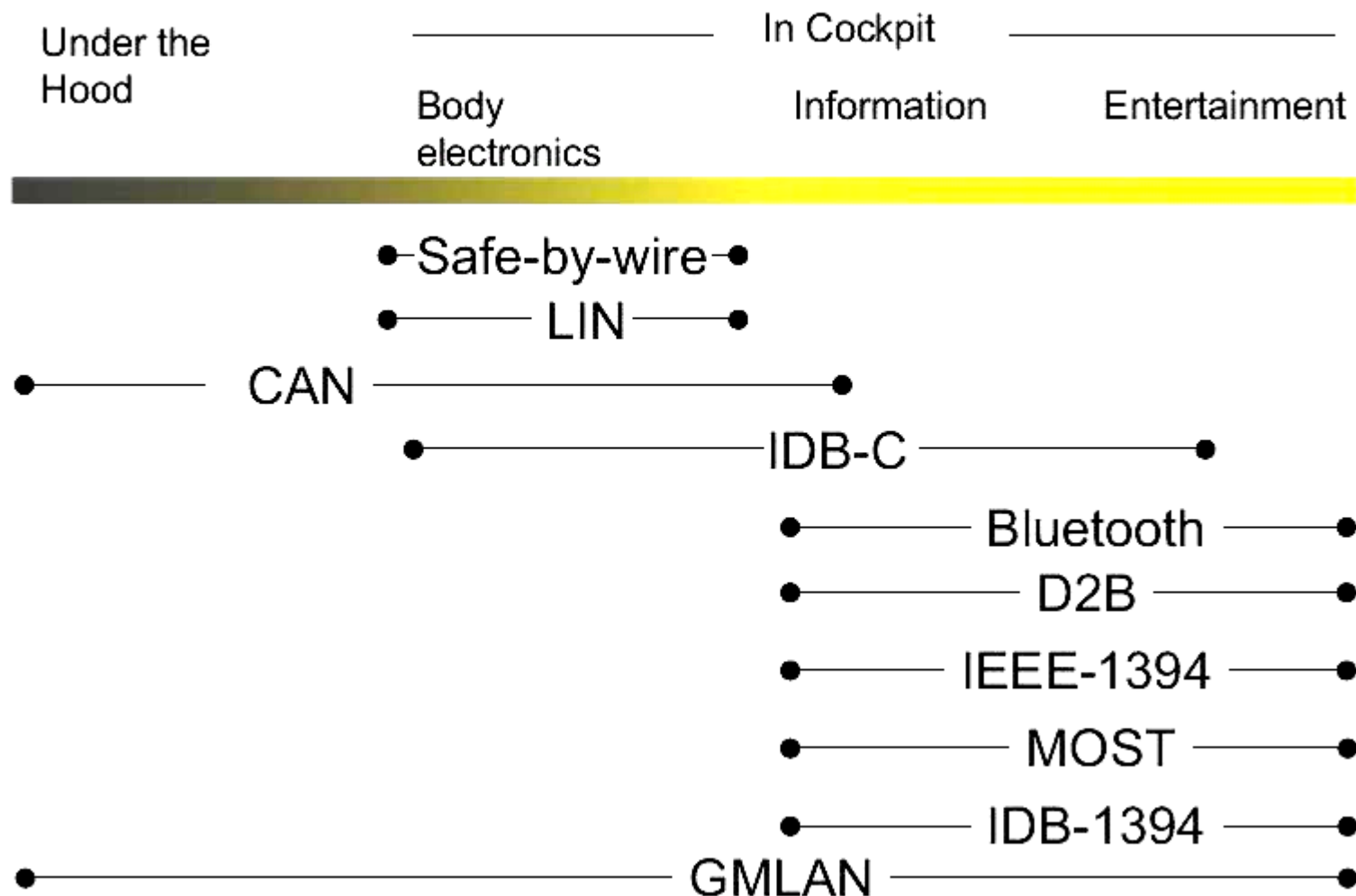
MUX Standards (Costs and Speeds)



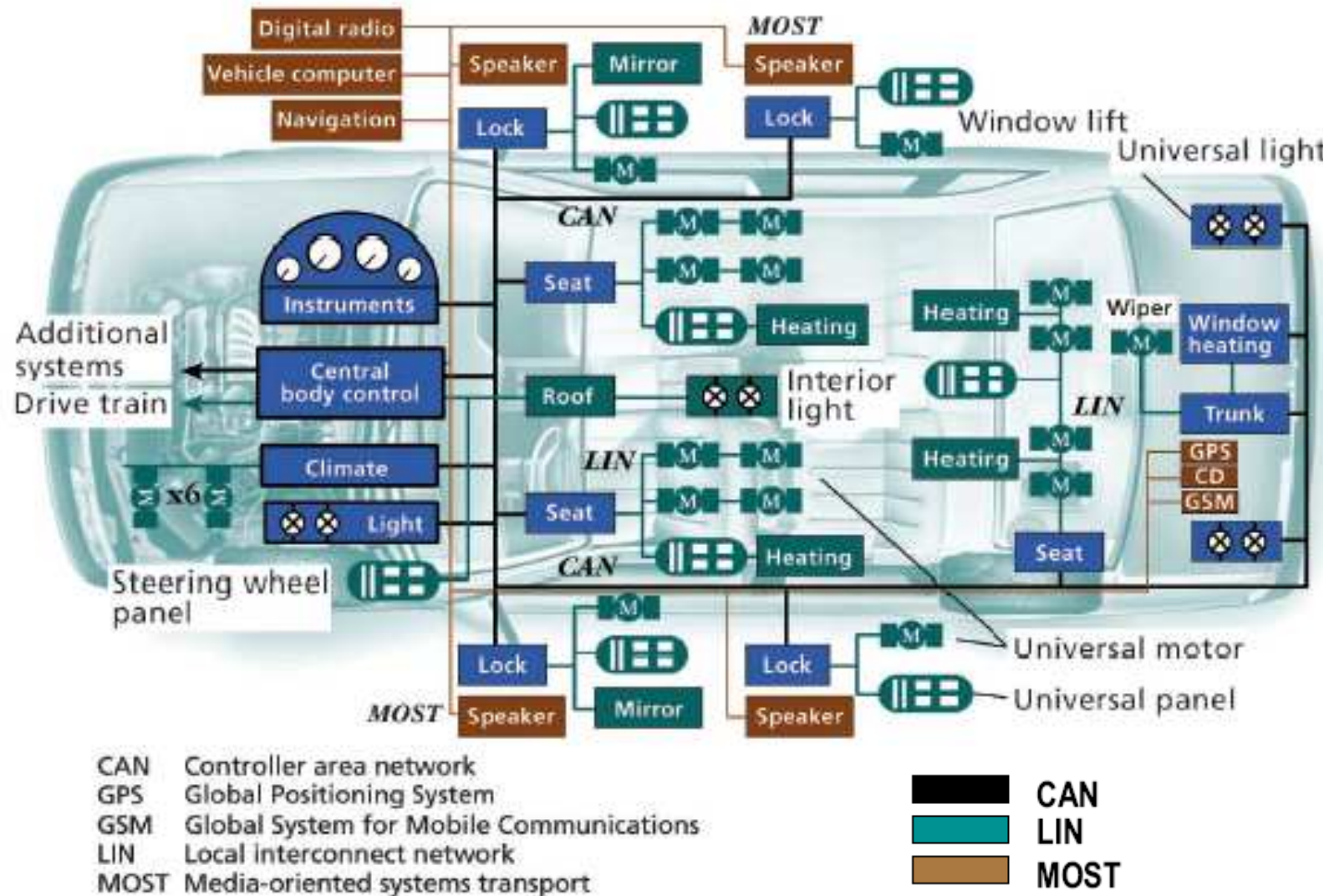
In-Car Networks



Applications for In-Vehicle Networking



Automotive Body Network



Typical LIN Applications

Roof:

(high amount of wiring)

Rain Sensor, Light Sensor,
Light Control, Sun Roof

...

(Rain Sensor needs to be
interrogated every 10-20ms)

Steering Wheel:

(very many controls are going to be
positioned on the steering wheel)

Cruise Control, Wiper,
Turning Light, ...

Optional: Climate Control,
Radio, Telephone, etc.

Seat:

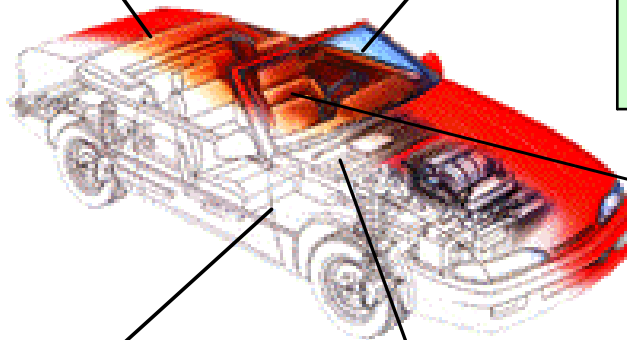
many Seat Position Motors,
Occupancy Sensor,
Control Panel

•Door/window/seat:

Mirror, Central ECU,
Mirror, Switch, Window
Lift,
Seat Control Switch,
Door Lock, etc.

