Accessing the shared communication medium

Media
Access
Conntrol

Network-topologies

- Bus
- Ring
- Star
- Tree
- grid, mesh
- fully connected

assessment criteria:
Overhead, latency, tolerance of transmission errors and network partitions

What are the impairments of predicatbilty ?
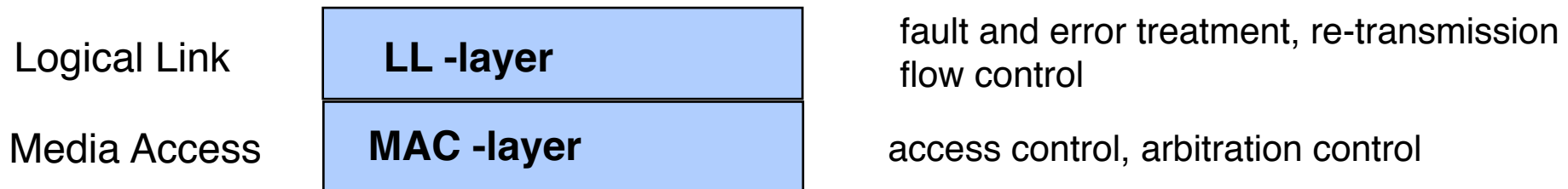
Load

Failures

Predictability

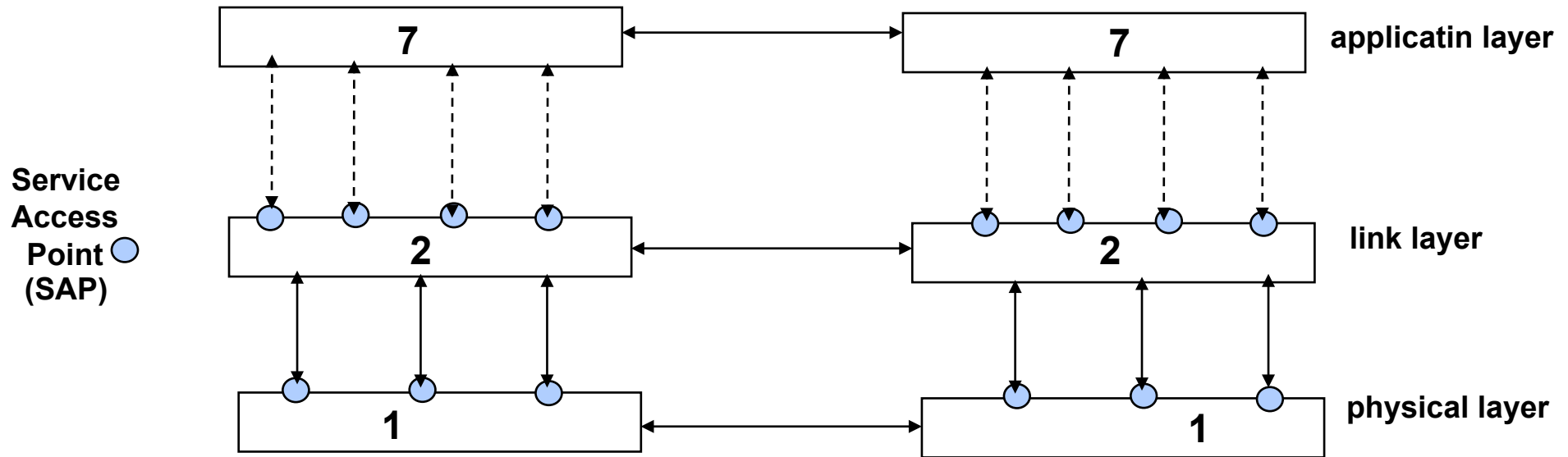Network contention
Arbitration conflicts

transmission errors,
lost messages

Media Access Control & Logical Link Layer

transfer of data blocks
flow control
fault and error handling
message re-transmissions

Logical Link	**LL -layer**
Media Access	**MAC -layer**

fault and error treatment, re-transmission
flow control

access control, arbitration control

Common Layering in the fieldbus area

Service Access Point (SAP) ○

7 — applicatin layer

2 — link layer

1 — physical layer

➡ **Assumption: homogeneous, closed system**

➡ **Not all layers are necessary (e.g. routing)**
Empty layers in the ISO/OSI- model

