Inovace tohoto kurzu byla v roce 2011/12 podpořena projektem CZ.2.17/3.1.00/33274 financovaným Evropským sociálním fondem a Magistrátem hl. m. Prahy.

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Embedded and Real-Time Systems

Real-Time Communication

Software Substitute

- Today mechanical and electrical control systems are replaced by computer based solutions.
- Contributing causes are:
  - It is possible to improve already existing technologies, e.g., brakes in cars
  - It is possible to do things previously seemed impossible, e.g., drive-by wire, electronic stability program in cars, etc..
- But …
  - Stress on reliability and safety

Accidents

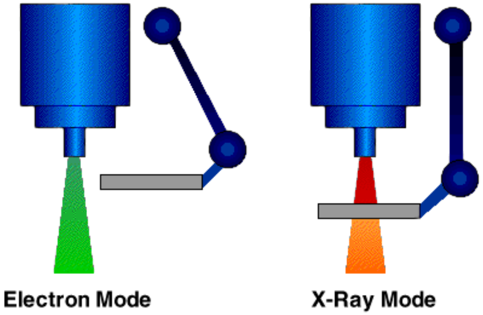
- Ariane 5
  - Exploded on June 4, 1996
    - only 39 seconds after launch
    - loss of about US$ 370 million
  - A 64-bit float was truncated to 16-bit integer in a “non-critical software component”
  - This caused unhandled hardware exception
  - The erroneous component (a method) was inherited/reused from Ariane 4 and had no practical use in Ariane 5

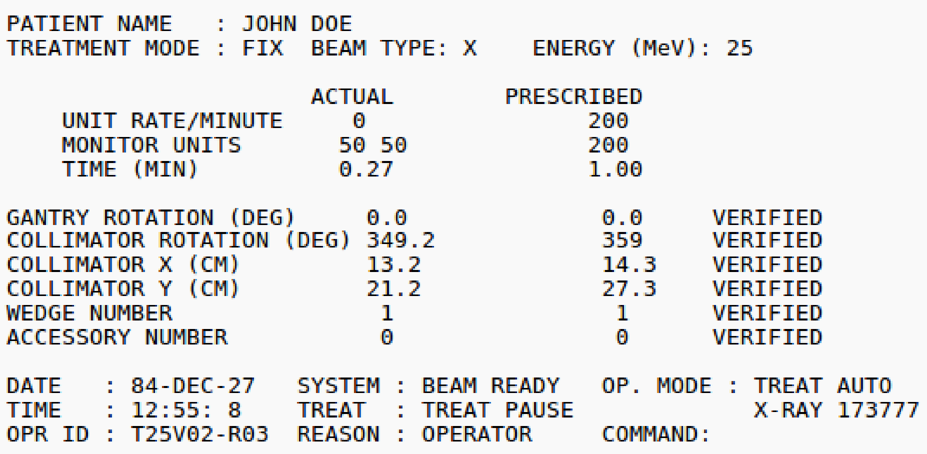
Accidents

- Patriot – Failure at Dhahran
  - February 25, 1991, an Iraqi Scud hit the barracks in Dhahran killing 28 soldiers
  - The area was protected by Patriot aerial interceptor missiles
  - Due to drift of system's internal
    - by one third of a second in 100 hours
    - amounted to miss distance of 600 meters
    - The system detected the missile but due to the time skew, it disregarded it as spurious

Accidents

- Therac-25
  - Computer controller radiation therapy machine
  - 6 accidents 1985-1987
    - three people died as the direct consequence of radiation burns
  - Race condition as the primary cause
  - Other causes included
    - Poor design, no review of the software
    - Bad man-machine interface
    - Overconfidence in the software
    - Not understanding safety
      - The software was in use previously, but different hardware design covered its flaws





```
PATIENT NAME     : JOHN DOE
TREATMENT MODE   : FIX  BEAM TYPE: X    ENERGY (MeV): 25

                       ACTUAL         PRESCRIBED
    UNIT RATE/MINUTE     0               200
    MONITOR UNITS       50 50            200
    TIME (MIN)          0.27             1.00

GANTRY ROTATION (DEG)      0.0          0.0      VERIFIED
COLLIMATOR ROTATION (DEG) 349.2         359      VERIFIED
COLLIMATOR X (CM)         13.2          14.3     VERIFIED
COLLIMATOR Y (CM)         21.2          27.3     VERIFIED
WEDGE NUMBER               1             1       VERIFIED
ACCESSORY NUMBER           0             0       VERIFIED

DATE    : 84-DEC-27   SYSTEM : BEAM READY   OP. MODE : TREAT AUTO
TIME    : 12:55: 8    TREAT  : TREAT PAUSE             X-RAY 173777
OPR ID  : T25V02-R03  REASON : OPERATOR     COMMAND:
```

Motivation

- A number of cooperating units.
- x-by-wire (e.g. drive-by-wire)
  - Mechanical and hydraulically control are replaced by digital control
  - Brakes, wheel steering, etc.

- Puts very high requirements on reliable communication!

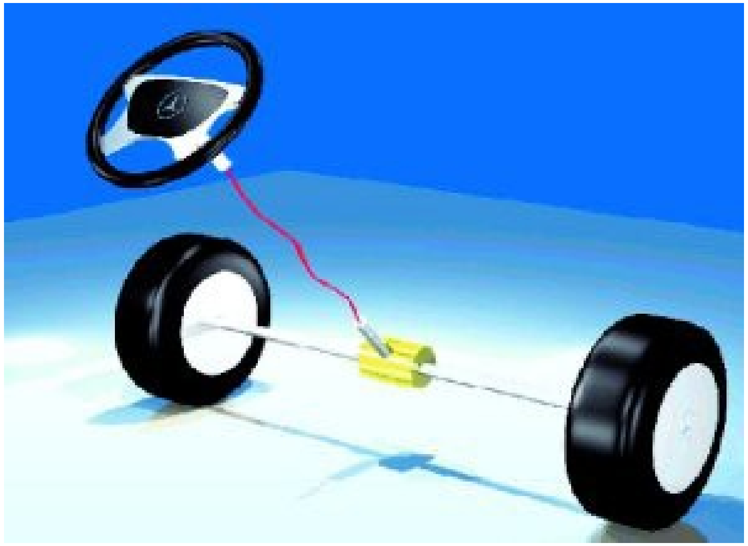


Figure taken from Issovic, D.:Real-time systems, basic course

Robust real-time communication

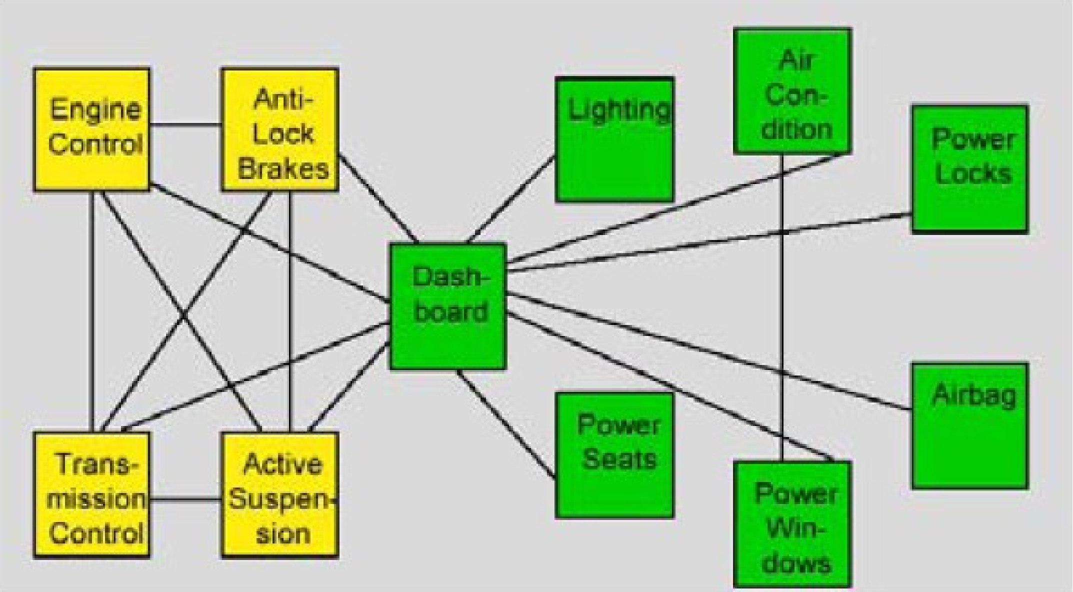
- Non real-time systems
  - Throughput
  - Average response time
  - Average latency