Climate:

many Small Motors
Control Panel





Application For LIN (1)

Typical areas for the LIN bus are:

- Assembly units (door, roof, seat, climate, ...)
- New mechatronic elements

(smart sensors or actuators can be easily connected to the car network and become accessible to all types of diagnostics and services)

- Reduction of wiring harness

(replacement of single wired and analog coded switches)



Application For LIN (2)

- door control (window lifter, seat control, mirror)
 - roof control (light/rain sensor)
 - steering control (switches)
 - seat control (motors)
 - climate control (motors, sensors)
 - intelligent wiper motor
 - RF-receiver for remote control
 - intelligent alternator
 - switch panels
- 



Body Control and Under-the-hood

- LIN (Local Interconnect Network)
 - Event Triggered
 - Low cost body electronics network based on UART SCI data format
 - Single master to multiple slave concept, single wire 12V bus
 - Transmission speed: **20kbit/s**
 - Serial bus used for distributed electronic systems in vehicles
 - Used for smart sensors and actuators where the bandwidth and versatility of CAN is not needed
 - Conceived in 1998
 - Consortium members include: Audi, BMW, Motorola, DaimlerChrysler, VW and Volvo (plus tool maker ICT and Motorola semiconductor)
 - Open source i.e no license
 - Typical applications: smart sensors, actuators or illumination in vehicles e.g. door control, steering wheel, seats, climate regulation, lighting and rain sensors
 - Also used in machine control as a sub-bus for CAN in industrial applications



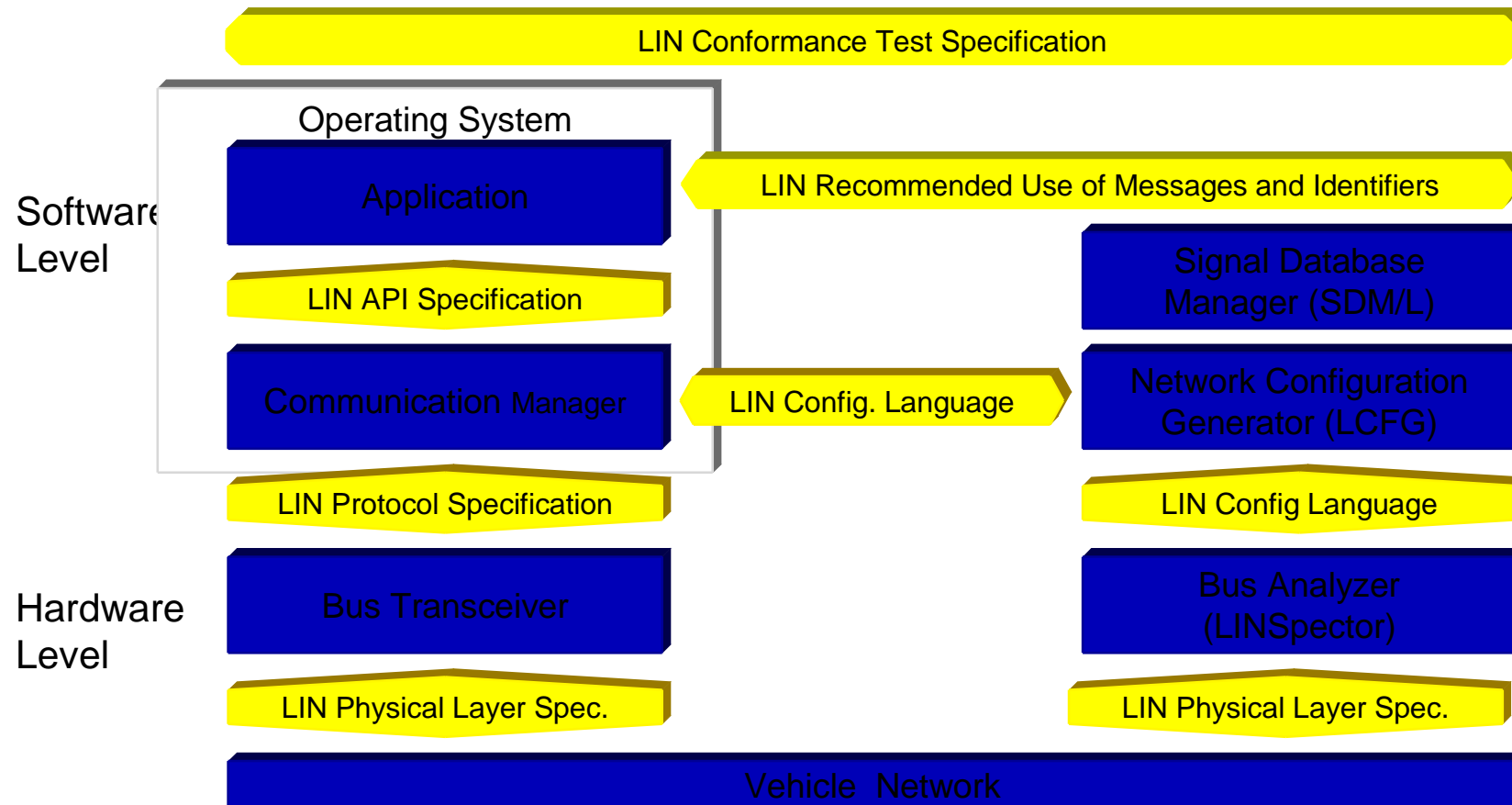
General Objectives

- Definition and implementation of an open standard for class-A serial buses in vehicles
- Enabling factor for the implementation of a hierarchical vehicle network to achieve further quality enhancement and cost reduction
- Reduction of the manifold of existing low-end multiplex solutions and cutting the cost of development, production, service, and logistics in vehicle electronics.

LIN Standard - Overview

ECU
(LIN relevant functions only)

Tools

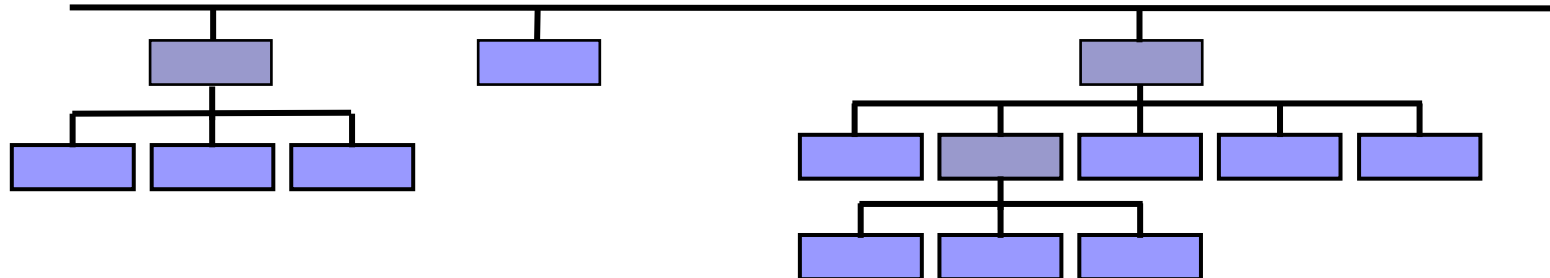


Hierarchical Network Structure

Flat Network



Hierarchical Network



- Subnets are necessary to reduce Busload on main Bus
- Solution examples:

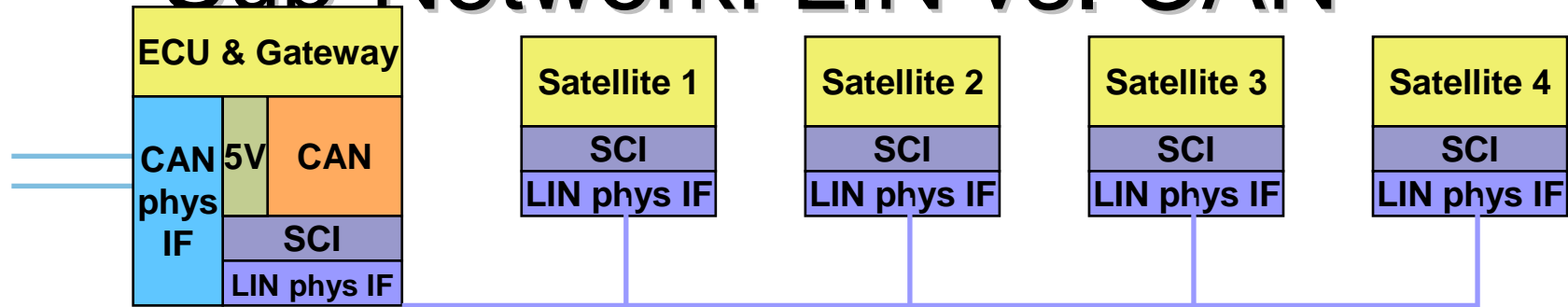
- **CAN**

- + Automotive Standard Bus
- + Compatible with Main Bus
- Expensive (Die Size/ Dual Wire)

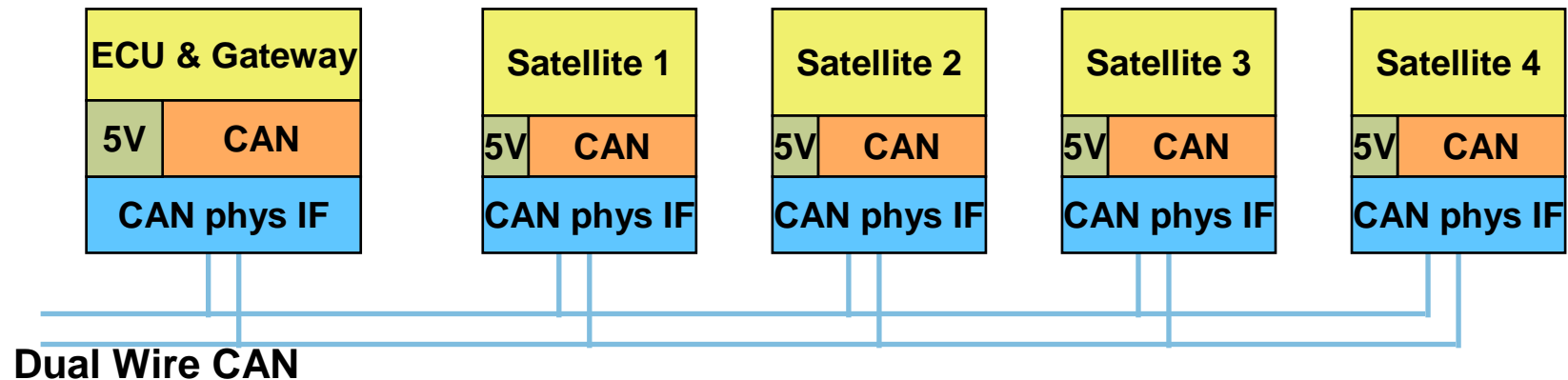
- **Serial Sub Bus**

- no standard Bus System
- not compatible with Main Bus
- + inexpensive
- + SCI-Based: Interface exists even on cheap devices
- + Interface can easily be reconstructed by ASIC or CPLD
- + Protocol can be done in Software

Sub-Network: LIN vs. CAN



LIN




Dual Wire CAN

Cost Factors:

CAN Module
Crystal
2nd Wire / Connector

Dual Wire Interface
5V supply for bus



Comparison of the main features of LIN and CAN protocol in body applications

	LIN	CAN
medium access control	single master	multiple master
typical bus speed	2.4 ... 19.6 kbps	62.5 ... 500 kbps
multicast message routing	6-bit identifier	11 / 29-bit identifier
typical size of network	2 ... 10 nodes	4 ... 20 nodes
bit / byte coding	NRZ 8N1 (UART)	NRZ w/ bit stuffing
data byte per frame	2, 4, 8 byte	0 ... 8 byte
transmission time for 4 data bytes	3.5 ms at 20 kbps	0.8 ms at 125 kbps
error detection (data field)	8-bit checksum	15-bit CRC
physical layer	single wire, 13.5V	twisted pair, 5V
quartz/ceramic resonator	no (except master)	yes
relative cost per network connection	x 0.5	x 1



Typical memory and CPU requirements for LIN and CAN microcontroller

	network speed	CPU clock	CPU load	memory flash/ROM	memory RAM
	[kbps]	[MHz]	[%]	[byte]	[byte]
LIN 16-bit master	19.2	4	10	1200	25
LIN 8-bit slave w/o quartz	19.2	4	15	750	22
LIN 8-bit slave with quartz	19.2	4	6	650	20
CAN 16-bit node	125	8	15	3000	150

LIN shall complement CAN and not replace CAN!



