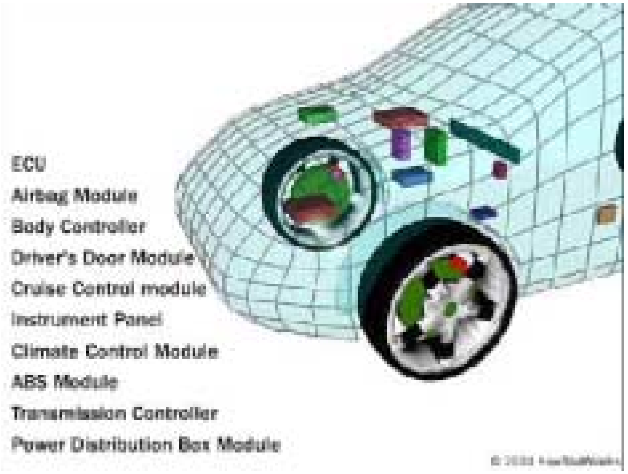
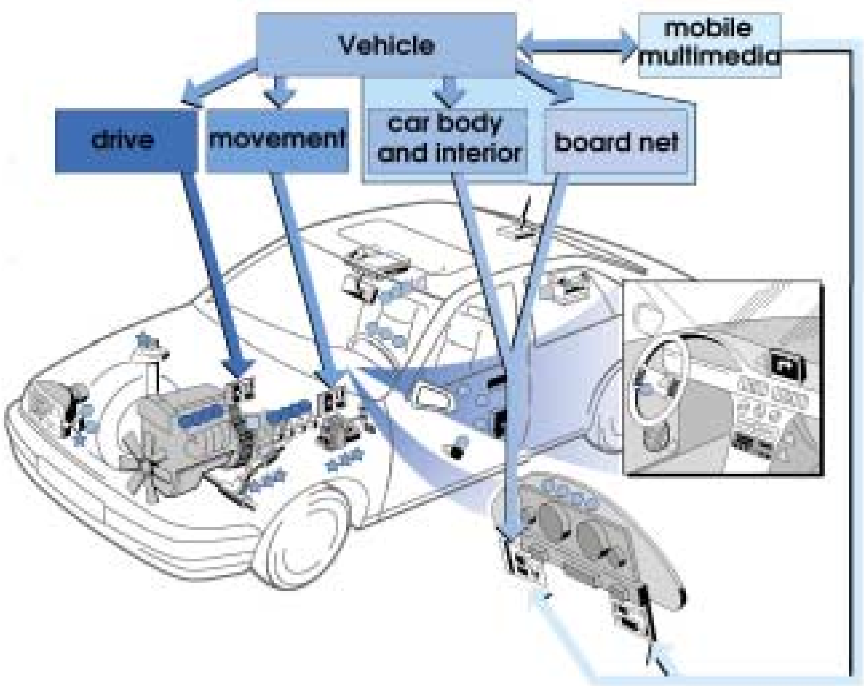
Field buses

Nico Fritz
Universität des Saarlandes

Communication of Embedded Devices

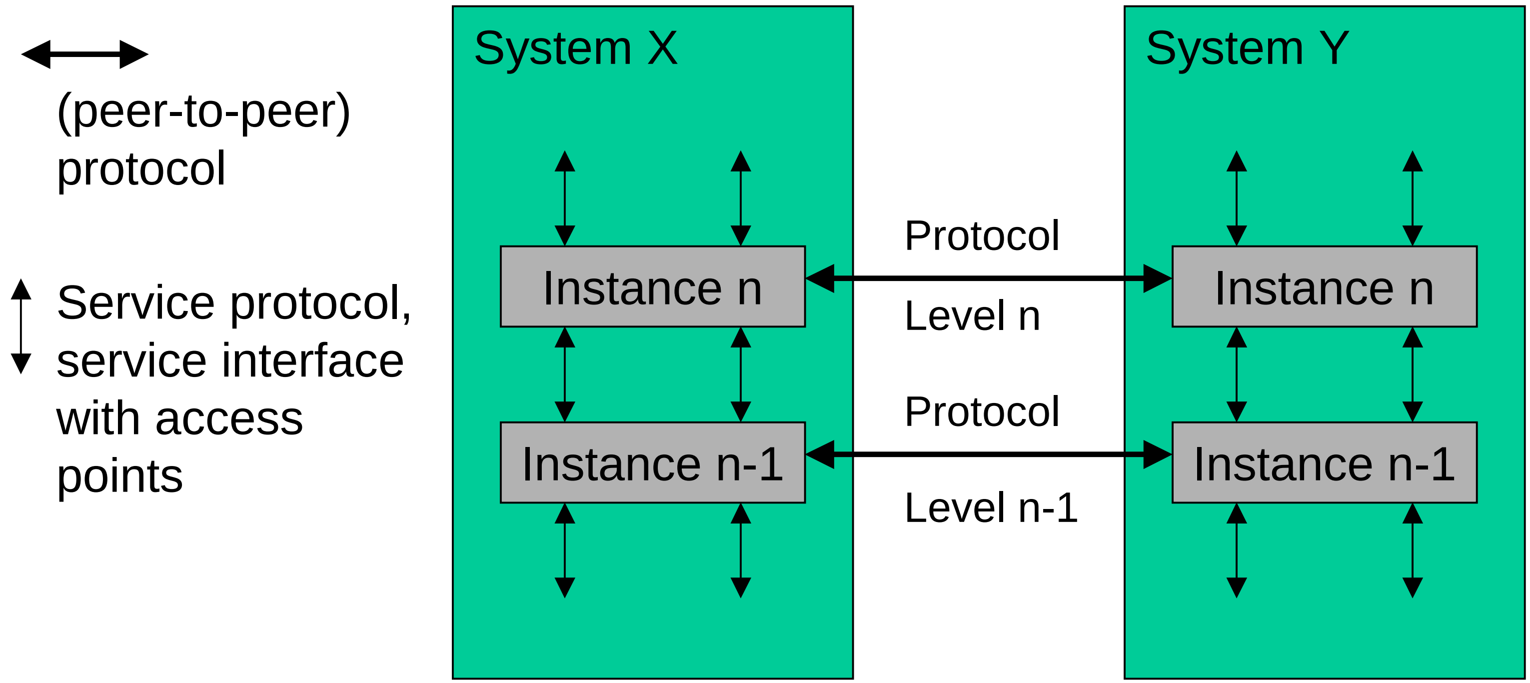
- With other embedded devices or with sensors.





Protocols

- Protocols are rules for data exchange between two partners on the same level (peer-to-peer).



ISO/OSI Layer model

- ISO: International Standards Organization
  OSI: Open Systems Interconnection Reference Model
- Recommendations for the structure and the course of the communication
  between two or more computers

7	Application Layer	Provides network services to end-users (like E-Mail, distributed data bases)	Software
6	Presentation Layer	Converts local representation into canonical form and vice versa	..
5	Session Layer	Allows applications on 2 different systems to establish, end and use sessions (log on/off),.	..
4	Transport Layer	Error recognition and recovery; repacking of long messages and rebuilding.	..
3	Network Layer	Establishes, maintains and terminates network connections. Routing; logical to physical address	..
2	Data Link Layer	Packing of raw bits into message frames; placing bits of frames into the physical layer	..
1	Physical Layer	Defines the cable or the physical medium itself, e.g. unshielded twisted pairs (UTP). All media are functionally equivalent. Differences in convenience, cost, maintenance	Hardware

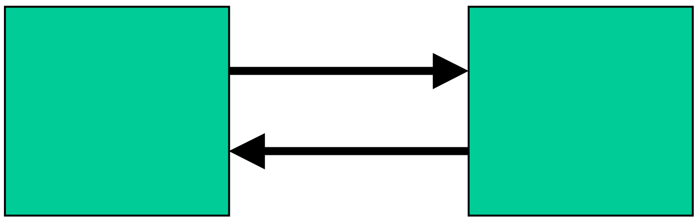
- Layer 5 to 7: Responsible for providing and getting data
- Layer 1 to 4: Responsible for data transport
- Field buses implement only layers 1, 2 and 7

Tasks and demands of field buses

- Mostly connecting processors and (simple) field devices
- Real-time conditions:
  - Deterministic access behavior
  - Cycle times from 1 to 10 msecs with 40 to 60 devices
  - Efficient protocol even with little data to be send
  - Priorities for messages
  - Optimization of cyclic messaging
- Many devices on the bus
- Other requirements:
  - Dependable in rough environments
  - Low costs
  - Simple and stable protocol