- Real-time systems
  - Predictability
  - Timing requirements on individual response times and latencies
  - Require predictable communications network
  - Analysis before the system is operating

- Challenge
  - to construct the computer systems that have at least as good reliability and safety as the system they are replacing

RT communication in past…

- It used to look like this…

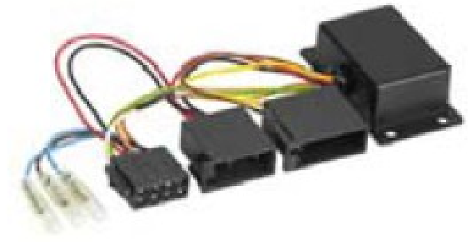


- As the number of electronic devices grew
  - the wiring gets more "messy"
  - the weight of the car increases

Figure taken from Issovic, D.:Real-time systems, basic course

Nowadays, it looks like this…

- In modern cars, point-to-point wiring is replaced by a common communication bus
  - Cost reducing
  - Flexibility
  - Functionality



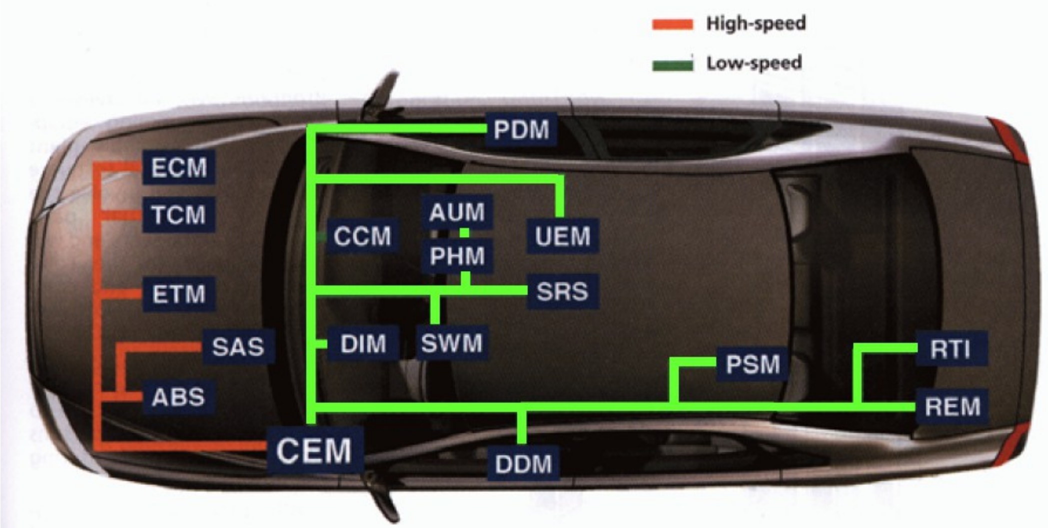


Figure taken from Issovic, D.:Real-time systems, basic course

Event-triggered communication

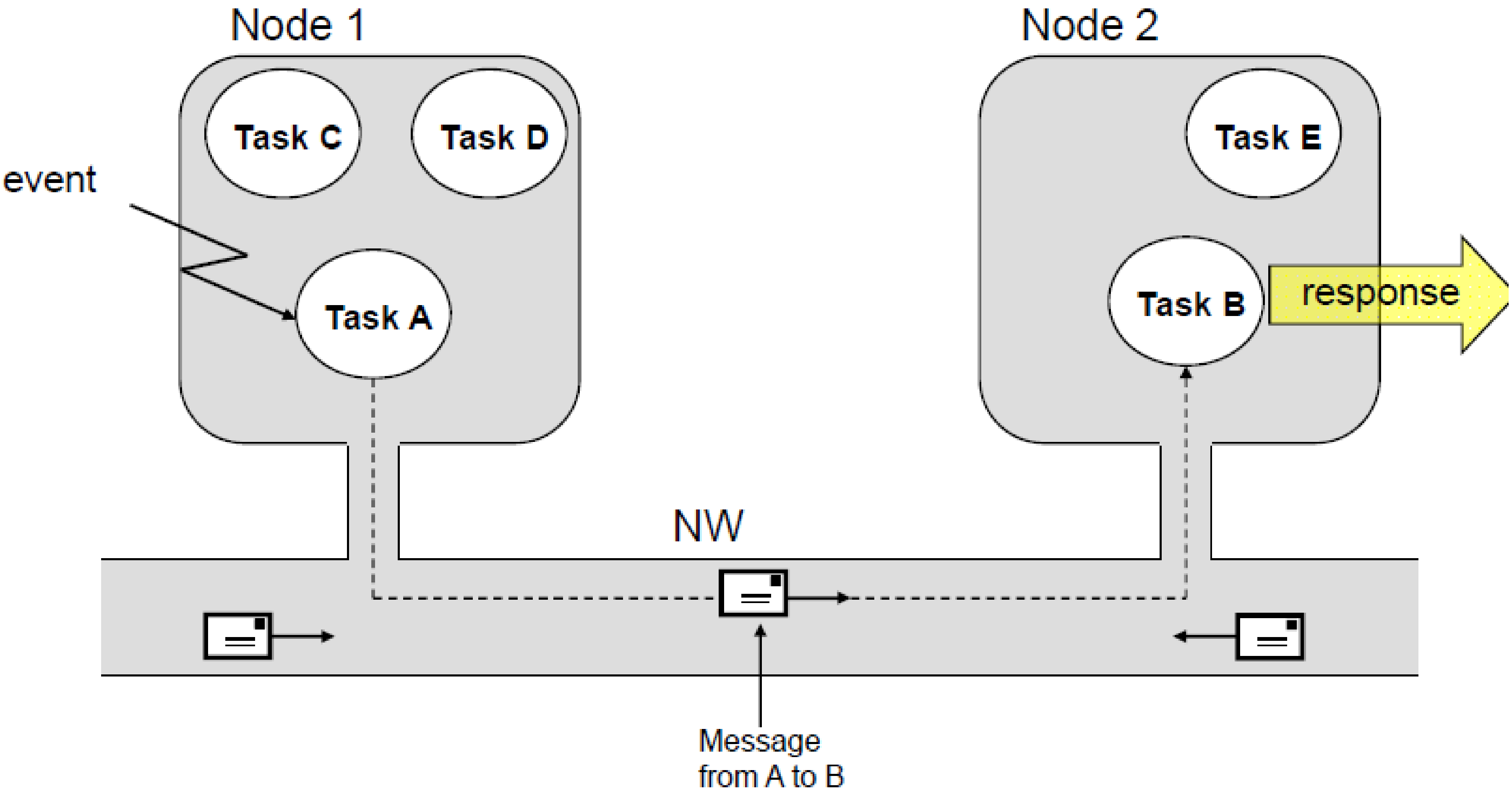


Figure taken from Issovic, D.:Real-time systems, basic course

Time-triggered communication

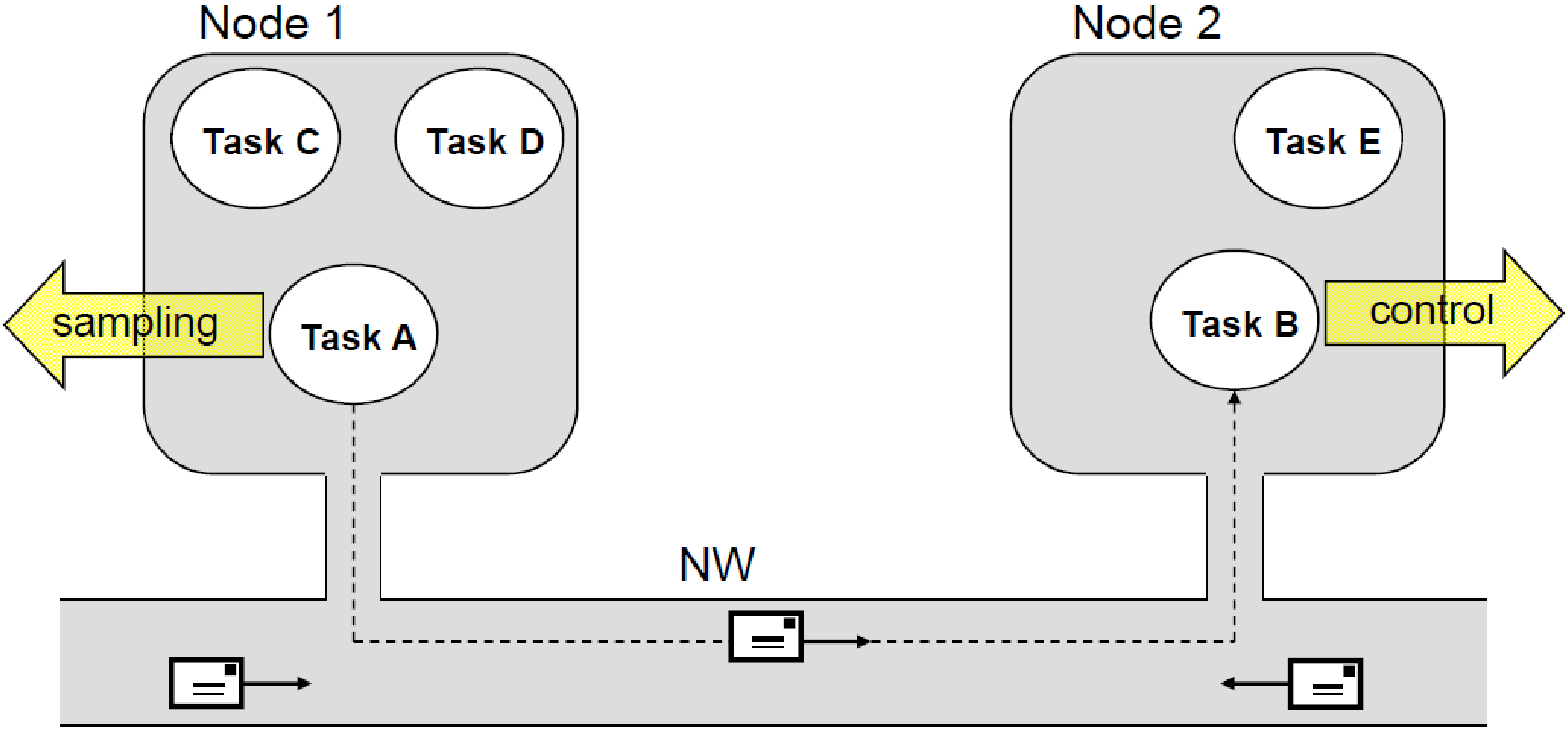


Figure taken from Issovic, D.:Real-time systems, basic course

“Ordinary” communication protocols

- Ethernet
  - Addressed broadcast messages
  - Collision → Nodes resend after a random time
  - Impossible to determine transmission times → Not suitable for hard realtime systems