SubNets

- Necessary to reduce Busload on main Bus

- Solutions

- CAN

- + Automotive Standard Bus
 - + Compatible with Main Bus
 - Expensive (Die Size/ Dual Wire)

- Serial Sub Bus

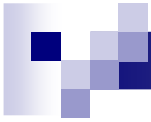
- no standard Bus System
 - not compatible with Main Bus
 - + inexpensive
 - + SCI-Based: Interface exists even on cheap devices
 - + Interface can easily be reconstructed by ASIC or CPLD



Sub Bus Concept

■ Basic Requirements:

- Satisfy Need for a Standard for Sub Busses
- Cost driven: The solution must be cheaper than CAN
- Reliability: Same Level as CAN expected
- Long Term Solution
- Logical Extension to CAN
- Scalable: Capability to extend Systems with additional nodes
- Lowering Cost of Satellite nodes:
 - No Crystal or Resonator
 - Easy implementation
 - Simple State Machines
- Low Reaction Time (100 ms max)
- Predictable Worst Case Timing



LIN Concept

- Technical Solution
 - Low cost single-wire implementation (enhanced ISO 9141)
 - Speed up to 20Kbit/s (limited for EMI-reasons)
 - Single Master / Multiple Slave Concept
 - No arbitration necessary
 - Low cost silicon implementation based on common UART/SCI interface hardware
 - Almost any Microcontroller has necessary hardware on chip
 - Self synchronization without crystal or ceramics resonator in the slave nodes
 - Significant cost reduction of hardware platform
 - Guaranteed latency times for signal transmission (Predictability)



Master / Slave Protocol

■ Master

- has control over the whole Bus and Protocol

The master controls which message at what time is to be transferred over the bus. It also does the error handling.

To accomplish this the master

- sends Sync Break
- sends Sync Byte
- sends ID-Field
- monitors Data Bytes and Check Byte, and evaluates them on consistence
- receives WakeUp Break from slave nodes when the bus is inactive and they request some action.
- serves as a reference with it's clock base (stable clock necessary)



Master/Slave Protocol

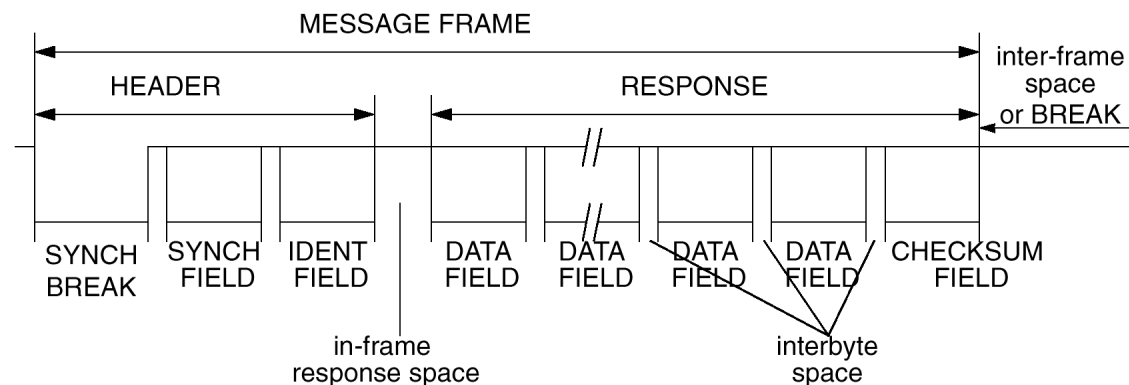
■ Slave

- Is one of 2-16 Members on the Bus and receives or transmits Data when an appropriate ID is sent by the master.
 - Slave snoops for ID.
 - According to ID, slave determines what to do.
 - either receive data
 - or transmit data
 - or do nothing.
 - When transmitting the slave
 - sends 1, 2, 4, or 8 Data Bytes
 - sends Check-Byte
 - The node serving as a master can be slave, too!

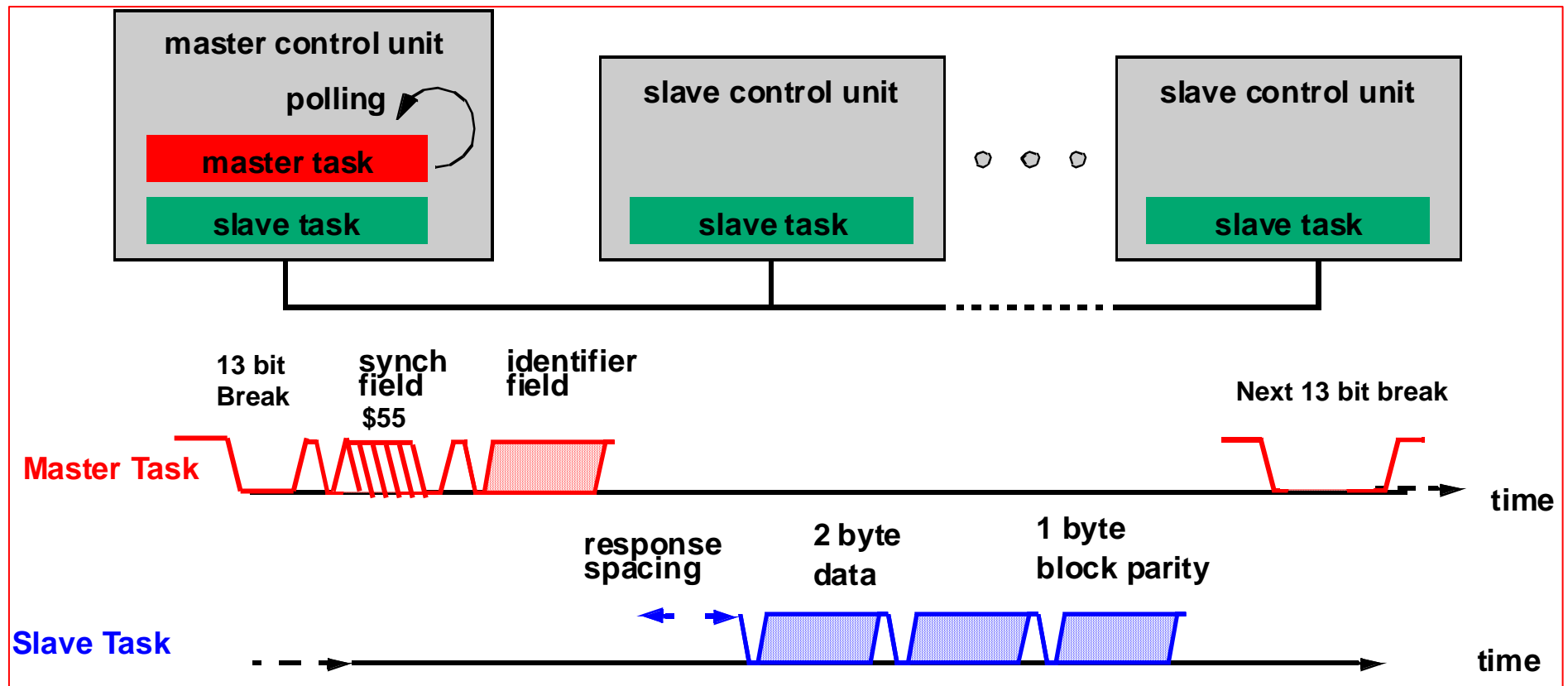
LIN protocol offers message timing predictability

Time Triggered Approach

- Message Length is known
 - Number of transmitted data bytes is known
 - ® minimum length can be calculated
 - Each Message has length budget of 140% of it's minimum length
 - ® maximum allowed length is known
 - ® distance between beginning of two messages