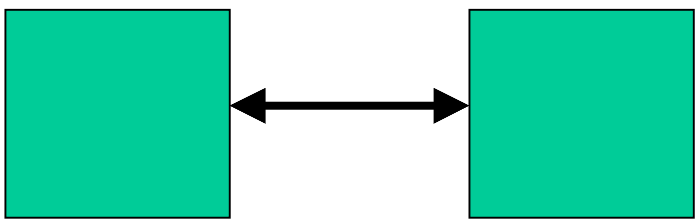
Connection structures

- Peer-to-peer, full duplex



□No conflicts

- Peer-to-peer, half duplex

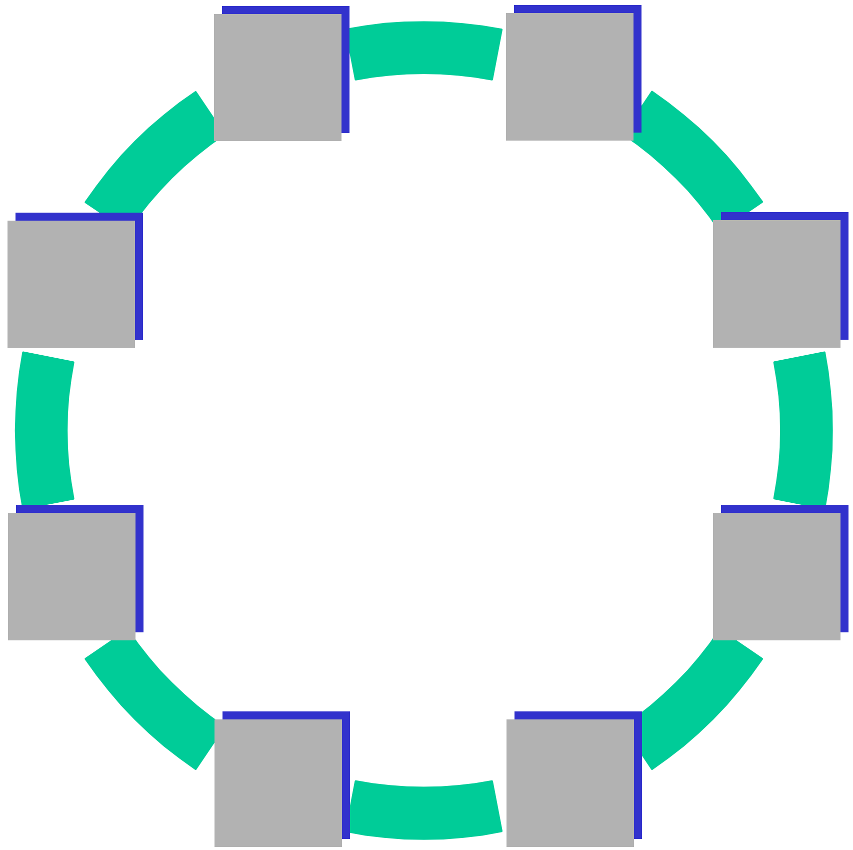


□Conflicts when writing at the same time

□Simultaneous usage must be avoided by some access strategy (arbitration).

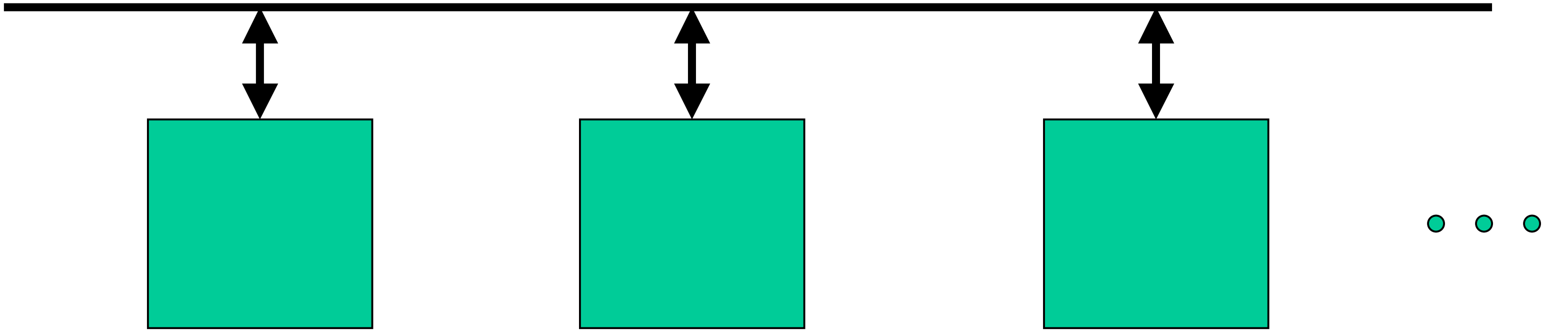
Connection structures

- Ring structure
  - Participants are connected to a closed transfer ring
  - Data runs through the ring and is deleted again by the sender
  - Arbitration mostly by tokens
  - Ring does not mean a continuous loop of cable
  - Example: Token-Ring



Connection structures

- Bus structure



  - Arbitration necessary
  - Examples: CSMA/CD (Ethernet), CAN-Bus
  - Note: Ethernet is not real-time capable since delay times can be arbitrarily high in case of high load

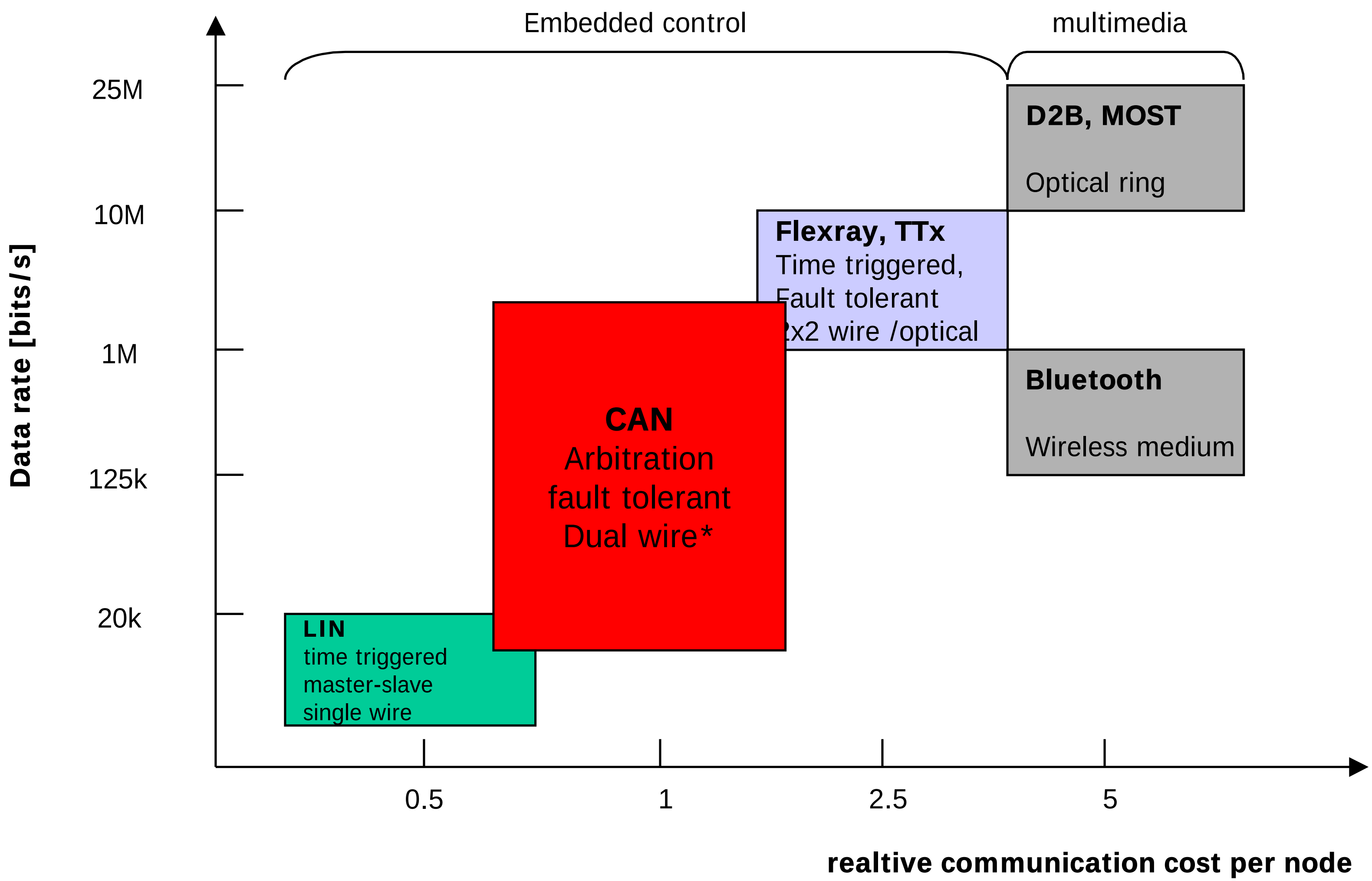
Bus structure

- Advantages:
  - Versatility:
    - new devices can be added easily
    - peripherals can be moved between systems with the same bus
  - Low costs
    - a single set of wires is shared in multiple ways
- Disadvantages:
  - It creates a communication bottleneck
  - Maximum bus speed is limited by
    - number of nodes
    - length of the bus
    - (support for devices with different latencies/ data transfer rates)