- Token ring
  - Circulating token
  - No collisions
  - RT guaranties possible

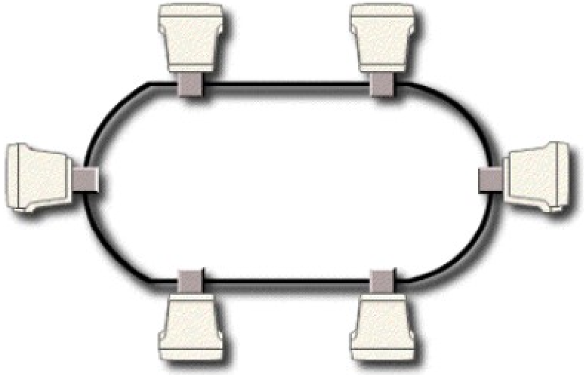
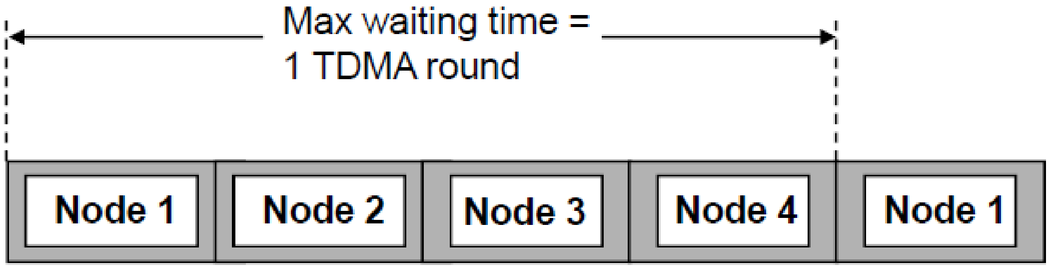


Figure taken from Issovic, D.:Real-time systems, basic course

Protocols suitable for RT-communication

- TDMA
  - Time-trigged (periodic)
  - Predictable
  - High testability
  - Example: TTP-protocol



- CSMA/CR
  - Priority based
  - Online scheduled
  - Flexible
  - Example: CAN-protocol

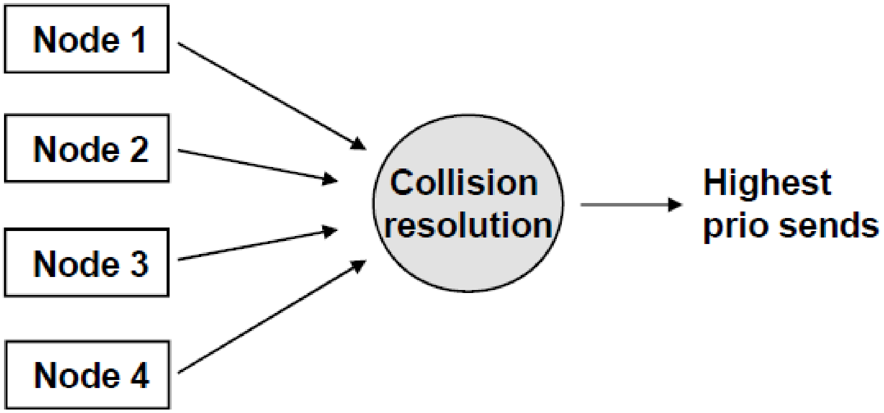


Figure taken from Issovic, D.:Real-time systems, basic course

CAN – Control Area Network

- Originally developed for automotive industry needs
  - 1983: BOSCH starts CAN development (Intel joins 1985)
  - 1987: First CAN chip
  - 1990: First car with CAN
  - 1993: ISO standard
- Now used even in industry applications
  - Very common in machinery
  - CAN-controllers by Philips, Intel, NEC, Siemens …
- An implementation of CSMA/CR
  - Priority based
  - CR is the central mechanism
  - Bitwise arbitration to resolve collisions

Structure and function

- Synchronous serial communication
  - A shared medium (cable) with connected nodes
  - Broadcast – data transmitted as frames can be picked up by all other attached nodes
  - 1 Mbit/s at 40m bus length
  - Behaves as an AND-grind: bus value = AND between all bits on the bus