Data Transmission



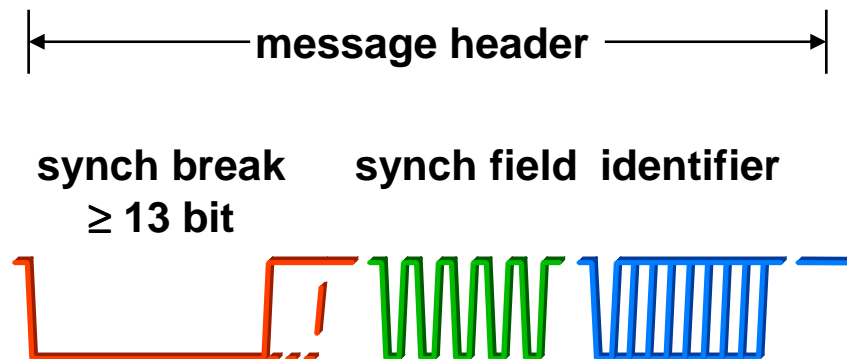


Message Frame

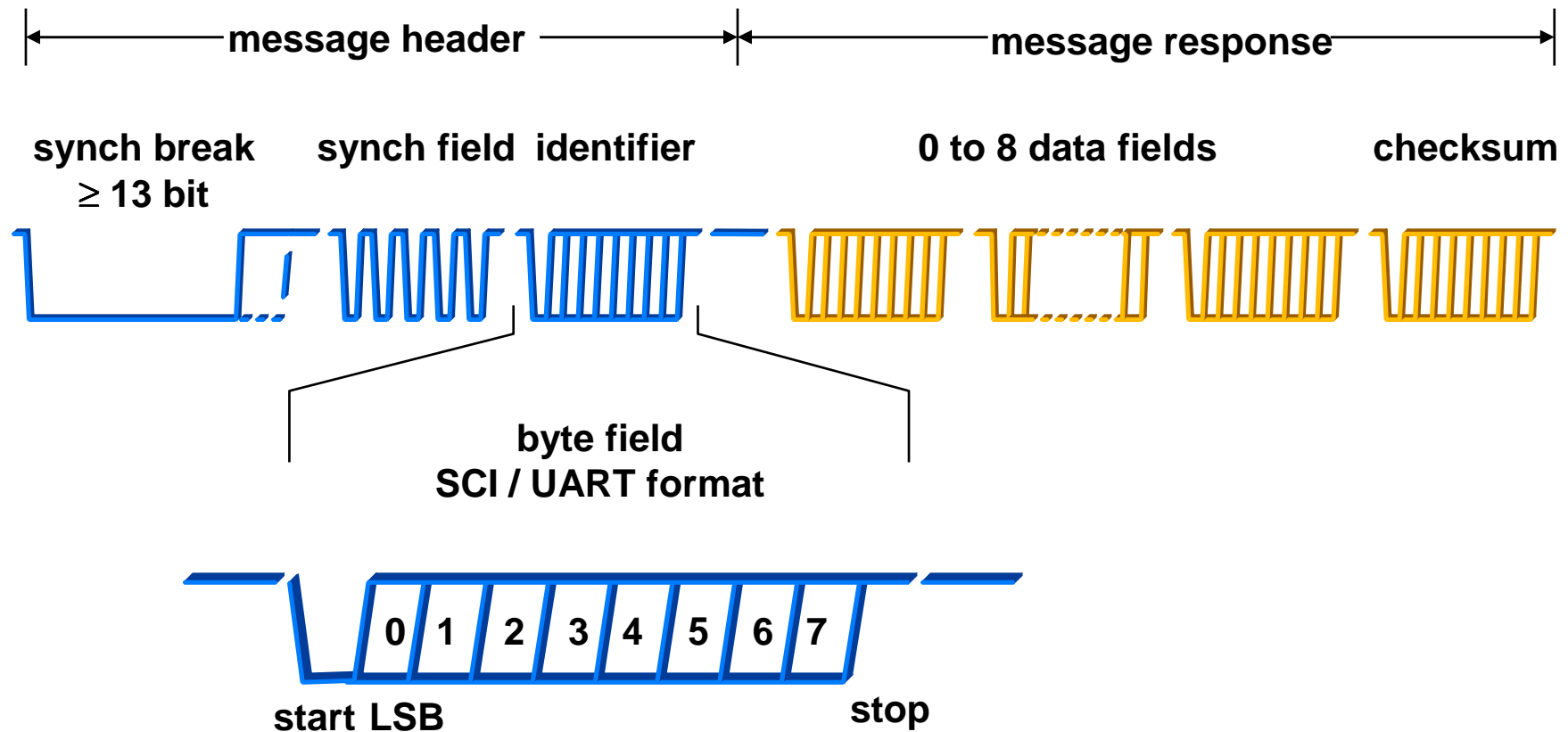
- Synch Byte:
 - Specific Pattern for Determination of Time Base
(Determination of the time between two rising edges)
 - A Synch Byte precedes any Message Frame
- ID-Field:
 - Message Identifier: Incorporates Information about the sender, the receiver(s), the purpose, and the Data field length.
Length 6 Bit.
4 classes of 1/2/4/8 Data Bytes. The length coding is in the 2 LSB of the ID-Field. Each class has 16 Identifiers. A total of 64 Message Identifiers are possible.
 - 2 Parity Bits protect this highly sensitive ID-Field.

Identifier

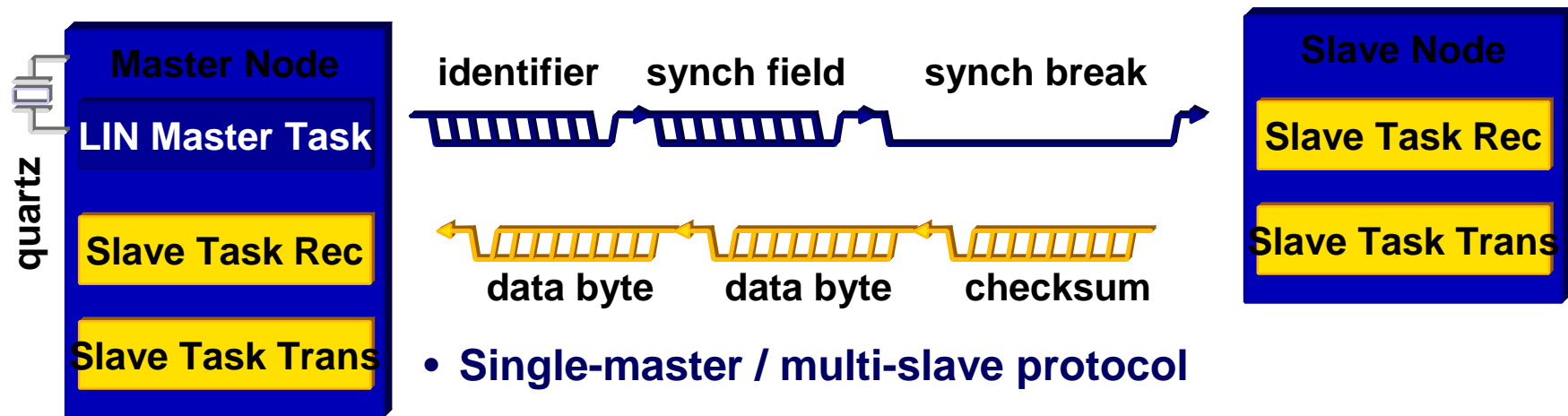
- The identifier field is sent by the master node to all LIN nodes
- This identifier normally contains one of 64 different values and includes 2 parity bits in the 8 bit data
- The identifier is normally associated with a collection of signals that are subsequently transmitted on the LIN bus
- In a specific case this can initiate SLEEP mode in the LIN slave nodes – in this case no further data is transmitted on the LIN bus