Bus access

- Frequency multiplexing
  - No real arbitration
  - One or more participant(s) per channel
- Time multiplexing
  - Central arbitration
    - Bus-arbiter
    - Daisy-Chain
    - Master-Slave
  - Decentral arbitration
    - Token passing
    - Special solutions (CAN-Bus)
  - Random access
    - Without reservation (CSMA/CD)
    - With reservation (Multi level multi access)

Field buses in the automotive industry



CAN-Bus

- CAN = Controller Area Network,
  - serial communications protocol
- Aims:
  - Reliable bus for connecting sensors, activators and cpus
  - Mainly used in automotive industry i.e. Anti-skid-systems
- Developed by Bosch in 1983
  - Licences to Siemens, Intel, Philips, Motorola, ...
- User organisation CiA (CAN in Automation)
- Standardized: ISO 11519 and ISO 11898 for layers 1 and 2
- Reference: Controller Area Network protocol specification, Version 2.0, Robert Bosch GmbH
  - (http://www.can.bosch.com)

CAN concepts

- Priorization of messages
- Guarantee of latency times
- Configuration flexibility
- Multicast reception with time synchronization
- System wide data consistency
- Multimaster
- Error detection and signalling
- Automatic retransmission of corrupted messages
- Error distinction

CAN layer 1

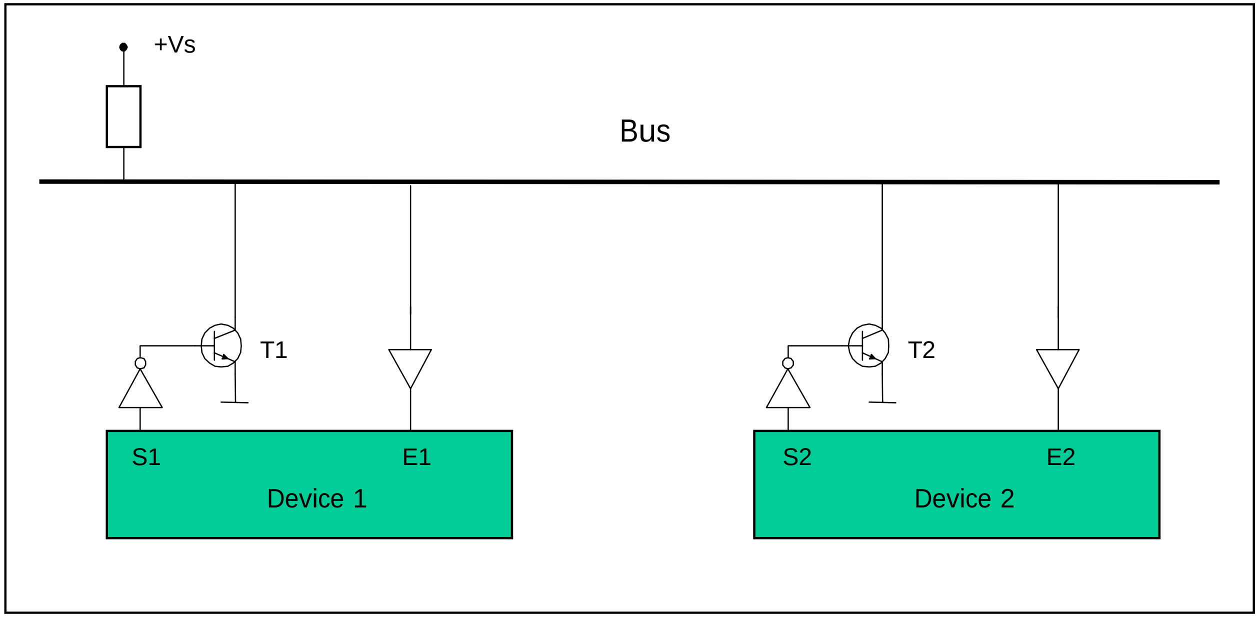
- Speed:
  - High speed: 125kbps to Mbps (up to 40m)
  - Low speed: 5kbps (10km) to 125kbps (~500m)
- Physical transmission layer not fixed
  - Two wire differential transmission (RS-485)
  - Single wire
  - optical
- Tasks:
  - Bit encoding and decoding
  - Bit timing
  - Synchronization
- CSMA/CA (carrier sense multi access / collision avoidance)

CAN layer 1: CSMA/CA

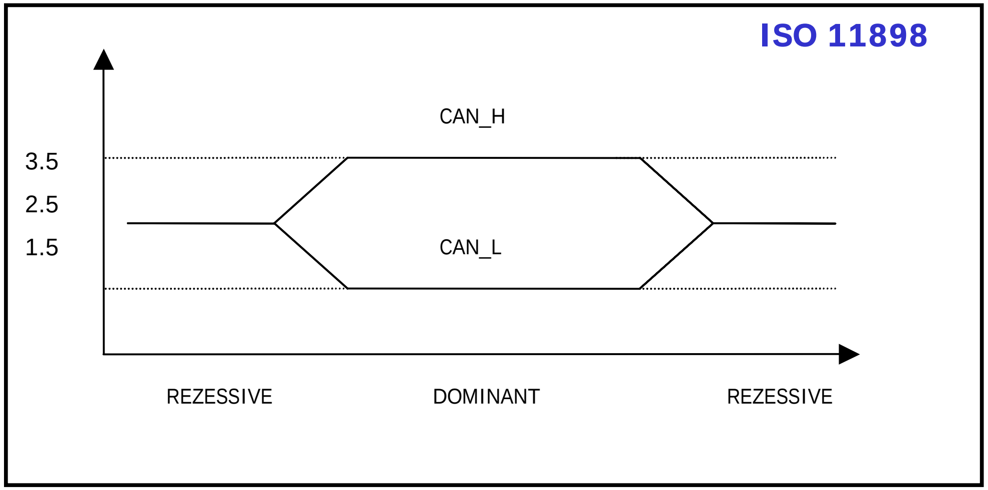
- A dominant D-Bit and a recessive R-Bit are implemented according to the bus gauge
  - D-Bit corresponds to a logical 0
  - R-Bit corresponds to a logical 1
- If two devices send at the same time, the D-Bit out weights the R-Bit
- The transmitter of the R-Bit can see the synchronous sending of the D-Bit on the bus gauge
- A station compares Ei with Si and stops sending if Ei<>Si

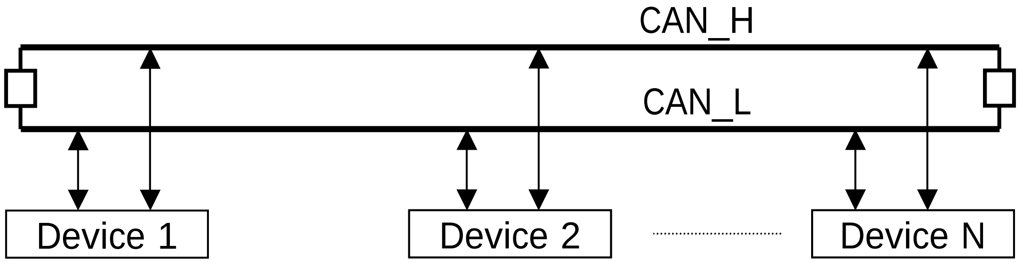
CAN layer 1

- Implementation of D- and R-Bit, i.e. wired-AND-circuit
  - Transistor Ti conducts if Si = 0, then bus gauge = 0
  - So, D-Bit out weights R-Bit



CAN layer 1: Voltage (2 wires)