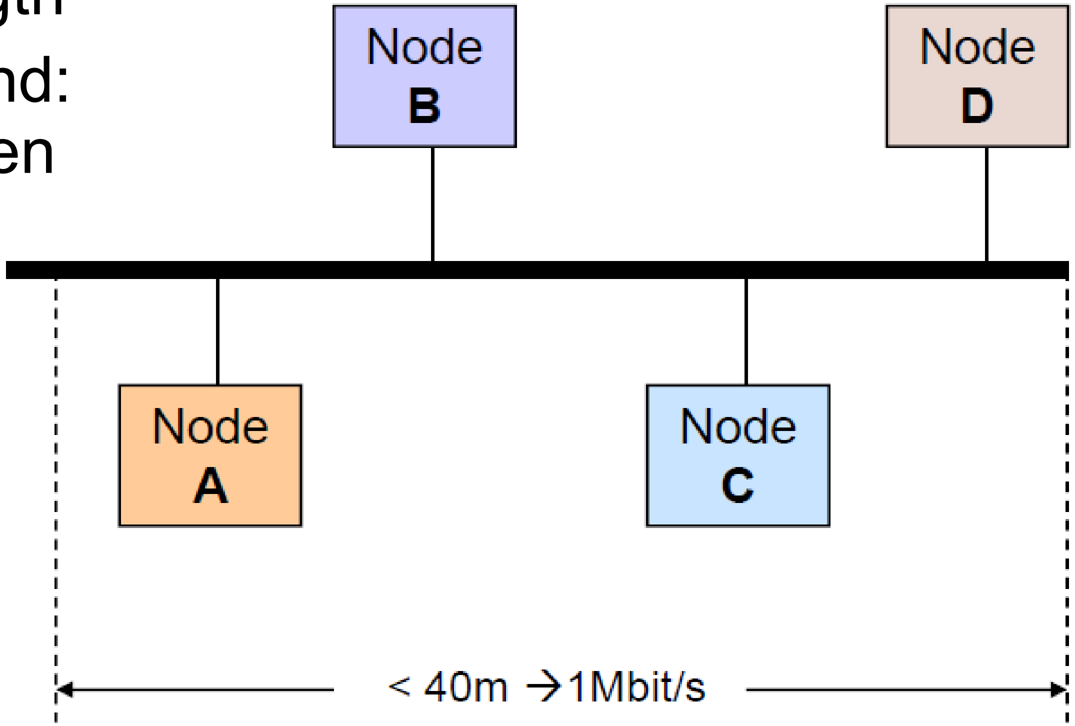


Figure taken from Issovic, D.:Real-time systems, basic course

A typical configuration

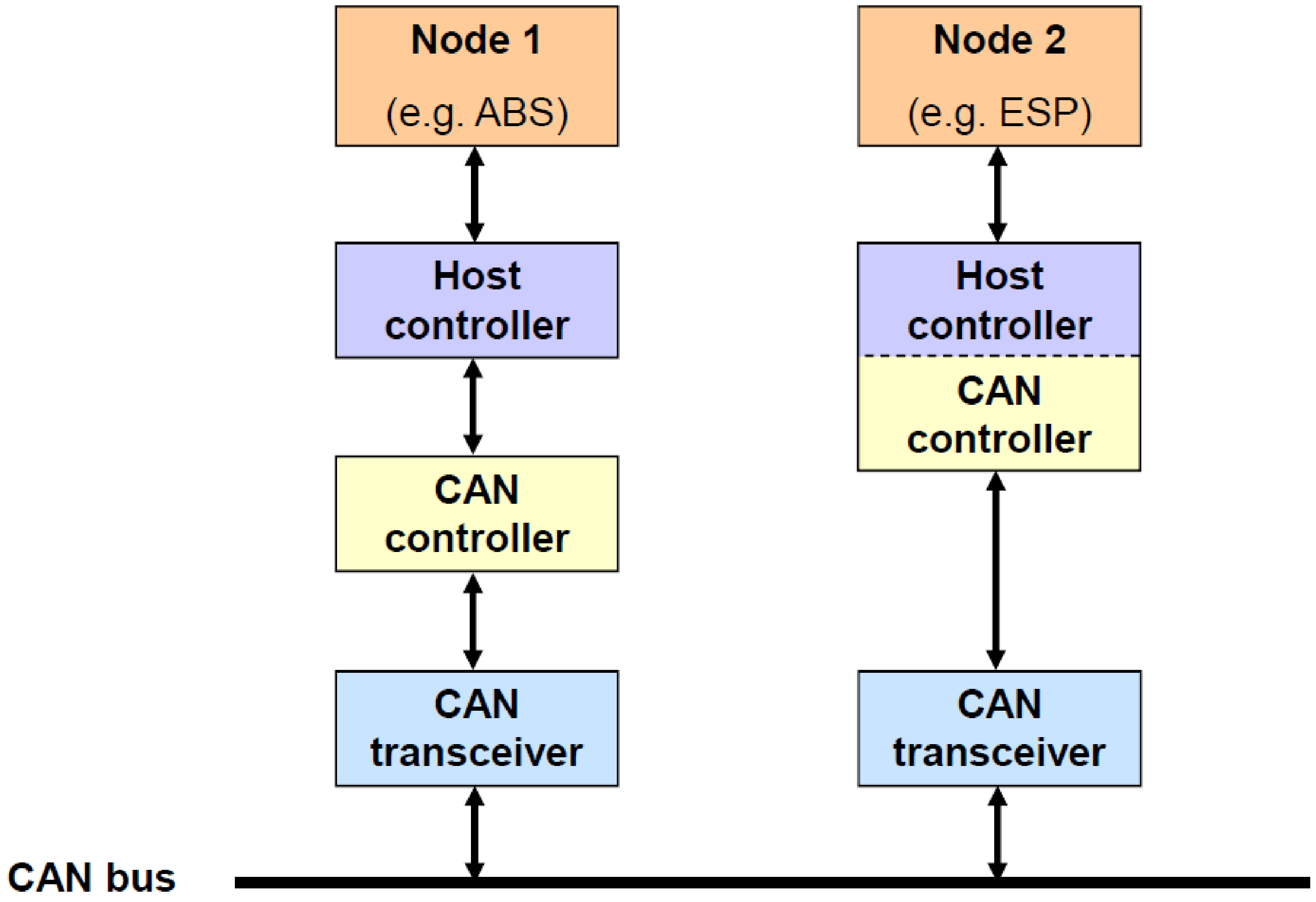
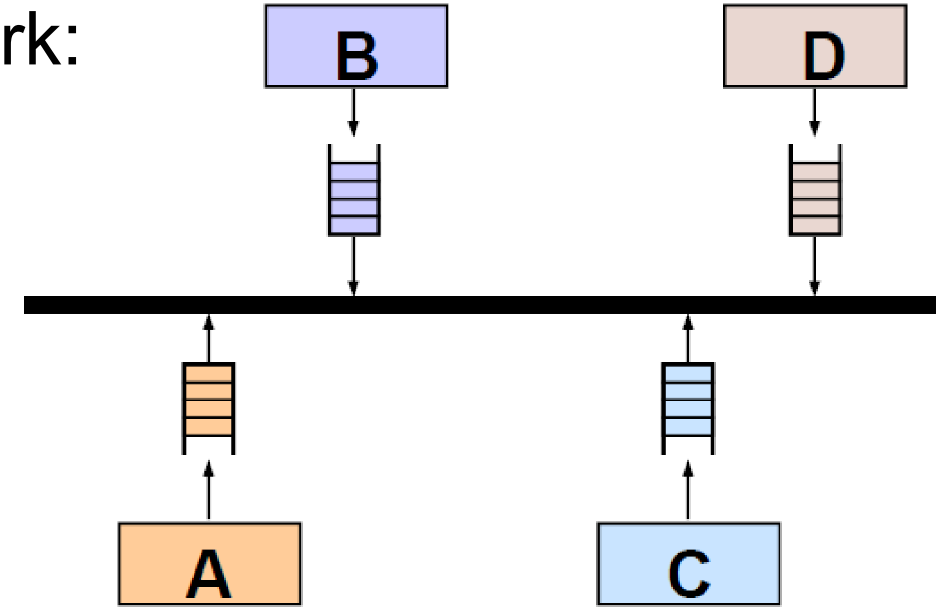