LIN Message Frame

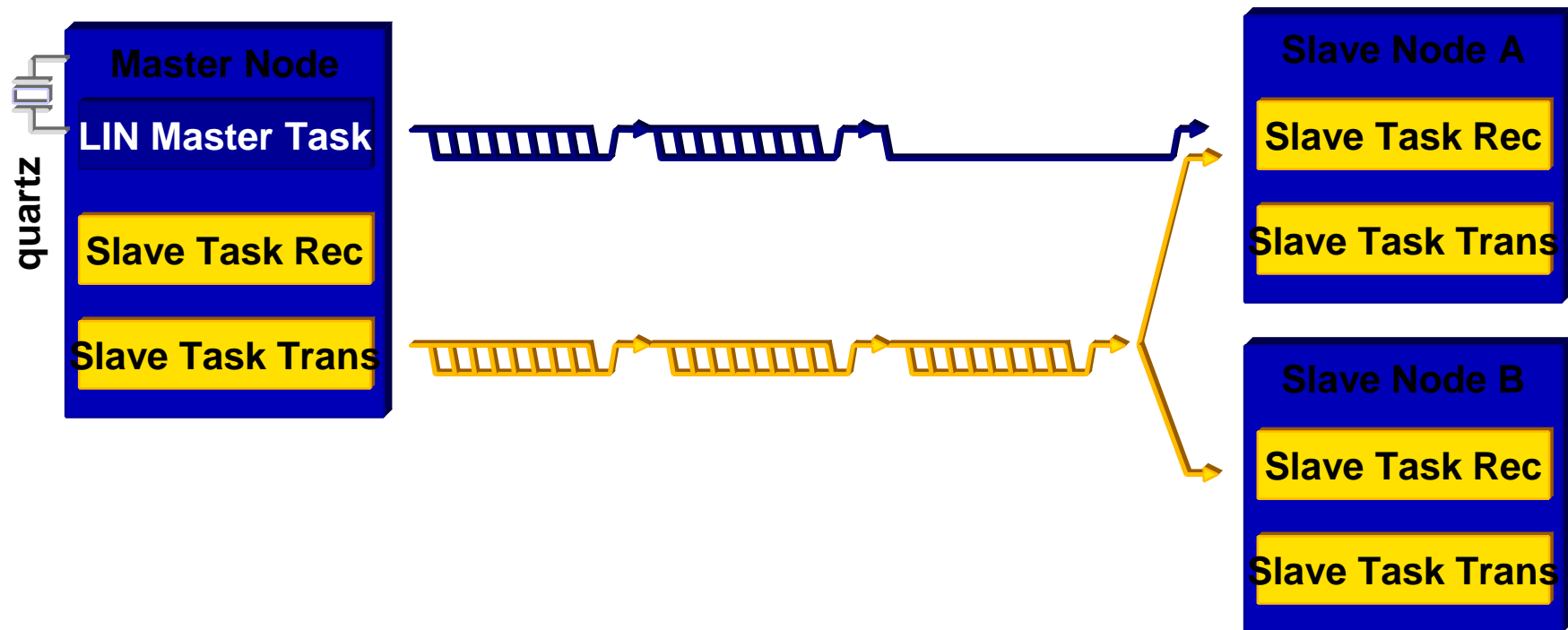


LIN Communication - Data from Slave to Master

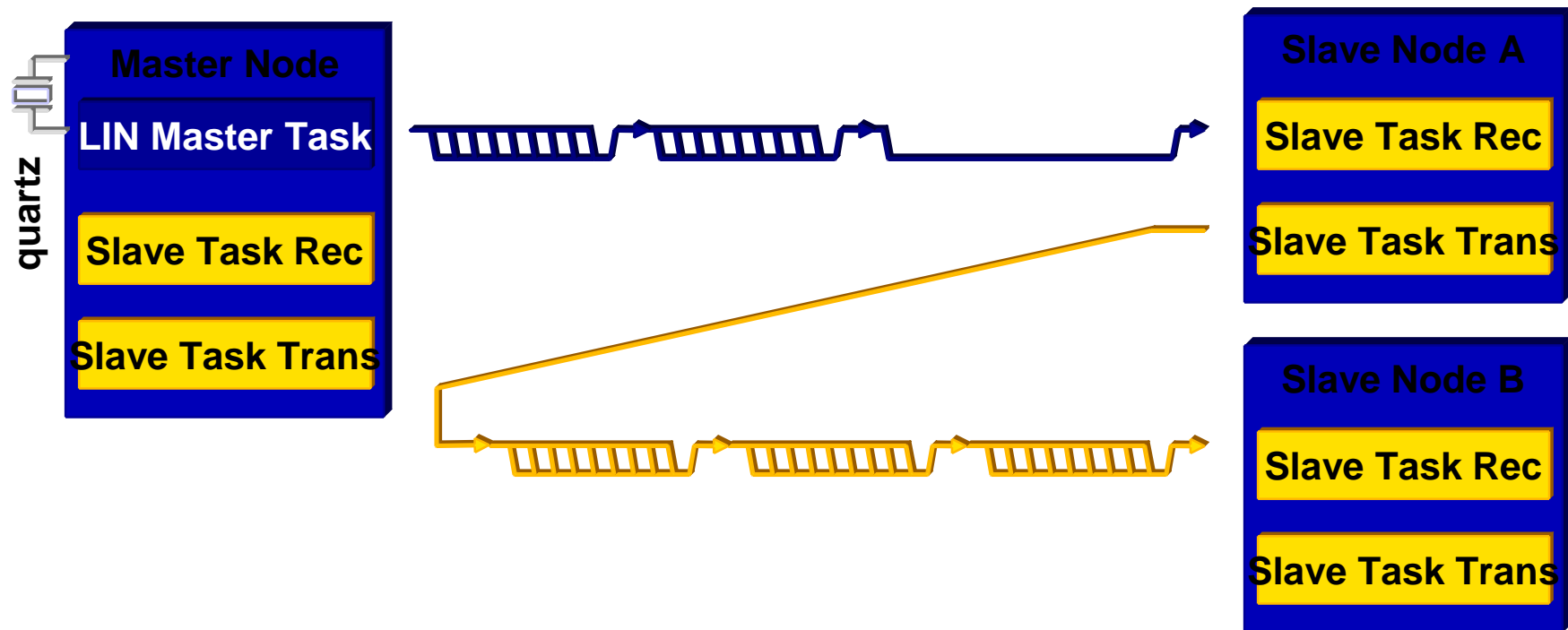


- Single-master / multi-slave protocol
- Time triggered, no arbitration
- Identifier denotes message content, not physical address
- Multicast messages
- Baud rate synchronization through protocol
- Power saving sleep mode

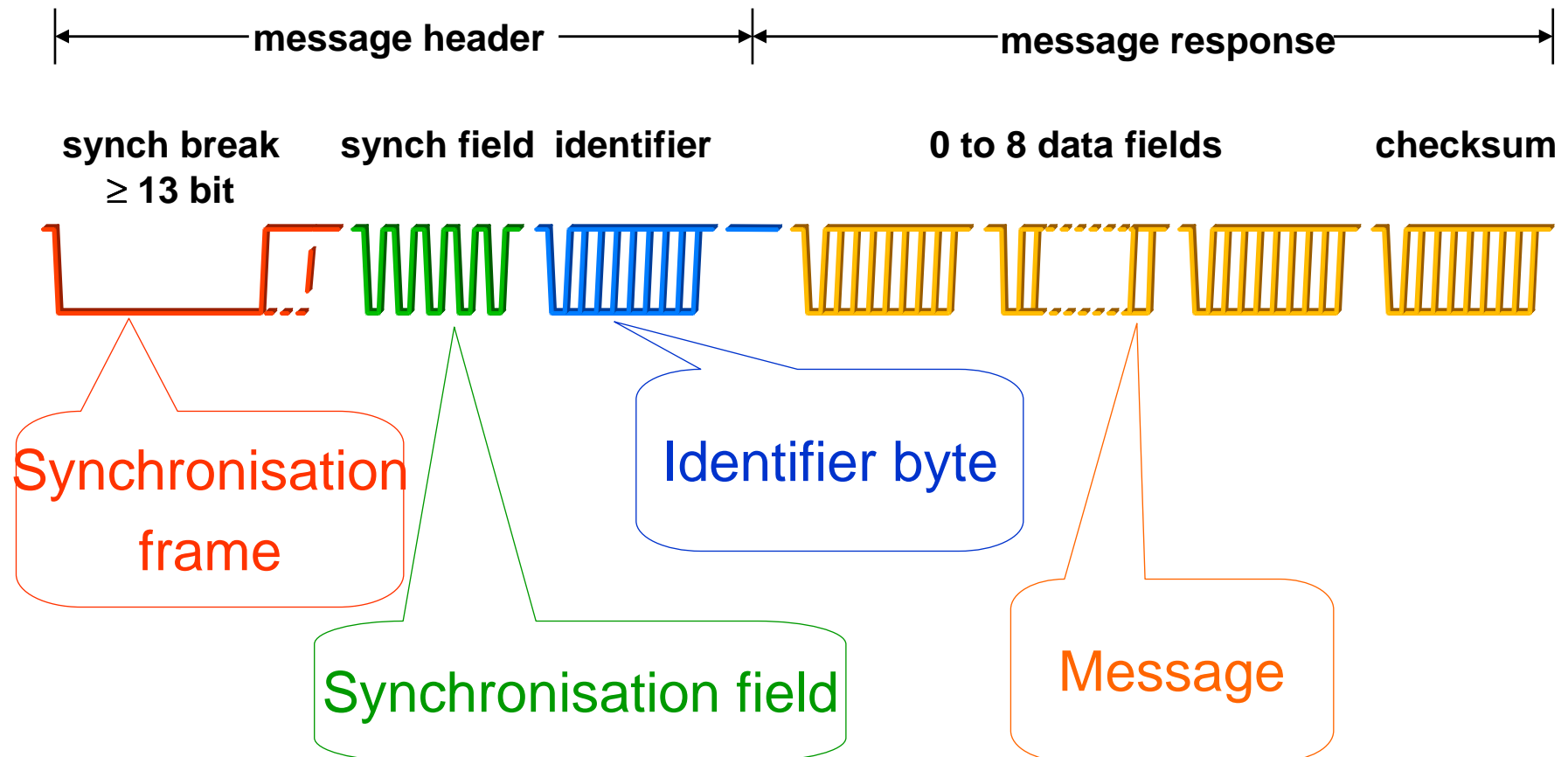
LIN Communication - Data from Master to Slave(s)



LIN Communication - Data from Slave to Slave



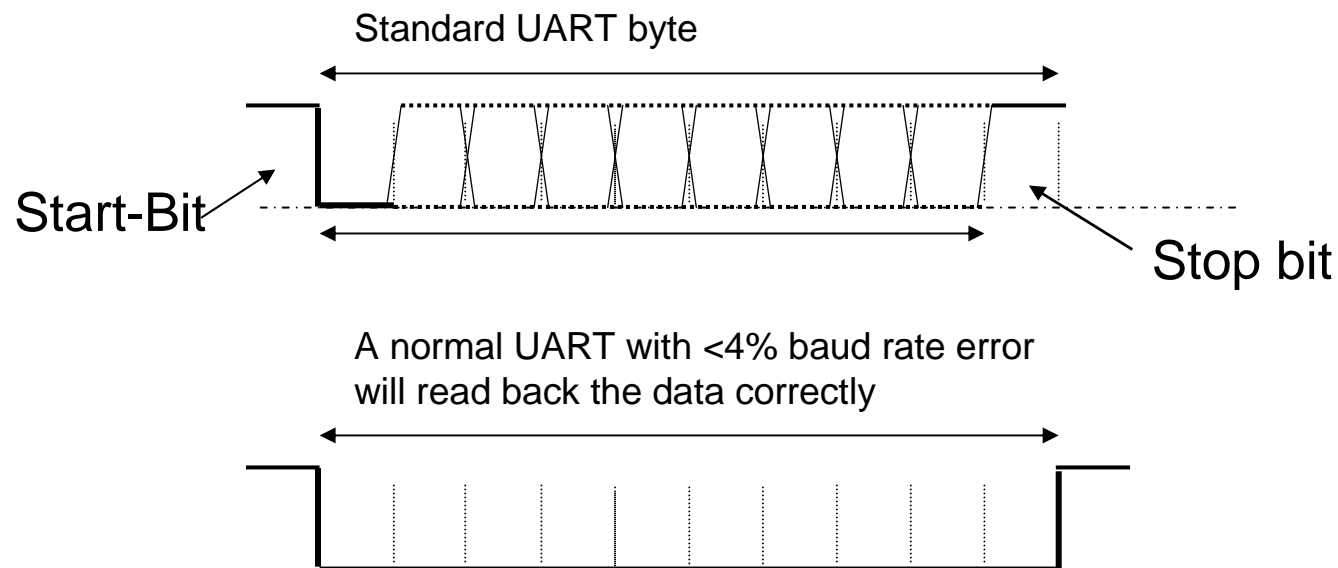
LIN Message Frame



Frame Synchronisation (1)

Initial conditions: $\pm 4\%$ baud rate accuracy relative the transmitting source