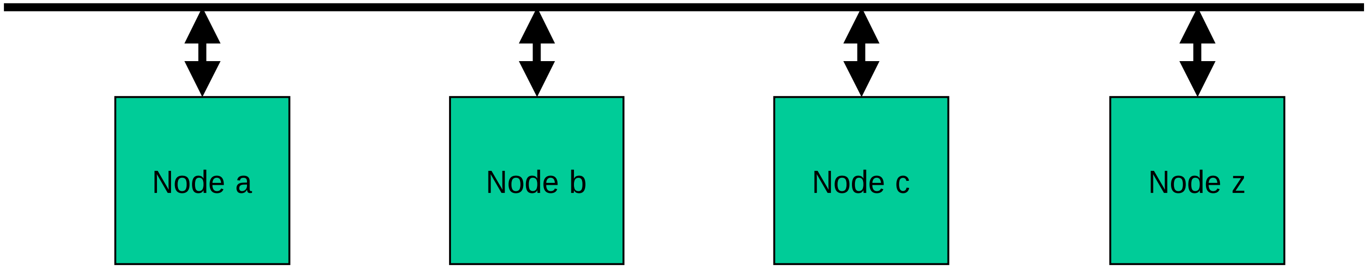


CAN layer 1: Arbitration

- A station may send if the bus is free (carrier sense)
- Any message begins with a field for unique bus arbitration containing the message ID
- The station with the lowest ID is dominant (D-Bit)
- So the lowest ID has highest priority
- Sending is not interfered since the propagation on the bus is much smaller than a duration of a bit

CAN layer 1: Arbitration

- Example



Node	ID					
a	0	1	1	X	X	X
b	0	1	0	1	0	1
c	0	1	0	1	1	X
bus	0	1	0	1	0	1

- b wins
- a,c have to wait for next idle

CAN layer 1

- Bit Timing
  - (nominal bit time) = 1/(nominal bit rate)
  - A NBT is divided in 4 segments:
    - Synchronization
    - Propagation Time
    - Phase buffer 1
    - Phase buffer 2
  - The sample point is between phase buffer 1 and phase buffer 2
- Synchronization
  - Hard synchronization in the synch_seg
  - Resynchronization after phase errors phase buffer 1 can be lengthened or phase buffer 2 can be shortened

CAN layer 2: Versions

- According to specification 2.0 there are 2 different versions of CAN
  - Version 2.0A which is similar to Version 1.0-1.2
  - Version 2.0B which has additional extended identifiers
- Complying with CAN 2.0
  - 2.0B active: works with 29bit identifiers
  - 2.0B passive: discards the additional 18 bits without error
  - a CAN 1.0-1.2 controller would detect an error when receiving an extended identifier

CAN layer 2

- Divided in 2 parts in version 2.0B: LLC and MAC
  - in 2.0A they are called object and transfer layer
- LLC = Logical Link Control
  - Acceptance filtering
  - Recovery management (from errors)
- MAC = Medium Access Control
  - Data de- and encapsulation
  - Frame coding (Bit-Stuffing, Destuffing)
  - Error detection and signalling
  - Acknowledgement
  - Serialization/Deserialization

CAN layer 2

A DATA FRAME consists of seven different bit fields:

DATA FRAME:

- IS: Interframe space
- SOF: Start of frame, one single D-bit, start only if the bus is IDLE, all devices have to synchronize to the leading edge caused by START OF FRAME.
- ID: Identifier (CAN 2.0A (standard) = 11 bit, CAN 2.0B (extended) = 29 bit)
- RTR: Remote transmission request
  - D-bit: data follows = DATA FRAME
  - R-bit: transmission request to receiver = REMOTE RAME
- DLC: Data Length Code = 6 bit, C[3] - C[0] length of data array, MSB first
  - REMOTE FRAME: number of requested data bytes
  - C[5], C[4] are used for indicating extended IDs (2.0B)
- CRC: Cyclic redundancy checksum; 15 bit and a leading 0, sum and a R-bit delimiter bit
- ACK: Acknowledge (2 bits: ACK slot a and ACK delimiter)
  - The bit in ACK slot is sent as a R-bit and overwritten as a D-bit by those transducers which have received the message correctly.
- EOF: End of frame (7 R-bits)

Bit	>3	1	11,1	6	0…64	16	2	7
	IS	SOF	ID, RTR	DLC	DATA	CRC	ACK	EOF

CAN layer 2

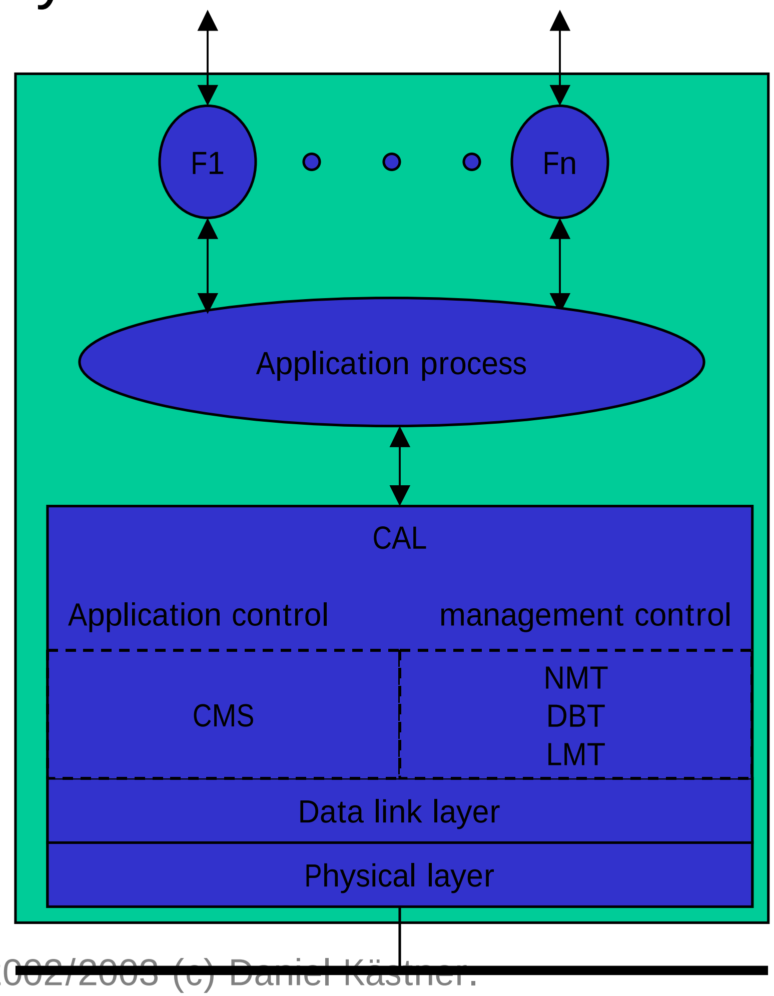
- Maximum length of a message with 8 byte of data is 3+1+11+1+6+64+16+2+7 = 111 bits
- This is the maximum delay of a high priority message
- ID contains object marker so that the receiver recognizes the content of the message
- A station only reads the messages destined for it (acceptance filter)
- Acknowledgement: Sender sets **a** to R. If the CRC-check succeeds, receiver sets **a** to D. So the sender sees the D as a positive acknowledgement.

CAN layer 2: Errors

- There are 5 different, not mutually exclusive error types:
  - Bit-Error: detected by a unit while sending.
  - Bitstuff-Error: 6 equal bits in a row.
  - CRC-Error: calculation of the receiver differs with the CRC field
  - Form-Error: fixed-form bit contains one or more illegal bits
  - Acknowledgement-Error: no D bit during ACK slot
- Handling:
  - Each node has an error counter
  - Different errors increase the counter by different values
  - When exceeding some limit a node can be cut off
  - Errors cause a resending of the message

CAN layer 7

- CAL = CAN Application Layer
  - Specified by CiA in 1993
  - Intended to provide standard app.-indep. communication facilities
- Structure:
  - CMS (CAN based message spec.)
  - DBT Distributor
  - NMT network management
  - LMT layer management



CAN layer 7: Modules

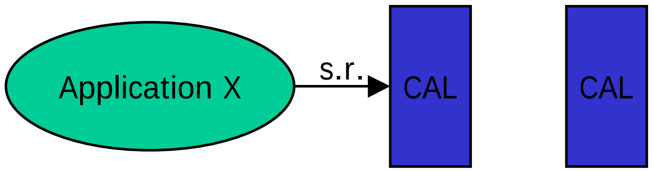
- CMS
  - Objects are described by name, type, priority, minimum sending repetition
  - Usual services are Read, Write, Notify, Load and so on
- DBT
  - Dynamic assignment of CAN identifier to CMS object at initialization
  - System error detection
  - Simplifies usage of devices from different manufacturers
  - Net wide consistency of IDs for senders and receivers