Figure taken from Issovic, D.:Real-time systems, basic course

Traffic model

- Abstraction of CAN network:
  - Frames in priority queues
  - No pre-emption



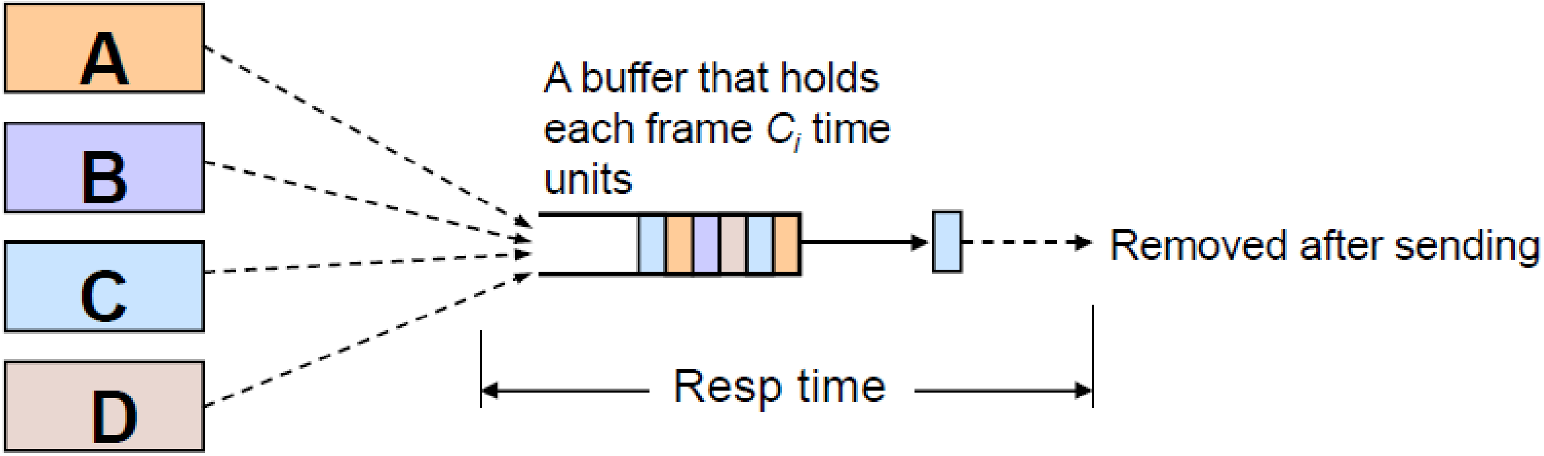


Figure taken from Issovic, D.:Real-time systems, basic course

Message format

- Data frames
  - Used for data transmission e.g., sampling values from a sensor
  - Standard CAN frame (CAN 2.0 A), 11 bits identifier
  - Extended CAN frame (CAN 2.0 B), 29 bits identifier
- Remote Frames
  - Used for information requests.
  - The transmitting node is asking for information of the type given by the identifier.
- Error frames
  - Used for error signaling
- Overload Frames
  - Used to delay the transmission of the next message frame
  - The node sending the Overload Frame is not ready to receive additional messages at this time

CAN-frame (version 2.0 A, standard format)

SOF	ID	RTR	Control	Data	CRC	CRC DEL	ACK	ACK DEL	EOF	IFS
1 bit	11 bits	1 bit	6 bits	0-8 bytes	15 bits	1 bit	1 bit	1 bit	7 bits	min 3 bits

SOF - *Start of Frame*, start bit (always 0), used for signaling that a frame will be sent (the bus must be free)
ID - ***Identifier*, identity for the frame and its priority**
RTR - *Remote Transmission Request*
Control - indicates the length of the data field
Data - **message data**
CRC - *Cyclic Redundancy Check,*
CRC DEL - *CRC delimiter*
ACK - *Acknowledgement*
ACK DEL - *ACK delimiter*
EOF - *End of Frame*
IFS - *Inter Frame Space*, resending wait time

Figure taken from Issovic, D.:Real-time systems, basic course

Arbitration mechanism

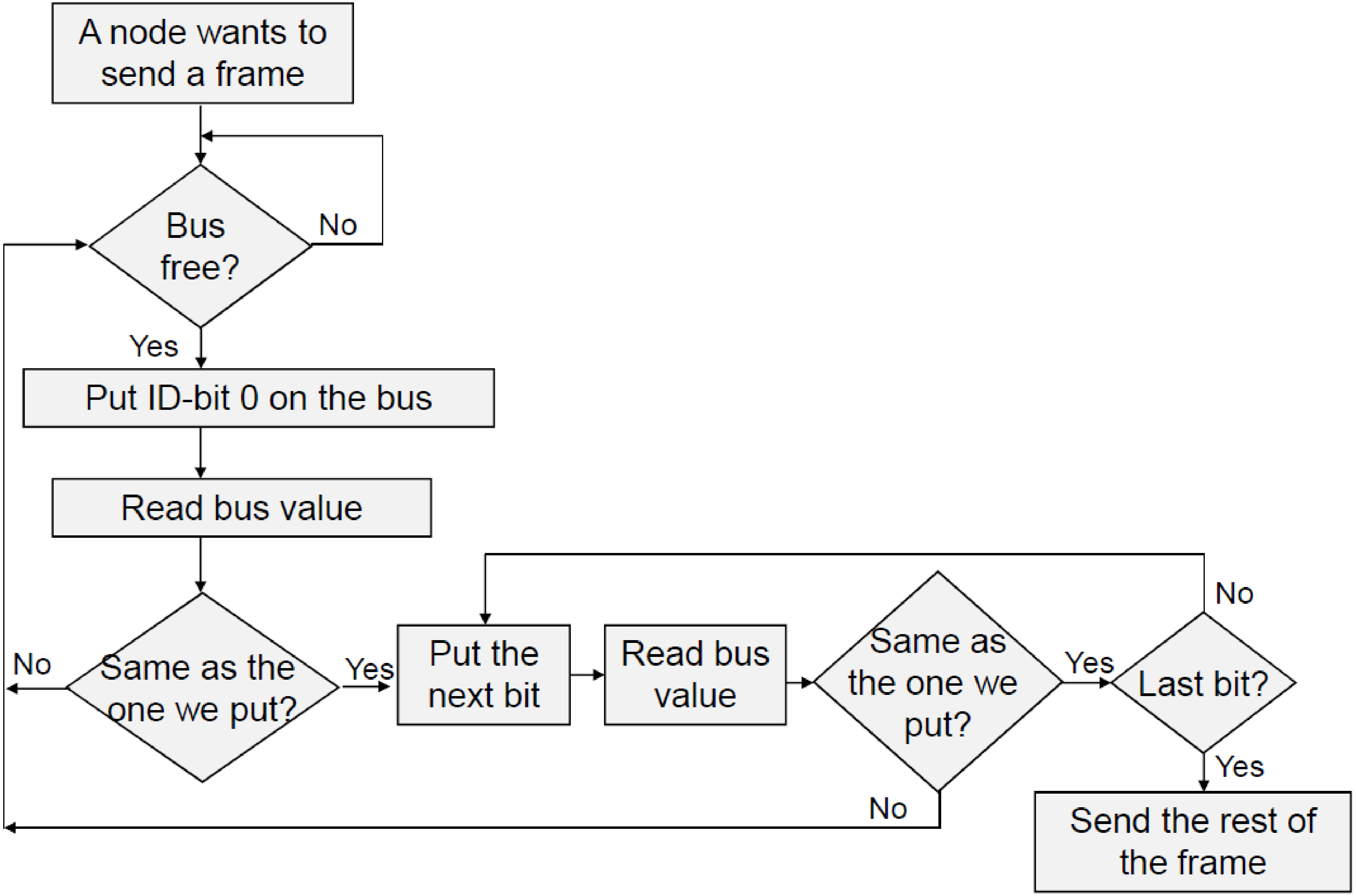
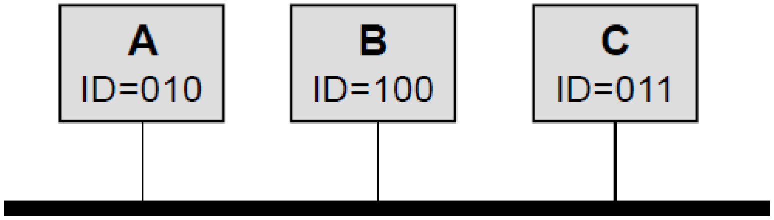


Figure taken from Issovic, D.:Real-time systems, basic course

Arbitration mechanism

- Example:
  - Assume a simplified CAN-system with three ID-bits and nodes A, B, C:



000 – highest priority

111 – lowest priority

which gives:

A-high prio, C-middle, B-low

  - How does the arbitration look like if the nodes are sending simultaneously?

Node	ID	Bit 0	Bit 1	Bit 2	
A	010	0	1	0	→ Send the rest of the frame
B	100	1	→ abort! (bit 0 ≠ bus value)		
C	011	0	1	1	→ abort! (bit 2 ≠ bus value)
Bus value:		0	1	0	