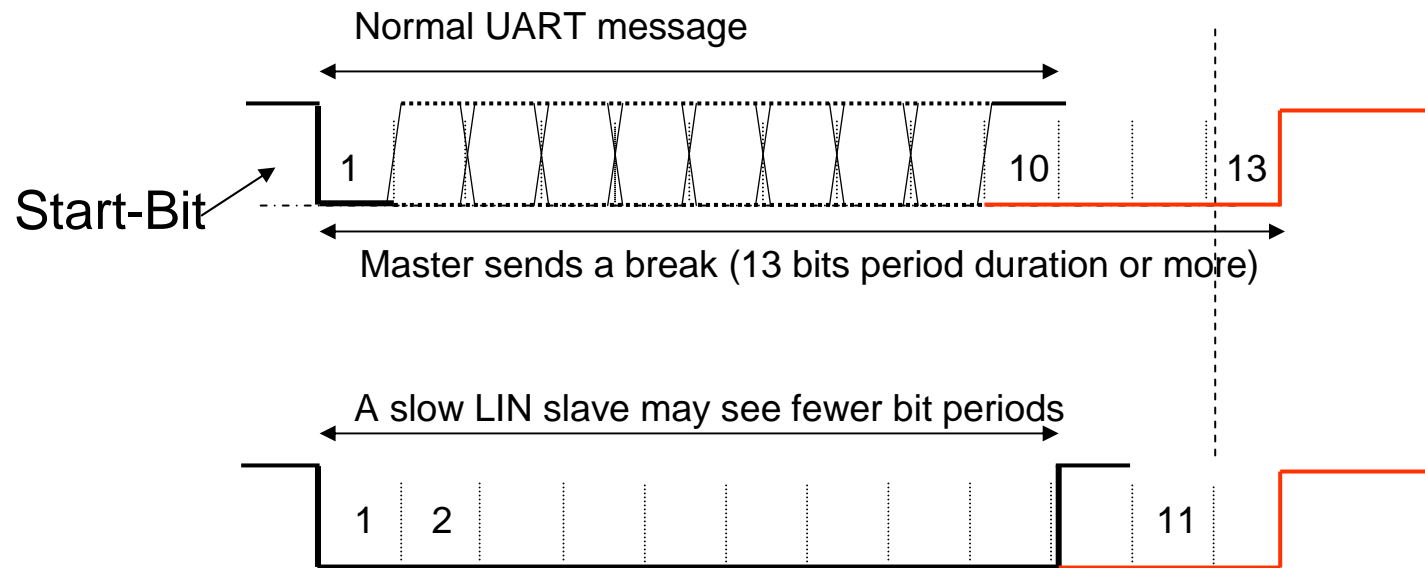
A standard transmission of data will require matched send and receiver baud rates



Frame Synchronisation (2)

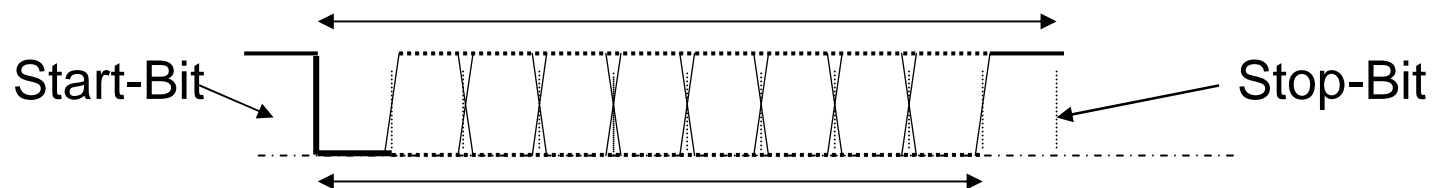
Initial conditions: +/- 15% baud rate accuracy relative the the LIN master transmitting the synchronisation frame

A synch break must be at least 13 bit periods in duration to allow for this initial variation in oscillator accuracy within the LIN slave

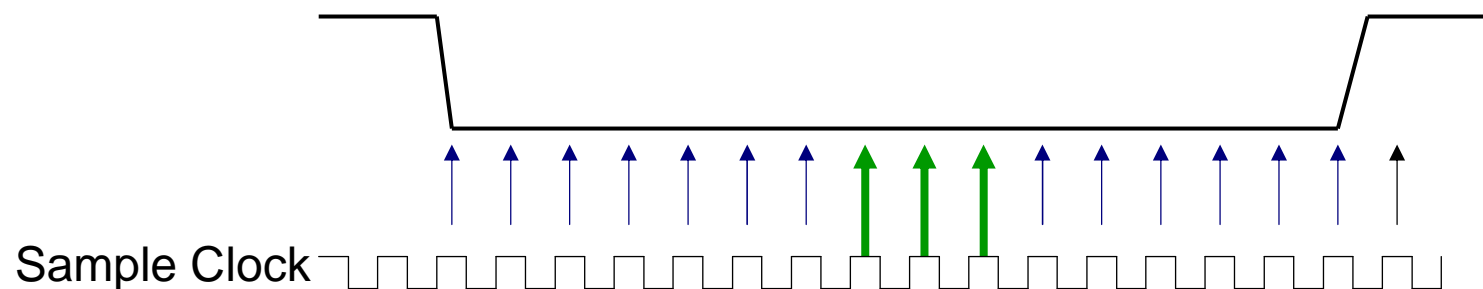


Bit-Synchronization

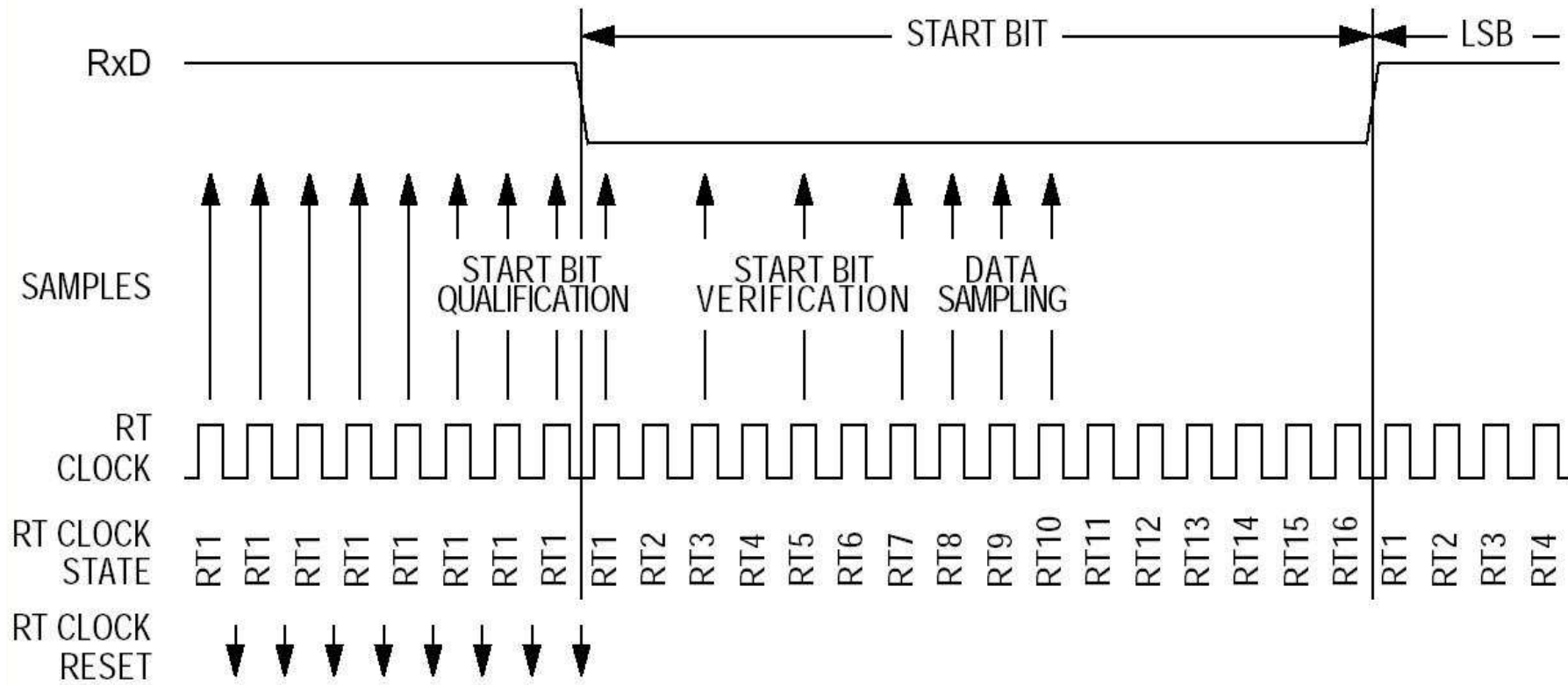
- A start bit transition to a low logic level (dominant) indicates a start of a byte, least significant first and completing with a logic high level (resessive) bit to indicate the STOP bit



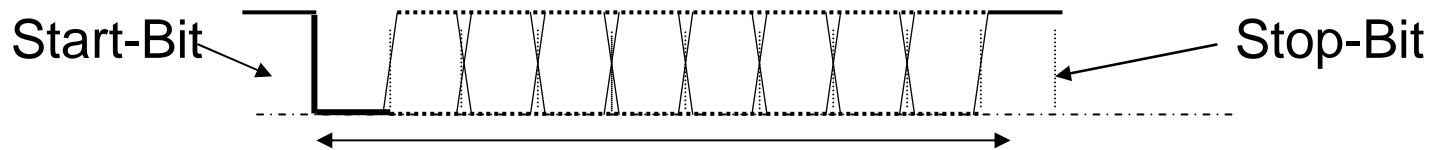
Data is sampled in the middle of the bit field:



Bit Sampling

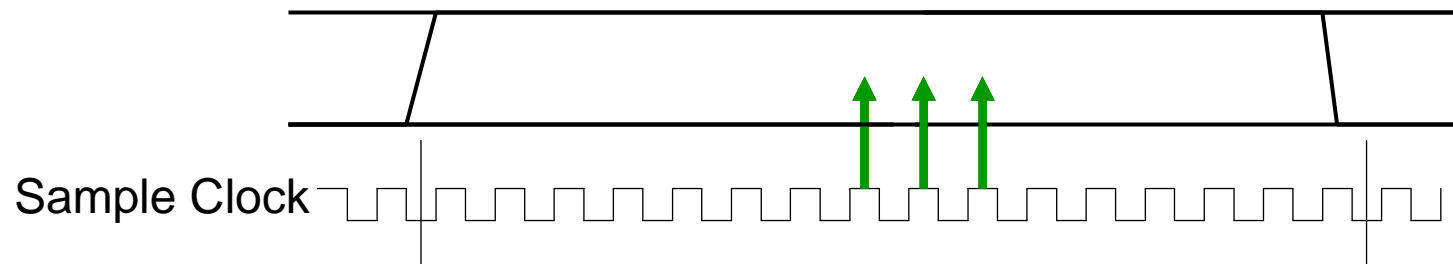


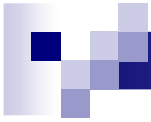
Bit-Synchronisation



After recognition of a Low level in the start bit, the data is sampled at a rate 16 times the bit rate expected. The middle 3 samples must all agree for an error free reception of the data.