CAN layer 7: Modules

- NMT:
  - Initialization, starting and stopping of processes on nodes
  - Detection of system errors
  - Read/Write of parameters on nodes during initialization
- LMT:
  - Setting up time parameters in layer 2

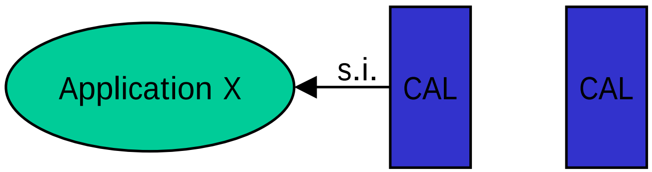
CAN layer 7: CMS services

- Local services: Executed by the CAL module itself



s.r. = service request

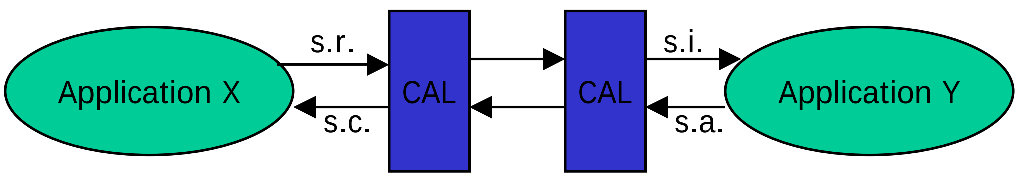
- Unrequested Services: CAL shows a detected event



s.i. = service indication

CAN layer 7: CMS services

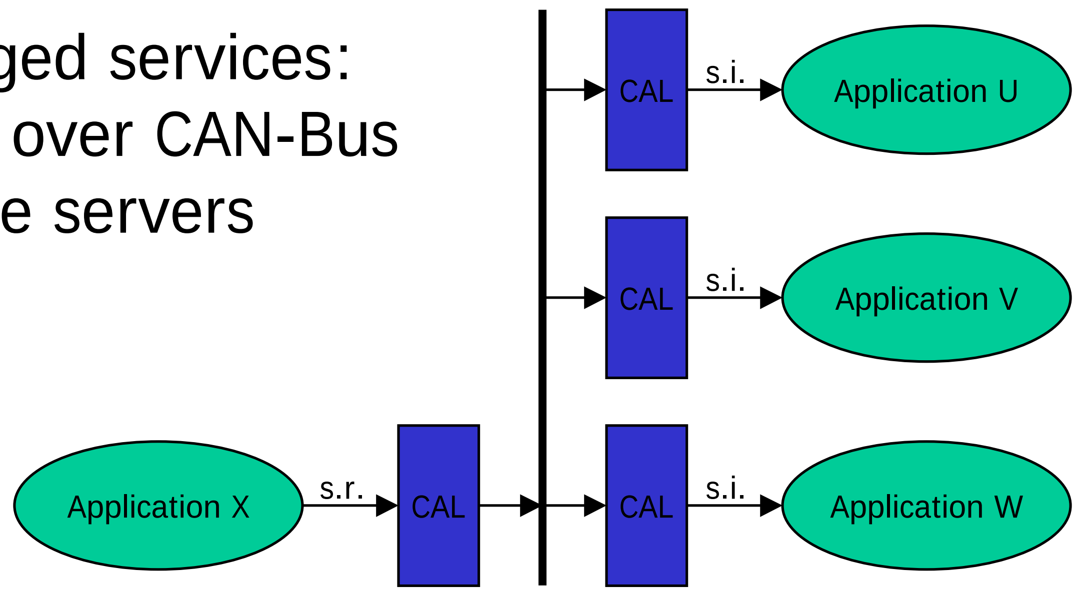
- Acknowledged services: Request by one server and response by one server



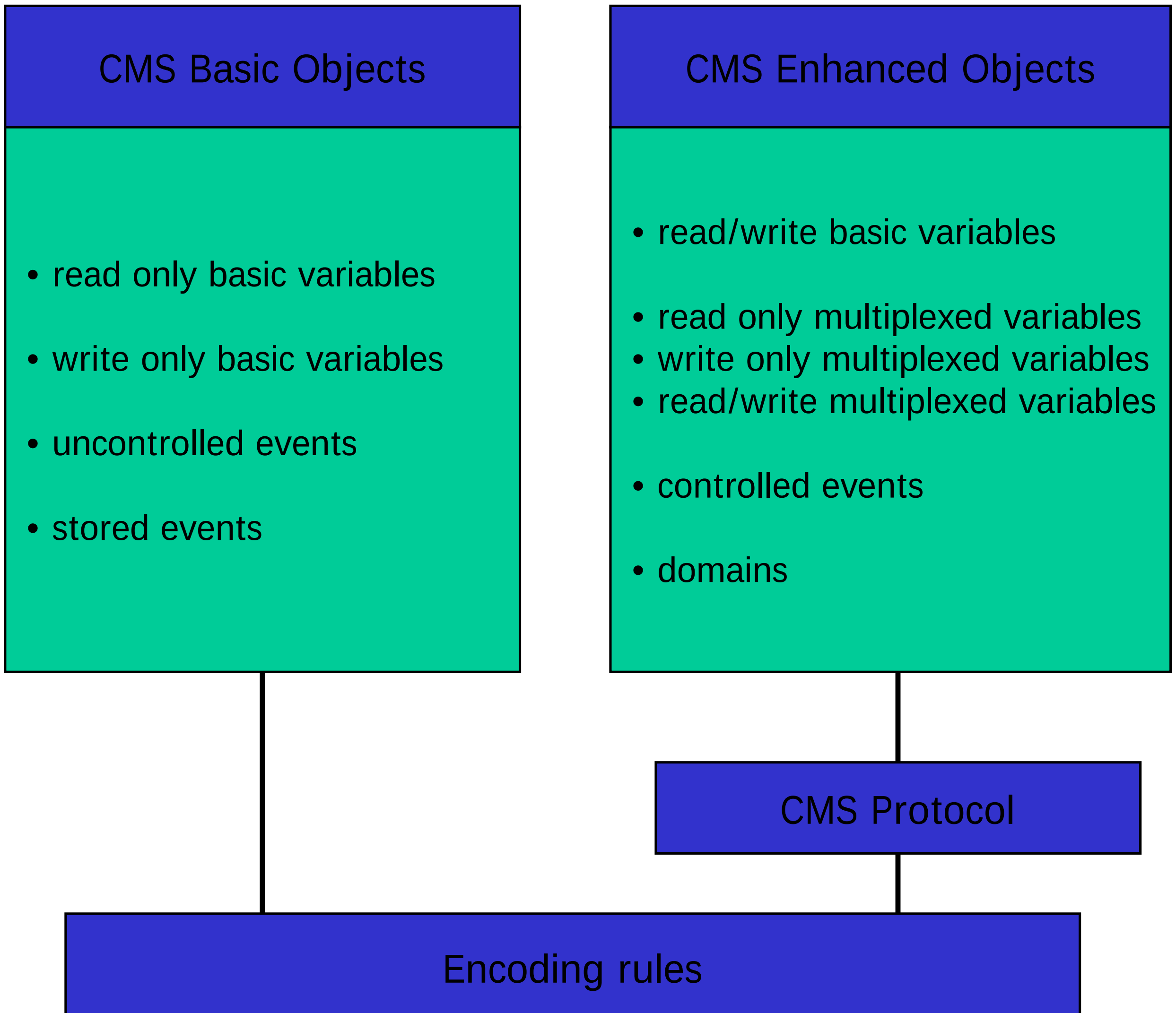
s.a. = service response

s.c. = service confirmation

- Unacknowledged services: Request send over CAN-Bus to one or more servers



CAN layer 7: CMS model

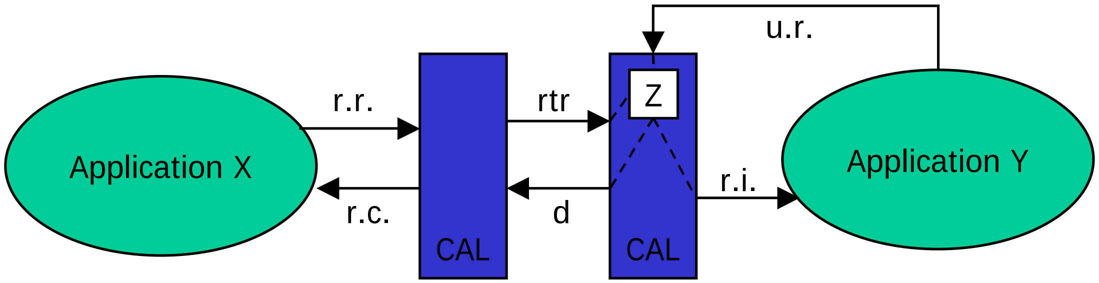


CAN layer 7: CMS basic objects

- Read only variable
  - Only readable by the client
  - Read indication send to server process
- Write only variable
  - Only readable by the client
  - Write indication send to server process
- Uncontrolled event
  - Message of an occurred event is instantly send to all (at initialization time defined) clients
- Stored event
  - Properties of a read only variable
  - Possible additional "store and notify". All clients get a notification with an event value.