Figure taken from Issovic, D.:Real-time systems, basic course

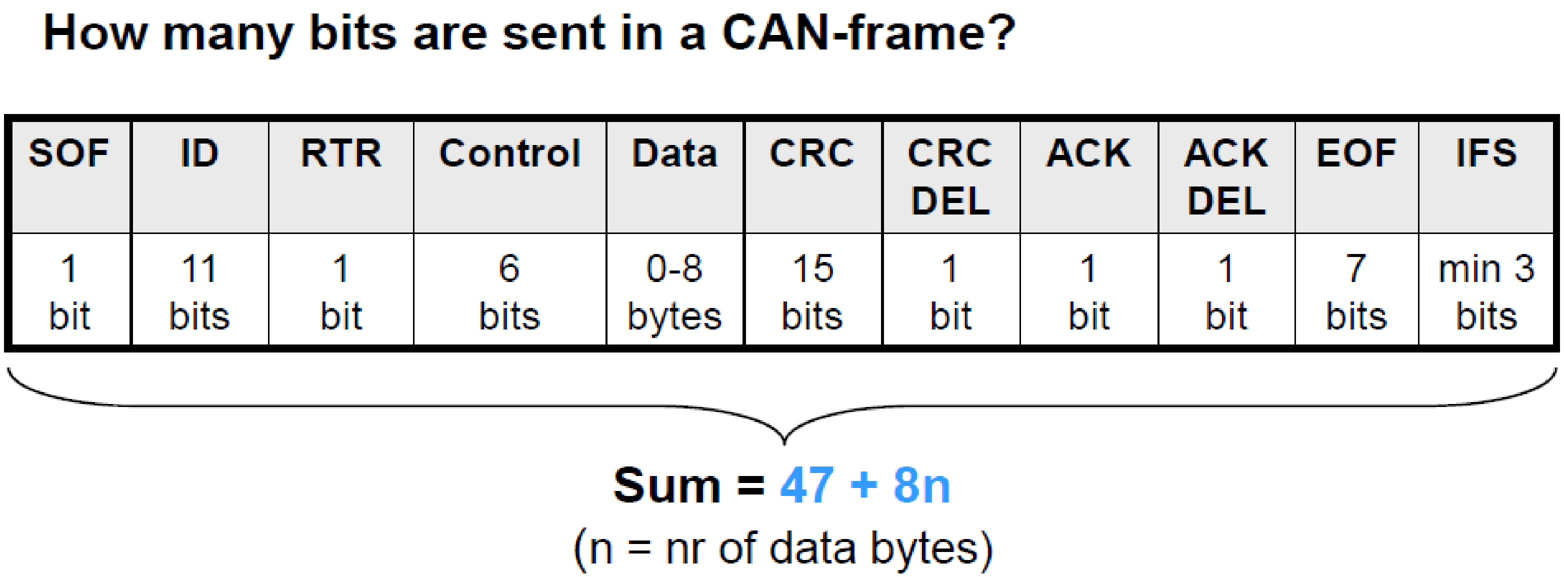
Error handling

- Error detection with check sum (CRC)
  - If the frame is received correctly, the ACK-bit is set to 0

- Error signaling
  - The node that detects an error puts instantly 000000 on the bus
  - Because zero is the dominant value, all nodes will detect the error rapidly
  - Some CAN-systems have one as the dominant bit → bit-pattern for error signaling is 111111

Timing properties

- CAN is time deterministic
  - The latency can be predicted
  - Possible to calculate how long time it takes to deliver a frame



How many bits are sent in a CAN-frame?

SOF	ID	RTR	Control	Data	CRC	CRC DEL	ACK	ACK DEL	EOF	IFS
1 bit	11 bits	1 bit	6 bits	0-8 bytes	15 bits	1 bit	1 bit	1 bit	7 bits	min 3 bits

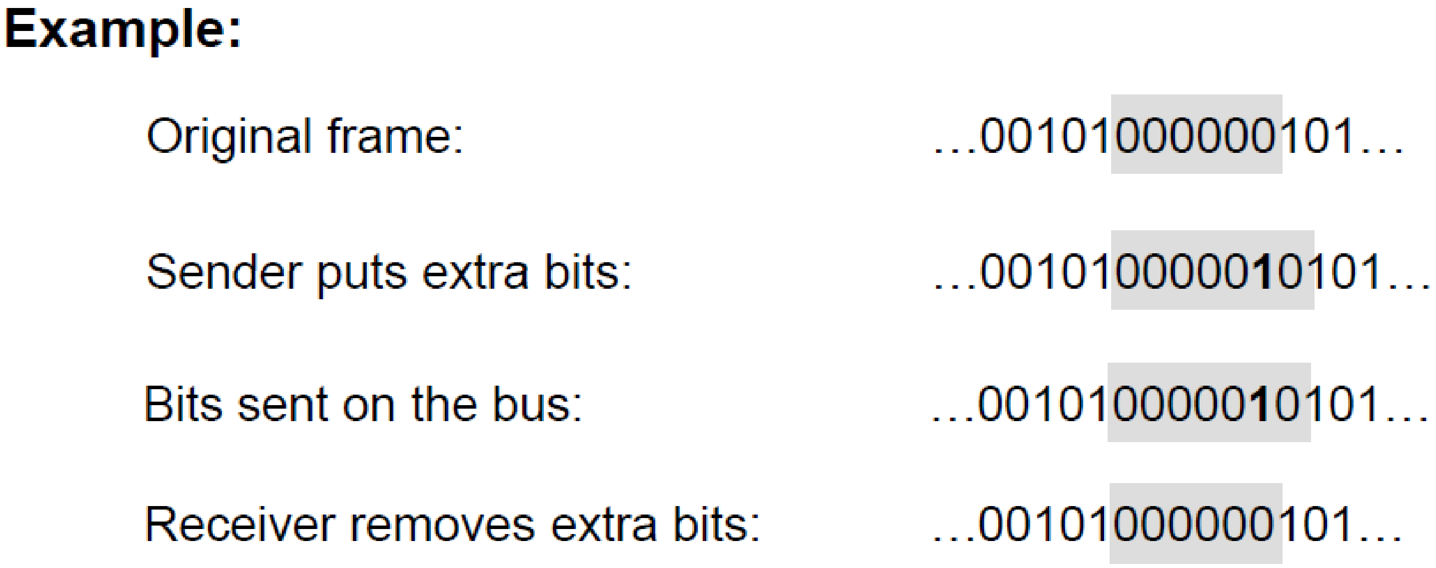
Sum = 47 + 8n

(n = nr of data bytes)

Figure taken from Issovic, D.:Real-time systems, basic course

Timing properties

- We must avoid two bit-patterns that are used for error signaling – i.e., 000000 and 111111:
  - Bit stuffing: the sender puts extra bits on strategic places to prevent forbidden bit-patterns
  - The receiver reconstruct the original frame by removing the extra bits



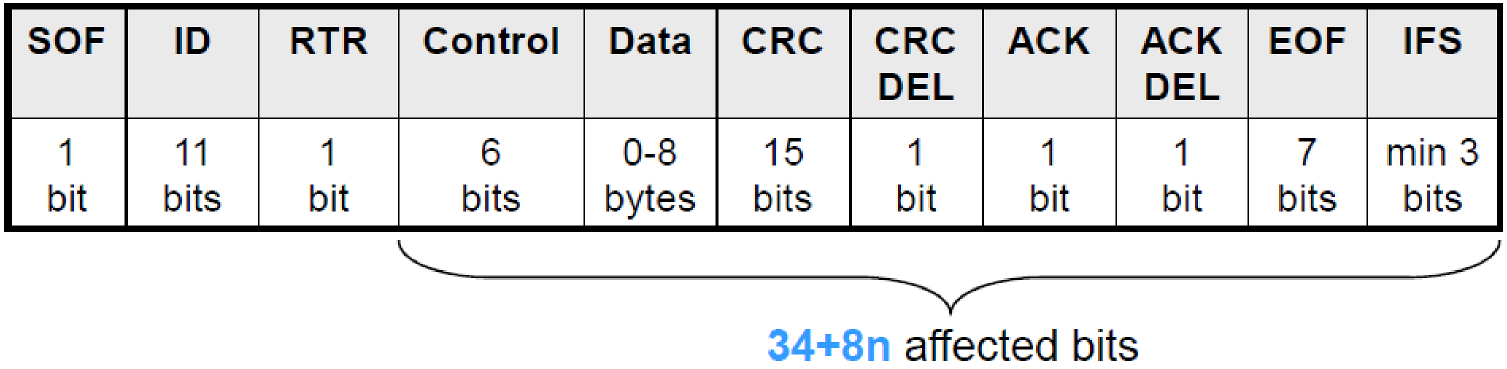
Example:

Original frame:	…00101000000101…
Sender puts extra bits:	…0010100000**1**0101…
Bits sent on the bus:	…0010100000**1**0101…
Receiver removes extra bits:	…00101000000101…

Figure taken from Issovic, D.:Real-time systems, basic course

Timing properties

- Do we need to perform bit stuffing on all 47+8n bits?
  - No, only 34 (of 47) control bits are affected
  - By forbidding some ID values we can avoid bit stuffing in the frame ID



SOF	ID	RTR	Control	Data	CRC	CRC DEL	ACK	ACK DEL	EOF	IFS
1 bit	11 bits	1 bit	6 bits	0-8 bytes	15 bits	1 bit	1 bit	1 bit	7 bits	min 3 bits

34+8n affected bits

Figure taken from Issovic, D.:Real-time systems, basic course

Timing properties

- The standard allows both 000000 and 111111 for error signaling.
  - To avoid forbidden bit patterns we must insert an extra bit after the first five bits and one extra bit after each fourth original bit.