➡ **Higher layers directly access the SAPs of lower layers.**

➡ **Efficiency improvement**

➡ **Direct mapping of layer 7 services to layer 2 functionality.**

MAC-protocols

controlled access

random access

Collision avoidance

Collision resolution

Reservation-based

Token-based

Time-based

Master-Slave

Priority-based

probabilistic

dynamic   static

ATM

TDMA:

TTP,

Maruti

Token-Ring

Token-Bus

Timed Token Protocol

CSMA/CA :
Collision Avoidance

IEEE 802.11
P-persistent CSMA

LON, VTCSMA

ProfiBus DP
FIP
CAN-Open

CSMA/CA :
Consistent Arbitration

CAN

CSMA/CD :
Carrier Sense Multiple Access /
Collision Detection

Ethernet

Predictability in random access networks:

probabilistic

very low overhead and latency in low load conditions
very flexible wrt. extensibility
thrashing in high load situations

Collsion avoidance

balances the latency against the collision probability

maintains a good average throughput in medium load situations

may adapt to high load conditions

Consistent arbitration with Collision Resolution

needs support from the physical layer

maintains a constant throughput in all load conditions

supports sophisticated fault handling

Controlled Access by Collision Exclusion:

Master/Slave

all control information in one place

maximum of control

easy to change

Global Time

Easy temporal co-ordination

Minimal communication overhead

Token-based

Decentralized mechanism

Integration of critical and non-critical messages

CAN-Bus
Controller Area Network

CAN Milestones

Year	Milestone
1983	Start of the Bosch internal project to develop an in-vehicle network
1986	Official introduction of CAN protocol
1987	First CAN controller chips from Intel and Philips Semiconductors
1991	Bosch's CAN specification 2.0 published
1991	CAN Kingdom CAN-based higher-layer protocol introduced by Kvaser
1992	CAN in Automation (CiA) international users and manufacturers group established
1992	CAN Application Layer (CAL) protocol published by CiA
1992	First cars from Mercedes-Benz used CAN network
1993	ISO 11898 standard published
1994	1st international CAN Conference (iCC) organized by CiA
1994	DeviceNet protocol introduction by Allen-Bradley
1995	ISO 11898 amendment (extended frame format) published
1995	CANopen protocol published by CiA
2000	Development of the time-triggered communication protocol for CAN (TTCAN)

http://www.can-cia.org/can/protocol/history/index.html

The CAN Standard

Developed by BOSCH,   http://www.semiconductors.bosch.de/pdf/can2spec.pdf

CAN Specification 1.2
CAN Specification 2.0

Difference between the specifications mainly is:
• the different lenth of message identifiers (CAN-ID)
Standard CAN: 11 Bit IDs (defined in CAN 2.0 A ← 1.2)
Extended CAN: 29 Bit IDs (defined in CAN 2.0 B)