CAN layer 7: CMS basic objects

- Services for basic objects:
  - Message identifier also uniquely determines the basic object
  - Access of variables directly handled by the CMS
  - Servers can request “store” (local) or “store and notify” (no ack.), clients can request “read” (ack’ed)
  - Example: Application Y holds a basic variable Z (I.e. temperature value) which is peridiodically updated. X can always access Z by a read request and Y is informed.



r.r. = read request

rtr = remote transmit request

u.r. = updated request

r.i. = read indication

r.c. = read confirmation

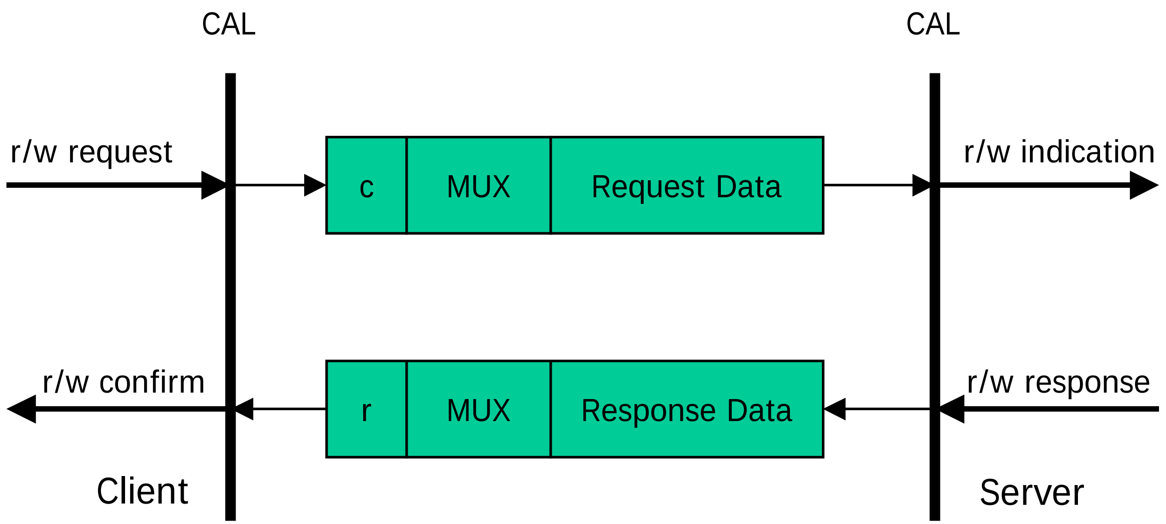
d = CAN message with data

CAN layer 7: CMS enhanced objects

- Read/write basic variables
  - Specified in CAN message
- Multiplexed variables
  - More variables are packed to a CMS object
  - Structures and fields
- Domains
  - Data field > 8 bytes, I.e. for programs
  - Services like upload and download
- Controlled events (for synchronization)
  - Indication of an event can be shown or hidden
  - Exactly one server and one client

CAN layer 7: CMS enhanced objects

- Example:



MUX: 0 basic variable or first element of one multiplexed variable;
>0 index of a multiplexed variable. (<=128)

c: request code (0 write, 1 read)

r: result code (0 success, 1 failure)

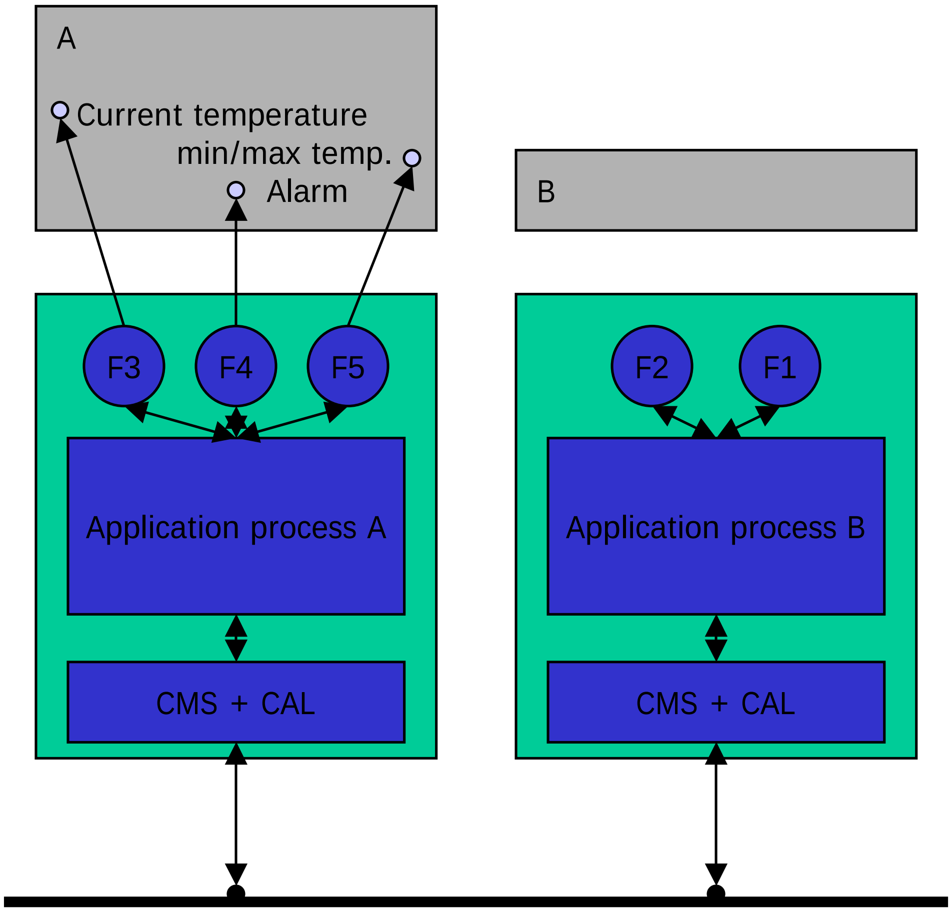
Request Data: Data to be written if c=0

Response Data: written data if c=0, r=0
read data if c=1, r=0

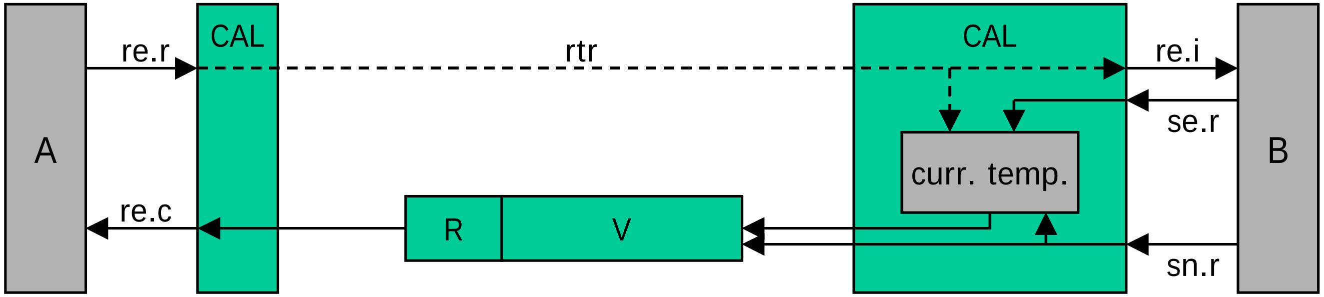
error code if r=1

CAN layer 7: Example

- Module A displays the current temperature:
  - F3 updates the temperature periodically
  - F4 rings an alarm when the limit is passed
  - F5 updates the display of the extreme temperatures
- Module B measures the temperature:
  - F1 reads the temperature periodically (sensor)
  - F2 determines the extreme values