- Example
  - Original: 1111 1000 0111 1000 0111 1
  - After bitstuffing: 1111 1**0**000 0**1**111 1**0**000 0**1**111 1

1111 1**0**	000 0**1**	111 1**0**	000 0**1**	111 1
Extra bit after 5 original bits	Extra bit after 4 original bits	Extra bit after 4 original bits	Extra bit after 4 original bits	etc...

Timing properties

Hence, the number of extra bits in a CAN-frame is:

$$\left\lfloor \frac{34 + 8n - 1}{4} \right\rfloor$$

Now we can calculate the total transmission time for a CAN-frame:

$$C_i = (47 + 8n + \left\lfloor \frac{34 + 8n - 1}{4} \right\rfloor \tau_{bit}$$

$n_{max} = 8$ and 1Mbit/s, and $\tau_{bit} = 1\mu s$ (bus speed) gives:

$$C_i = (47 + 8 * 8 + \left\lfloor \frac{34 + 8 * 8 - 1}{4} \right\rfloor 1\mu s = 135\mu s$$

Response time analysis for CAN

- CAN is priority based, non-preemptive
  - Once a frame has managed to send the first bit, it will continue sending the rest uninterrupted

Reponse time for a frame i:

$$w_i = \tau_{bit} + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i}{T_j} \right\rceil C_j$$

$$R_i = w_i + C_i - \tau_{bit}$$

Where the blocking time for a frame is given by:

$$B_i = \max_{\forall k \in lp(i)} C_k \leq 135\tau_{bit}$$

$hp(i)$ … high priority frames (that can delay the first bit)

$lp(i)$ … low priority frames (that can block the first bit)

Response time analysis for CAN

- Even frames can have jitter:
  - variations in time when a frame is queued
  - usually due to the sender task's jitter

$$w_i = \tau_{bit} + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i + J_j}{T_j} \right\rceil C_j$$

$$R_i = J_i + w_i + C_i - \tau_{bit}$$

  - The equations above can be re-written as:

$$w_i = B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i + J_j + \tau_{bit}}{T_j} \right\rceil C_j$$

$$R_i = J_i + w_i + C_i$$

CAN – Example

- Assumptions:
  - Dominant bit: 0
  - Bus speed:1 Mbit/s
  - Task instances send their messages at the end of the execution
  - The size of each message is 135 bits
  - Task priority assignment is according to Rate Monotonic

Node 1 (id=011)

Task	T	C	Msg
A1	10000	3000	m1
A2	7000	1000	-

Node2 (id=001)

Task	T	C	Msg
B1	5000	1000	m2
B2	4000	1000	-

Node 3 (id=000)

Task	T	C	Msg
C1	4000	1000	m3
C2	10000	1000	-

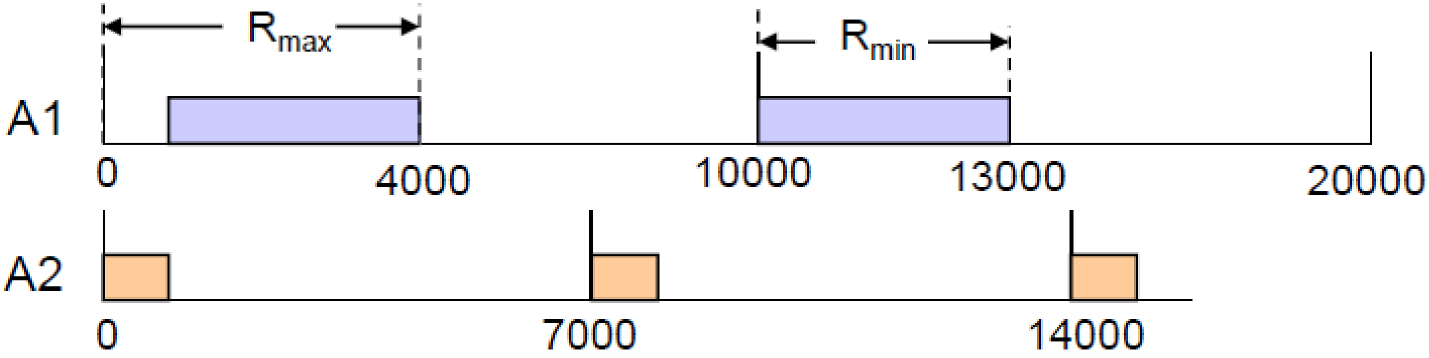
a) Calculate jitter for the messages

b) Calculate response times for the messages

Figure taken from Issovic, D.:Real-time systems, basic course

CAN – Example (Jitter)

Node 1:



$$J_{m1} = R_{max}(A1) - R_{min}(A1) = 4000\text{-}3000 = 1000$$

Node 2:

(Same as above)

$$J_{m2} = R_{max}(B1) - R_{min}(B1) = 2000\text{-}1000 = 1000$$

Node 3:

$$J_{m3} = R_{max}(C1) - R_{min}(C1) = 0$$

(Note! No jitter, C1 has high prio)

Figure taken from Issovic, D.:Real-time systems, basic course

CAN – Example (Response time)

m3: lp(m3)={ m1,m2} → B(m3)=max(C_{m1}, C_{m2})=max(135, 135)=135

$$w_{m3} = B_{m3} + 0 = 135\mu s \qquad R_{m3} = J_{m3} + w_{m3} + C_{m3} = 0 + 135 + 135 = 270\mu s$$

m2: lp(m2)={ m1} → B(m2)=C_{m1}=135

$$w_{m2}^{0} = 0 \qquad w_{m2}^{1} = B_{m2} + \left\lceil \frac{w_{m2}^{0} + J_{m3} + \tau_{bit}}{T_{m3}} \right\rceil C_{m3} = 135 + \left\lceil \frac{0+0+1}{4000} \right\rceil 135 = 270$$

$$w_{m2}^{2} = 135 + \left\lceil \frac{270+0+1}{4000} \right\rceil 135 = 270 \qquad R_{m2} = J_{m2} + w_{m2} + C_{m2} = 1000 + 270 + 135 = 1405\mu s$$

m1: lp(m1)={ } → B(m1)=0

$$w_{m1}^{0} = 0$$

$$w_{m1}^{1} = B_{m1} + \left\lceil \frac{w_{m1}^{0} + J_{m2} + \tau_{bit}}{T_{m2}} \right\rceil C_{m2} + \left\lceil \frac{w_{m1}^{0} + J_{m3} + \tau_{bit}}{T_{m3}} \right\rceil C_{m3} = 270$$

$$w_{m1}^{2} = 0 + \left\lceil \frac{270+1000+1}{5000} \right\rceil 135 + \left\lceil \frac{270+0+1}{4000} \right\rceil 135 = 270$$

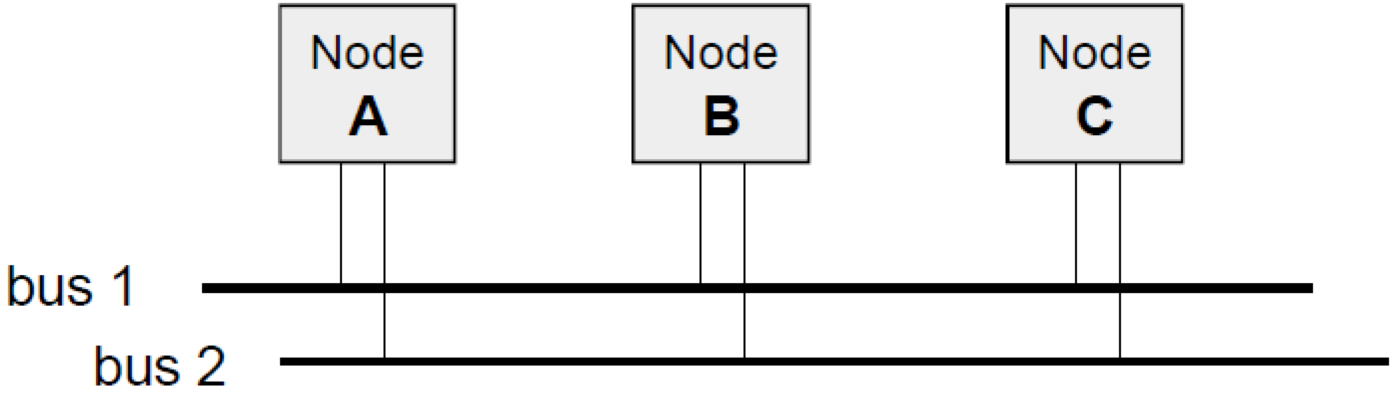
$$R_{m1} = J_{m1} + w_{m1} + C_{m1} = 1000 + 270 + 135 = 1405\mu s$$