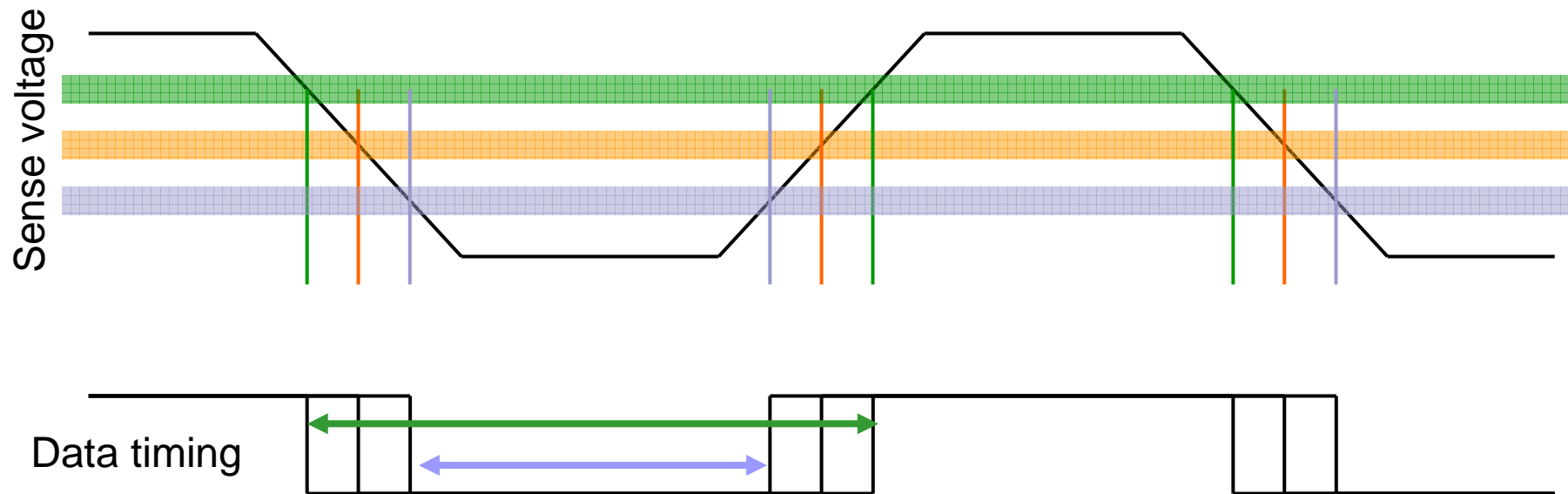
A stop bit is expected after 1 start bit and 8 data bits in a typical message





Taking account of Ground-Shift

The detection point for data transitions can be affected by voltage references. Ground shift can change this reference by a significant amount, affecting the bit timing of the data

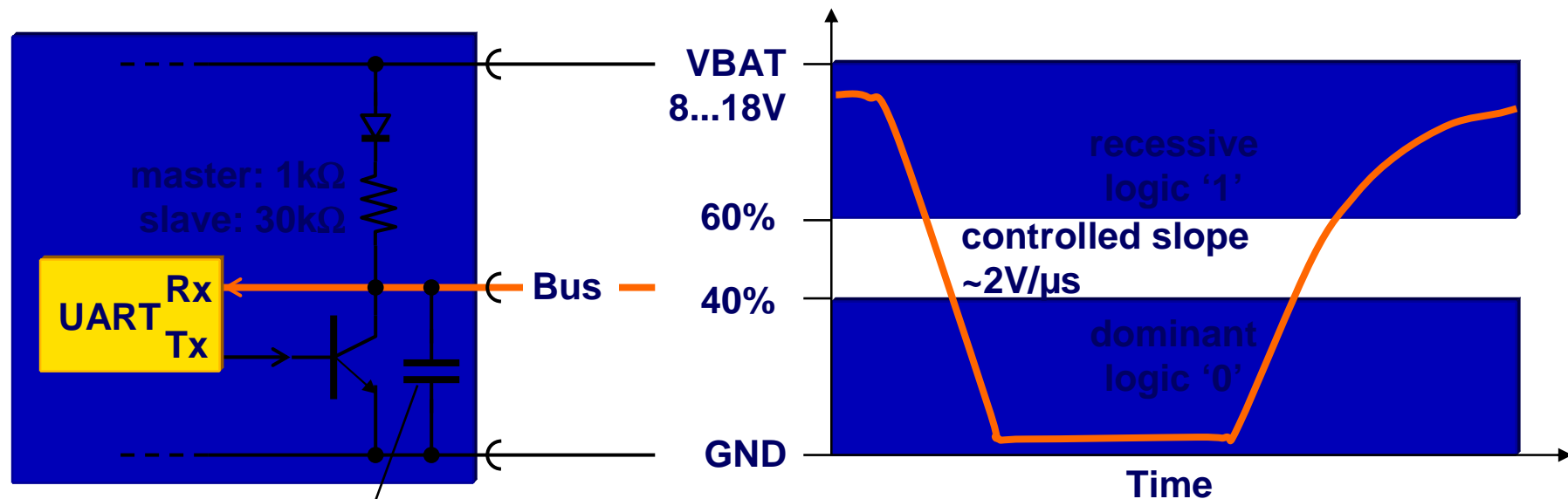


Available bit sampling zone can reduce worst case bit width to around 40us at 20k baud
This affects the overall baud rate tolerance required for safe LIN communications

LIN Physical Interface

Electronic Control Unit

Bus Voltage



Example capacitances

master: 2.2nF

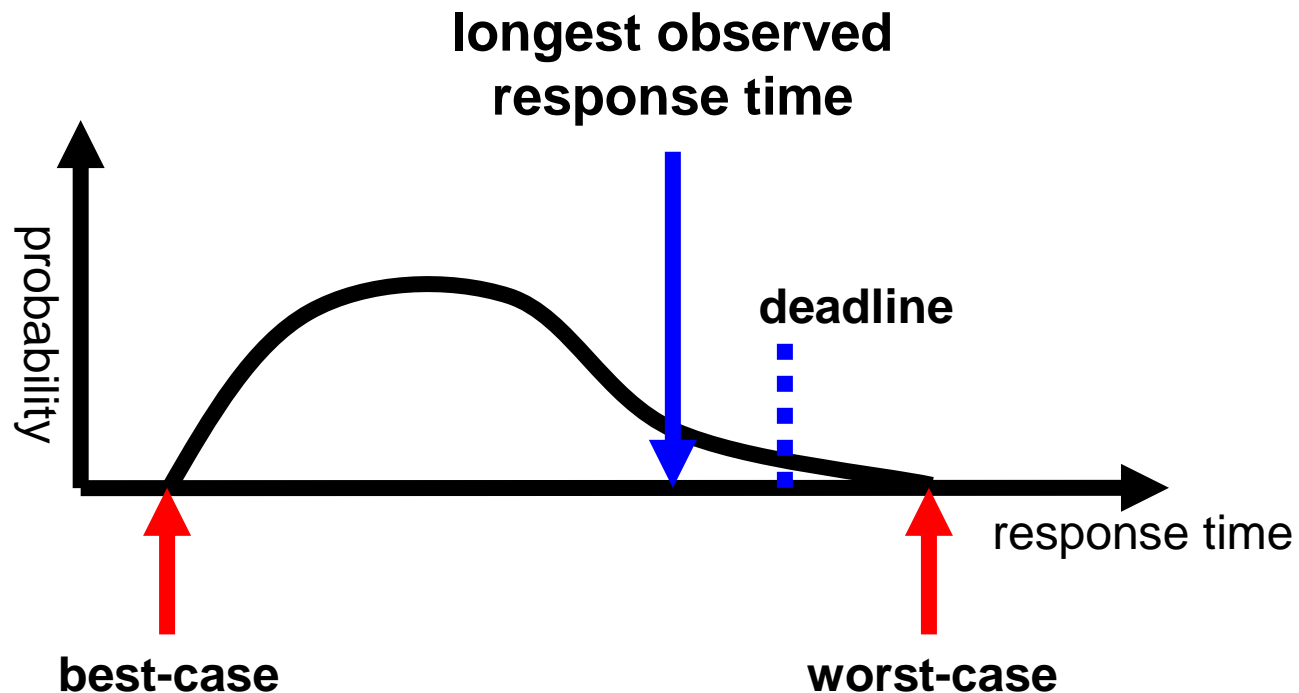
slave: 220pF

Note:

The LIN specification refers to the ECU connector voltages !

Examination of whether the Deadline is met

- Signal based messaging with static latency analysis ensures that all signals meet defined minimum latency times
- Drives need for complete configuration tool support to ensure guaranteed timing of all signals in a LIN network





signal latencies

■ generation latency

the time between an event input (e.g. a button pressed) and the signal being placed in a buffer, ready for transmission