CAN layer 7: Example



re.r = read event request

rtr = remote transmission request

re.i = read event indication

se.r = store event request

sn.r = store and notify request

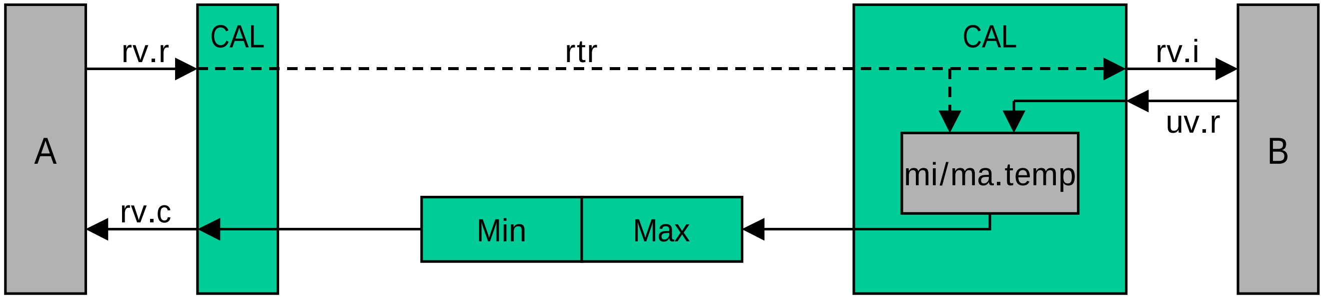
re.c = read event confirmation

R = in/out of range

V = value of temperature

curr.temp. = current temperature, stored event, structure: R (1Byte), V (2Byte)

CAN layer 7: Example



rv.r = read variable request

rtr = remote transmission request

rv.i = read variable indication

uv.r = update variable request

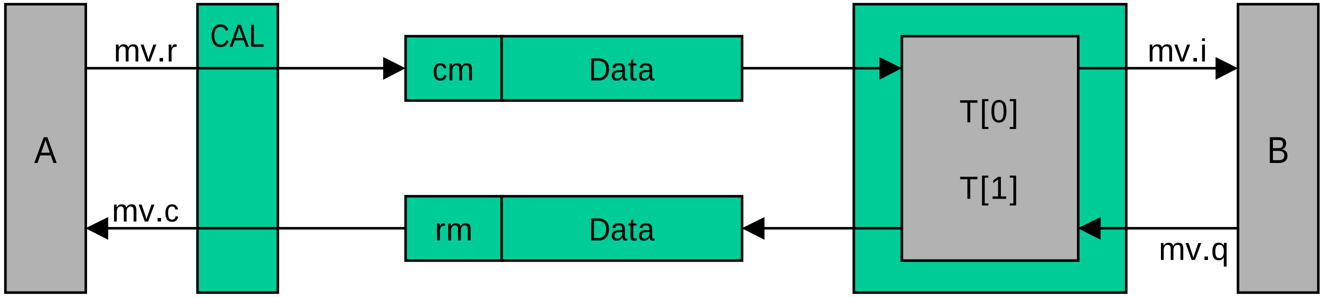
Min = minimum temperature

Max = maximum temperature

mi/ma.temp. = min. and max. temperature,

read only var., structure: Min (2Byte), Max (2Byte)

CAN layer 7: Example



mv.r = r/w mux variable request

mv.i = r/w mux variable indication

mv.q = r/w mux variable response

mv.c = r/w mux variable confirmation

cm = c (1Bit), mux (3Bit)

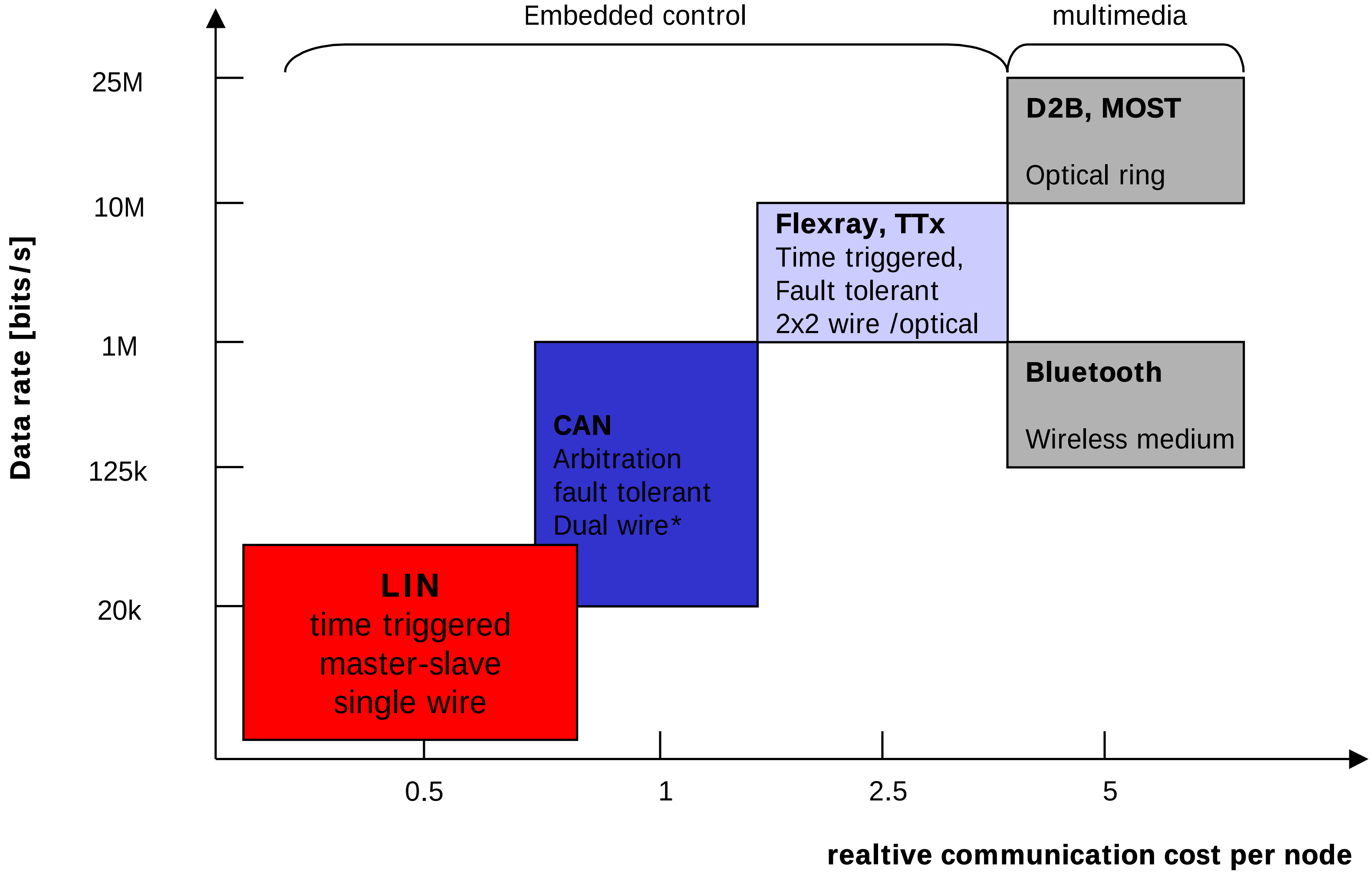
rm = r (1Bit), mux (3Bit)