Figure taken from Issovic, D.:Real-time systems, basic course

TTP – Time Triggered Protocol

- An implementation of TDMA
  - Time-trigged
  - Bus access is pre-defined in an offline schedule
  - Nodes can be assigned several slots
- Originally developed on Technical University of Vienna in
- Corporation with several car manufacturers
  - Commercial development by TTTech
- Aimed for X-by-wire applications
  - Boeing 777, Airbus 340, Audi,…
- Very high demands on reliability
  - Safety-critical real-time systems that require fault tolerance

TTP - typical system configuration



- Fail Silent nodes
  - Nodes detect errors by themselves
  - They either deliver correct result or no result at all
- Grouped in FTUs (Fail Tolerant Unit)
  - Several nodes that do the same in parallel
  - FTU:n is working as long one of the nodes is working

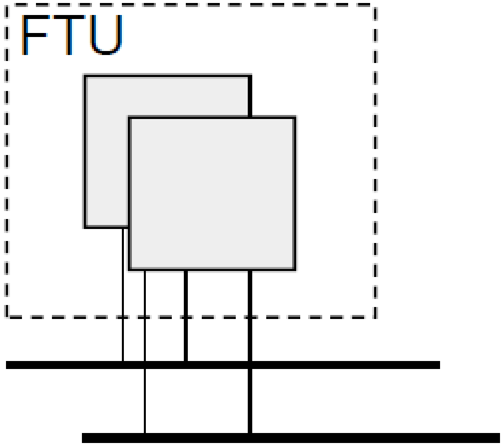


Figure taken from Issovic, D.:Real-time systems, basic course

TTP – Synchronization

- Time-trigged → clocks must be synchronized
  - continuous synchronization
  - some tens of microseconds

- The receiver compares actual receiving time with expected receiving time

CAN vs. TTP

- TTP
  - Time-trigged (periodic)
  - Easier analysis
  - Predictable
  - High testability
- CAN
  - Priority based
  - Faster response times for high priority messages
  - Flexible

FlexRay

- Started in year 2000 as an industrial consortium:
  - BMW, Daimler-Chrysler, Philips and Motorola.
  - Today, more than 100 members world wide.
- Goals and properties
  - High speed, an order of magnitude higher than CAN (10Mbps)
  - Deterministic communication
  - Fault-tolerant communication
  - Different connection possibilities

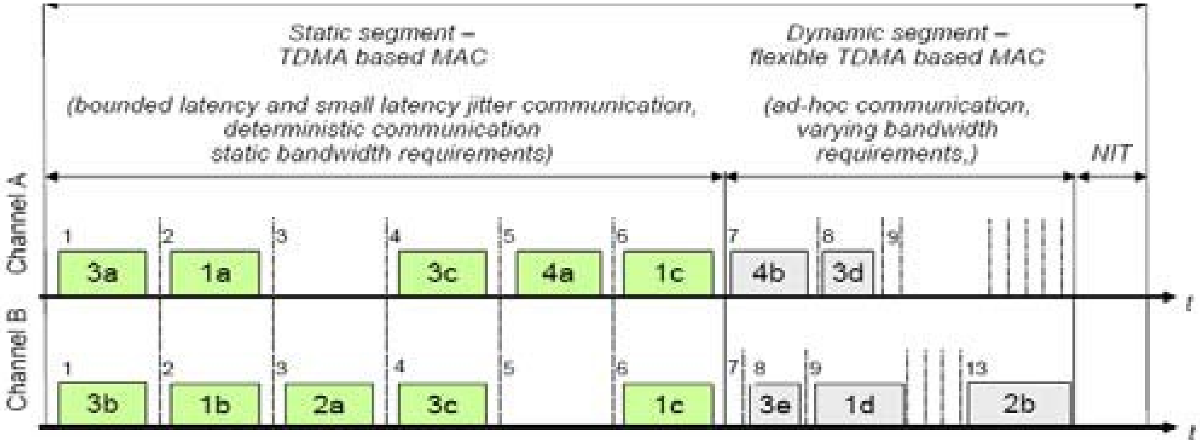
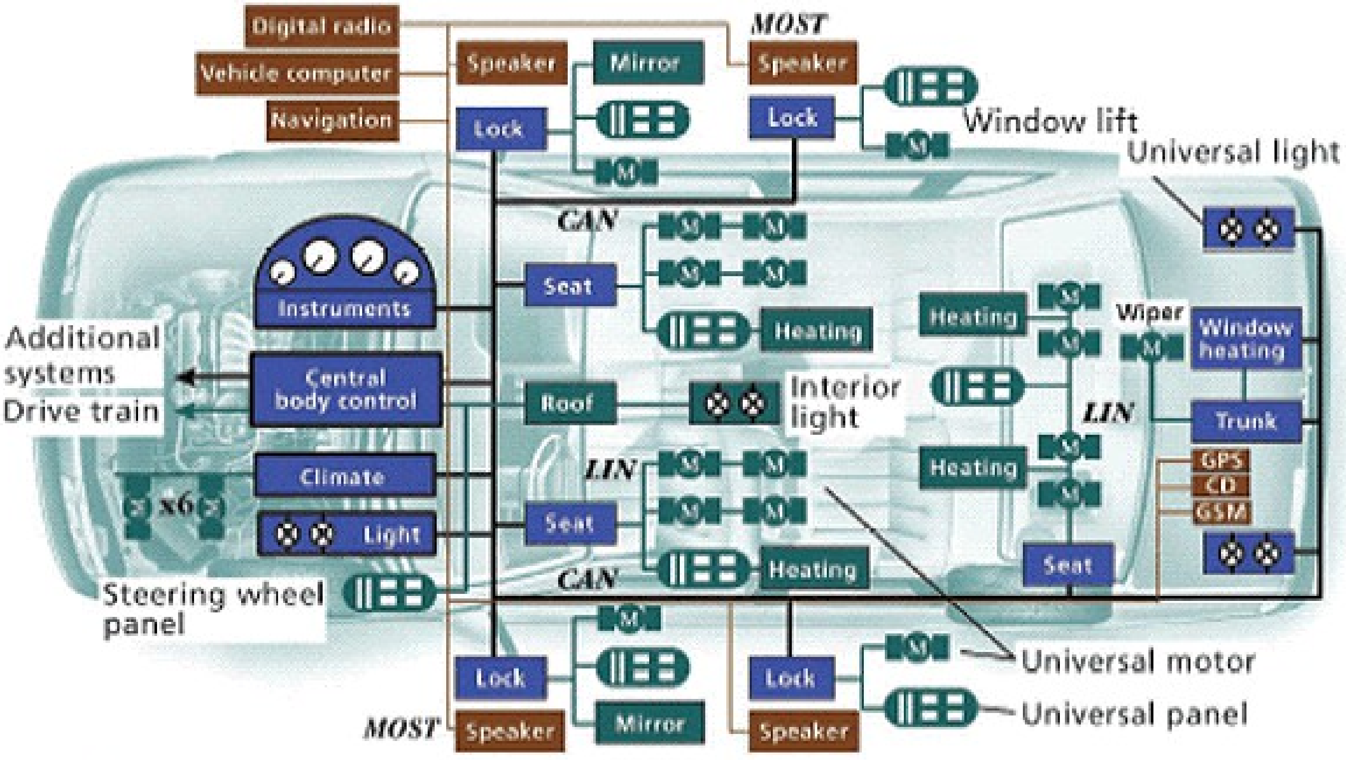


Figure taken from Issovic, D.:Real-time systems, basic course

State of the art

- Combination of different buses



CAN	Controller area network
GPS	Global Positioning System
GSM	Global System for Mobile Communications
LIN	Local interconnect network
MOST	Media-oriented systems transport

Figure taken from Issovic, D.:Real-time systems, basic course