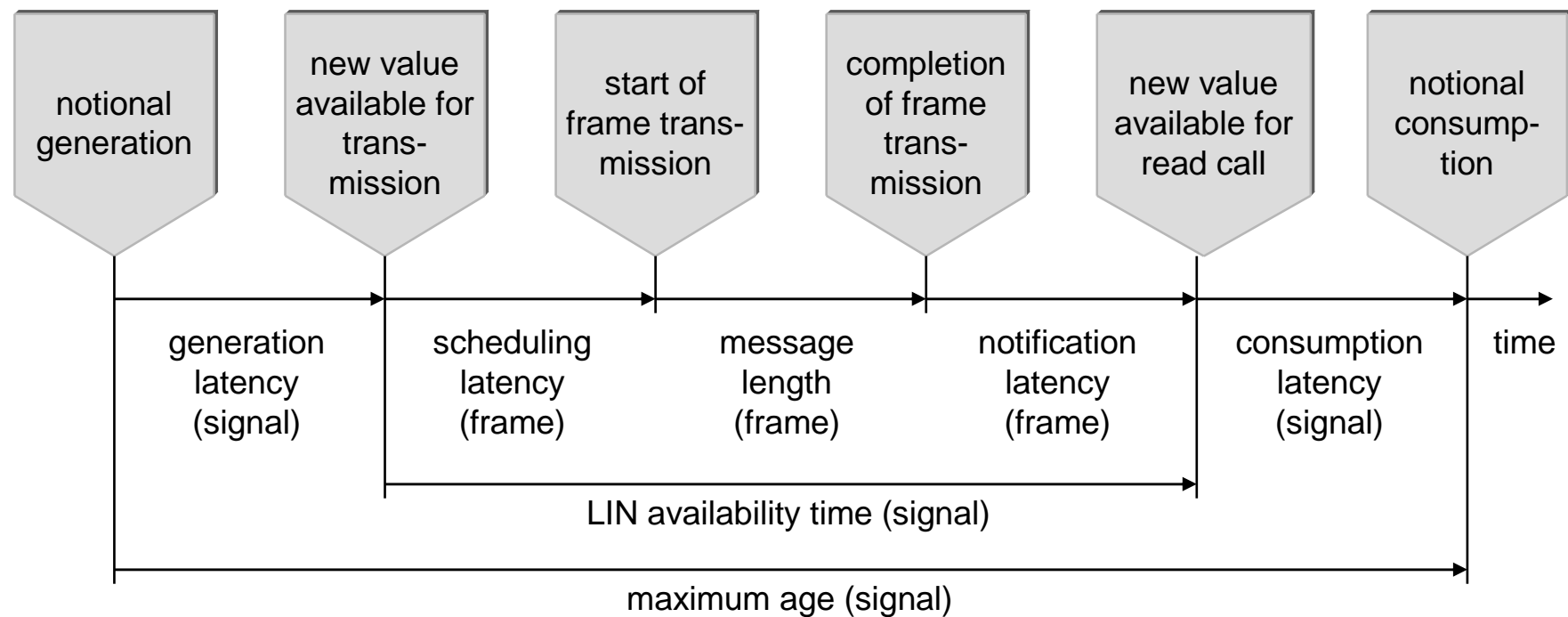
■ consumption latency

the time from when a signal has been received from the network into a buffer until it has been read into the subscribing application and some action has taken place (e.g. a motor starts moving).

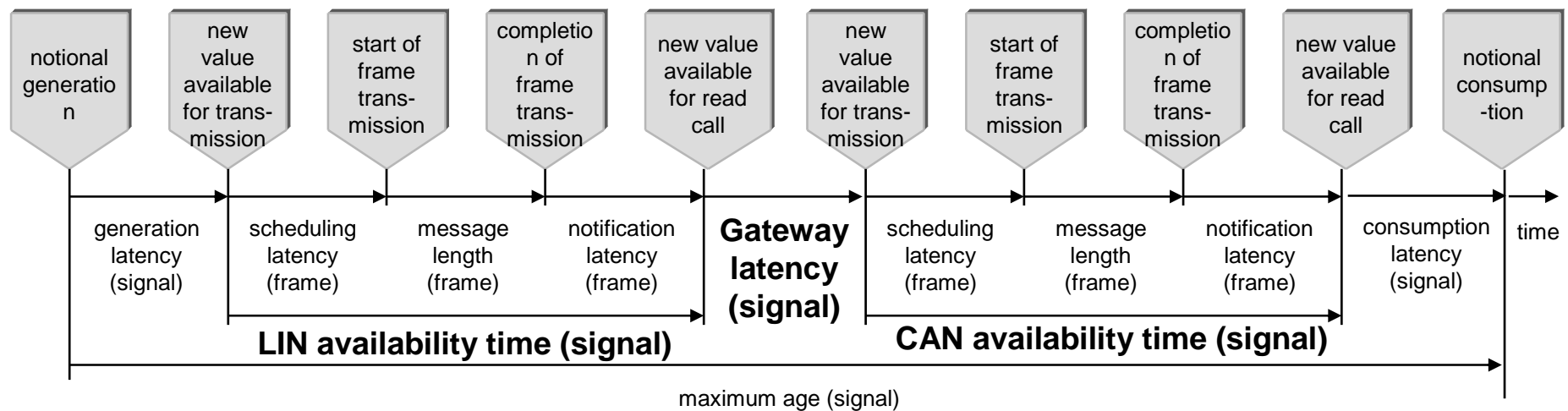
■ maximum age

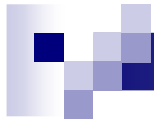
This is the maximum allowed time between a user action in the publishing node until the subscribing node is actually carrying out an action.

Message latency



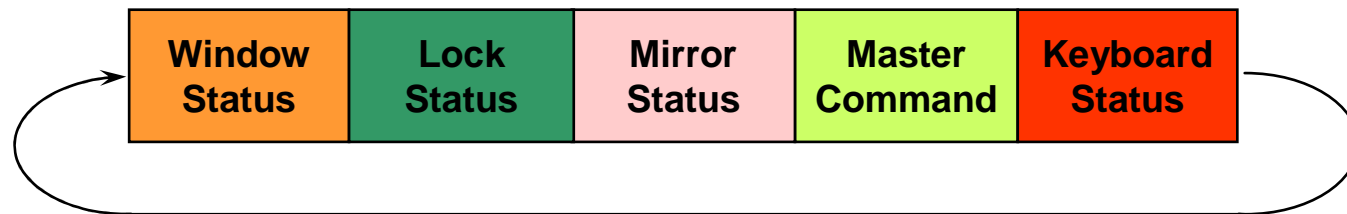
Message latency across a network



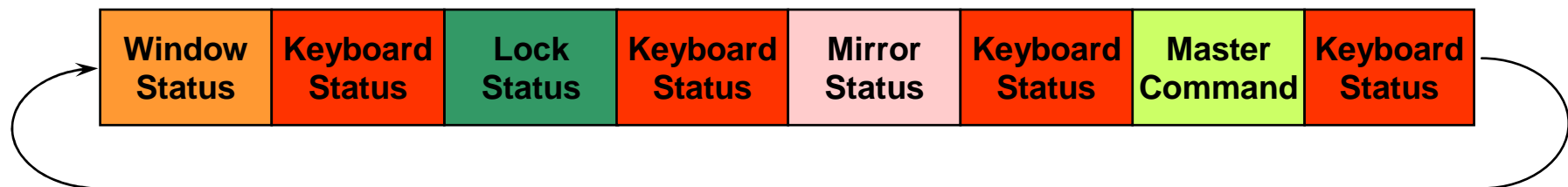


Latency optimizations with LIN

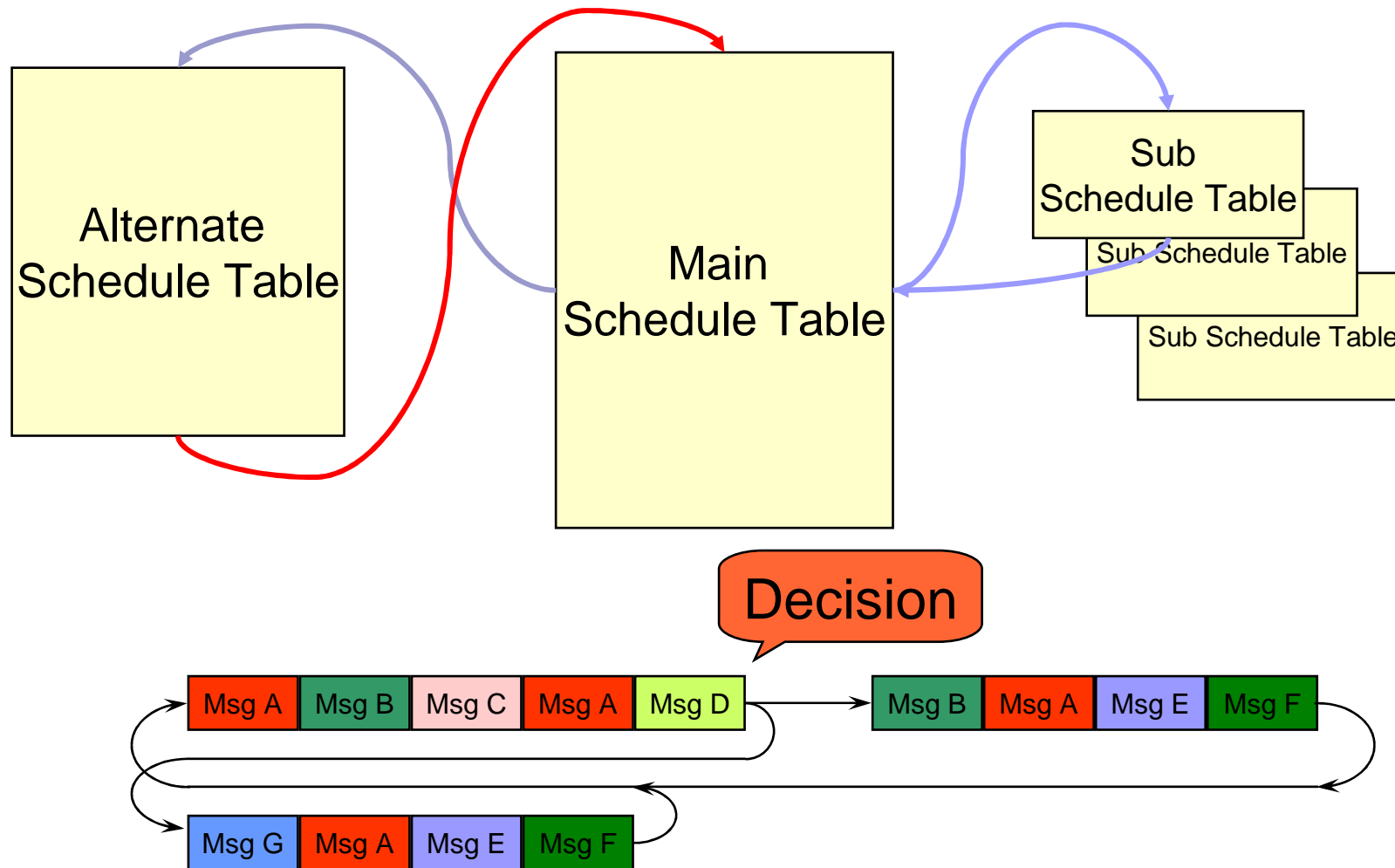
Basic schedule



Alternate schedule for low latency signals from a keyboard



Variables Scheduling





Event Triggered Message

- Problem
 - Specific node communication required but this takes up too much time for all network messages
- Solution : Event Triggered frame:
 - Header is sent out
 1. normal case: no answer
 2. Rare response: only one node responds
 3. Very rare response : several nodes respond simultaneously
- Cases 1 and 3 are exceptions that should be addressed at the application design.
- Event triggered messaging is complementary to the regular signal based messaging scheme

LIN Development Flow