CAN-Controller Implementations:
Basic CAN: 1 Transmit + 1 Receive (Shadow) Buffer
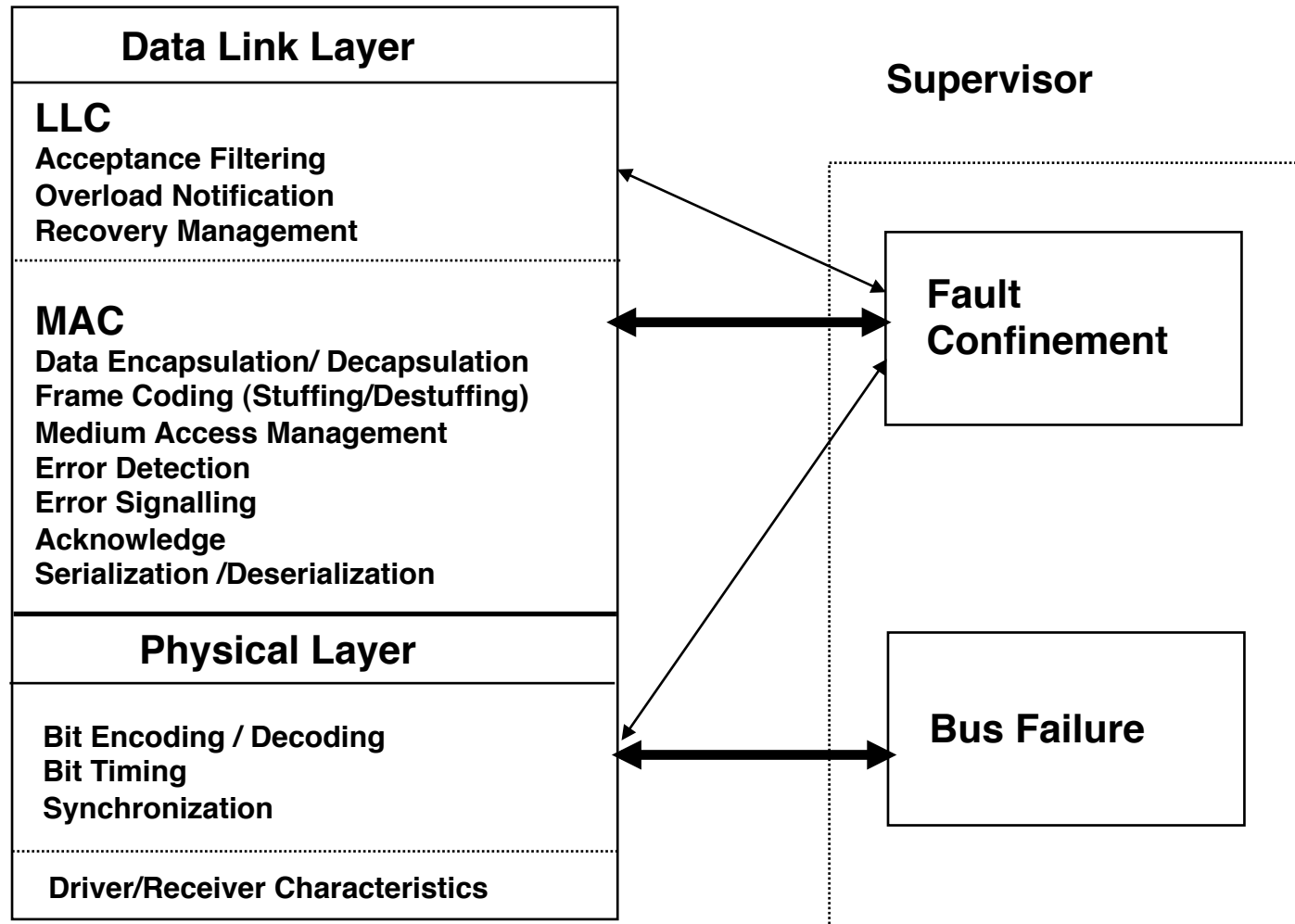**Extended CAN**: 16 Configurable Transmit/Receive Buf.