T = temperature, r/w multiplexed variable

T[0] = upper bound temperature

T[1] = lower bound temperature

Field buses in the automotive industry



LIN

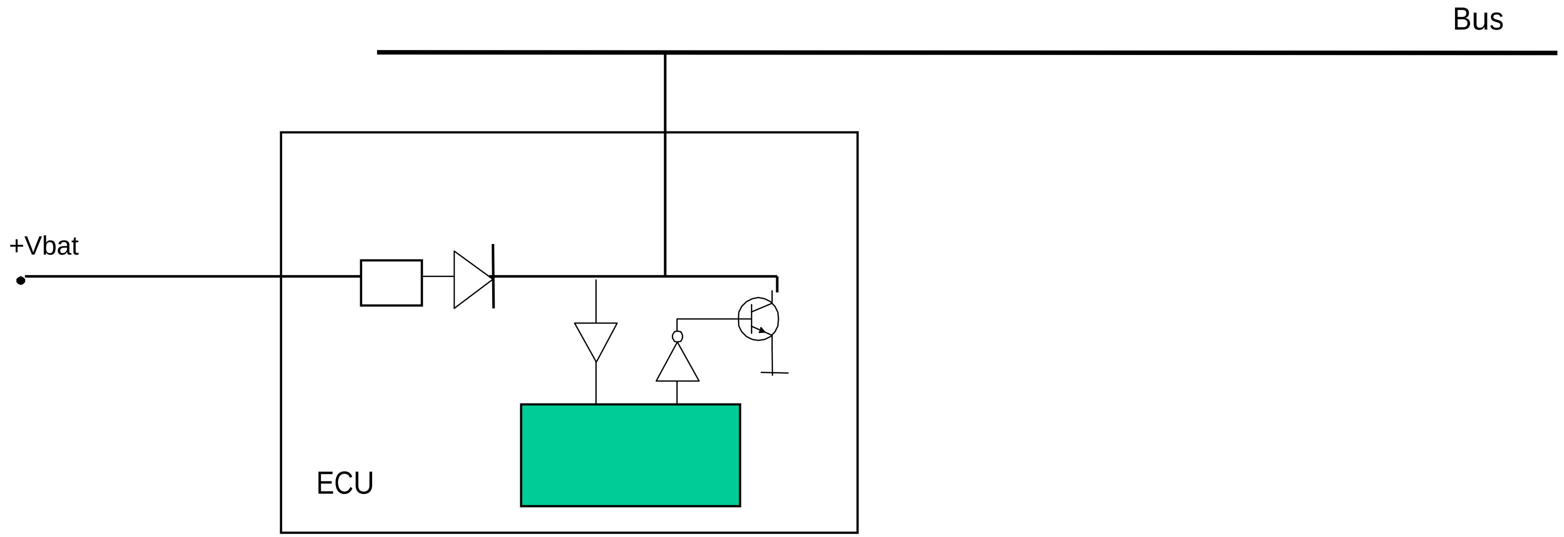
- LIN = Local Interconnect Network
  - Serial communications protocol
- Developed in 2000 by Audi, BMW, DaimlerChrysler, Motorola, Volcano Comm. Techn., VW, Volvo
- Speed up to 20kbps
- Aims:
  - Low cost automotive network
  - Quality enhancement (hierarchical vehicle networks)
  - Cost reduction (replacing existing low-end multiplex solutions)
- Reference: LIN Specification Package 1.2
  - http://www.can.bosch.com/LIN/LIN.html

LIN: basic concepts

- Single master / multiple slaves (no arbitration)
- Multi-cast reception with self synchronization
- Guarantee of latency times for signal transmissions
- Low cost single wire implementation
- Minimum cost for semiconductor components (small die-size, single-chip systems)
- Error detection and signalling
- Maximum devices: 60 (ID field 64, 4 reserved). No more than 16 recommended

LIN physical layer

- Single line wired-AND bus with pull-up resistors in every node
- Supplied by the vehicle power net (Vbat)
- Diode prevents ECU (electronic control unit) being powered from the bus in case of battery loss



LIN physical layer

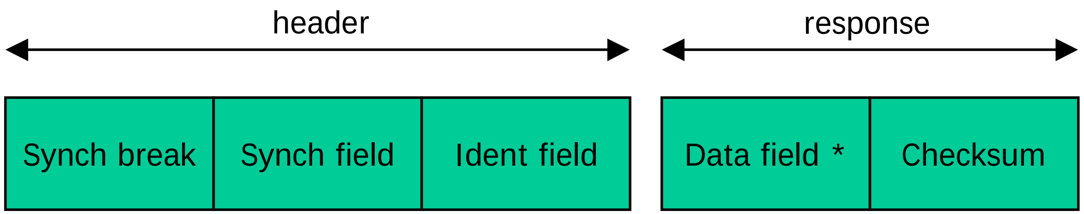
- Synchronization is done at the beginning of each message in the synch field.
- Bit Timing:
  - Bit timing of the master used

LIN data link layer

- Single master – no arbitration:
  - Only master node can send message header
  - Slave tasks respond to this header
  - Error occurs if more than one slave respond
- Safety
  - Monitoring ‚should‘ and ‚is‘ values
  - Checksum for data field
  - Double parity protection for id field
  - Errors are locally detected and provided on diagnostic request
- Fault confinement
  - Every node is able to distinguish short disturbances from permanent

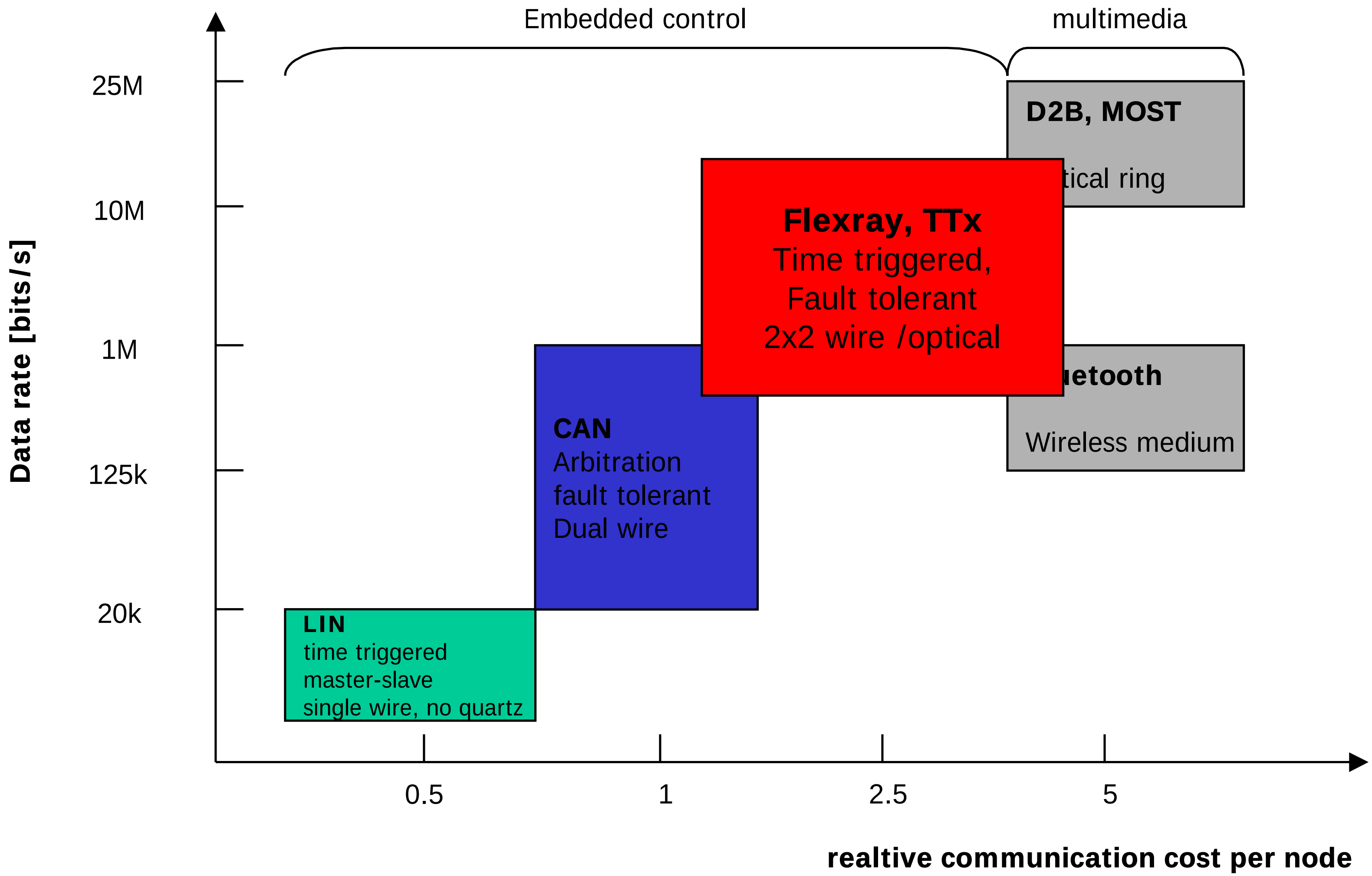
LIN data link layer

- No acknowledgement
  - In case of inconsistency the master task can change message schedule
- Message filtering by id of the message
- Message frame:



- a byte field consists of a D bit, 8 data bits and a R bit
- between header and response is an in-frame response space
- ident field has 4 id bits, 2 data length bits and 2 parity bits

Next time:


Embedded control
multimedia
Data rate [bits/s]
25M
10M
1M
125k
20k
D2B, MOST
tical ring
Flexray, TTx
Time triggered,
Fault tolerant
2x2 wire /optical
uetooth
Wireless medium
CAN
Arbitration
fault tolerant
Dual wire
LIN
time triggered
master-slave
single wire, no quartz
0.5
1
2.5
5
realtive communication cost per node