Basic CAN properties

- **Prioritised messages**

- **Bounded and guaranteed message delay for the highest priority message.**

- **Constant troughput in all load situations**

- **Error detection and signalling in the nodes.**

- **Automatic re-transmission.**

- **Fail silent behaviour of nodes.**

- **Consistent message delivery.**

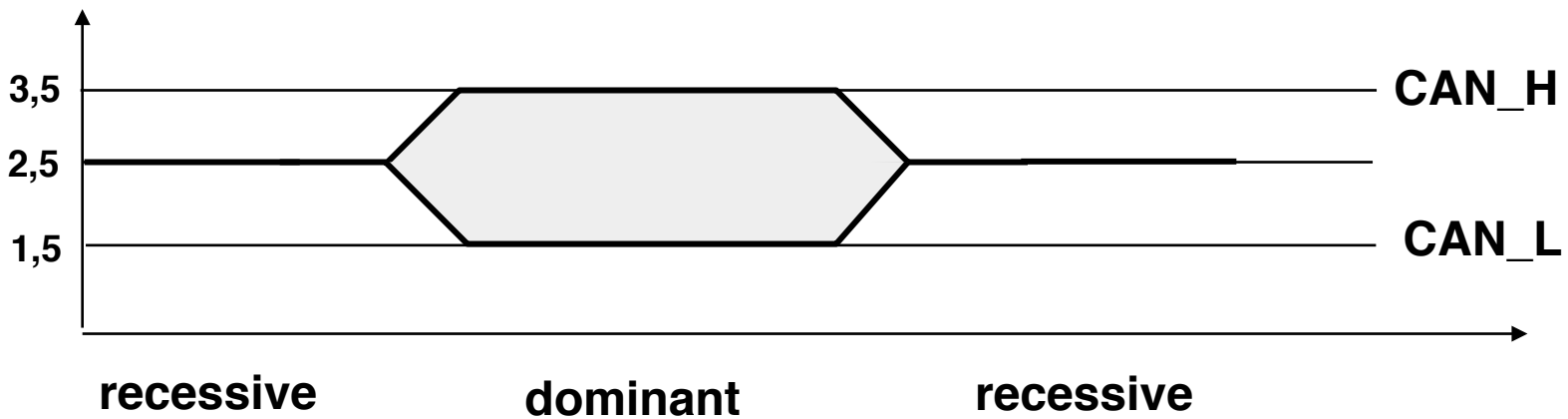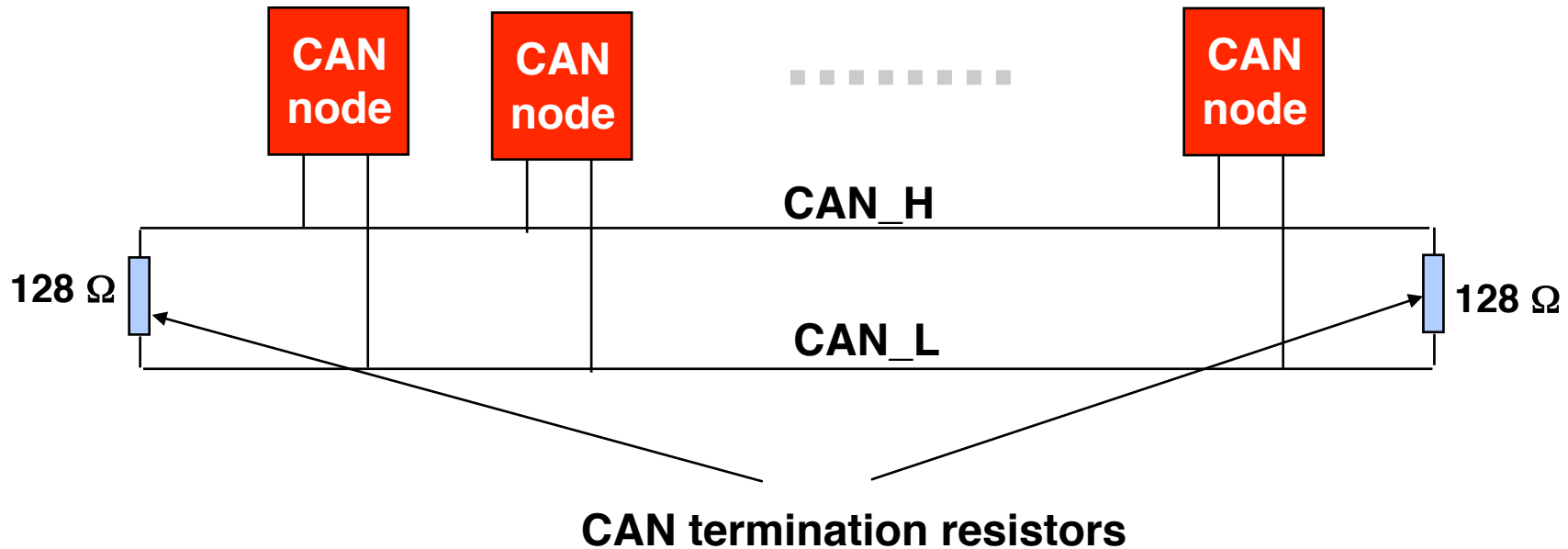- **Multicast with time synchronization.**
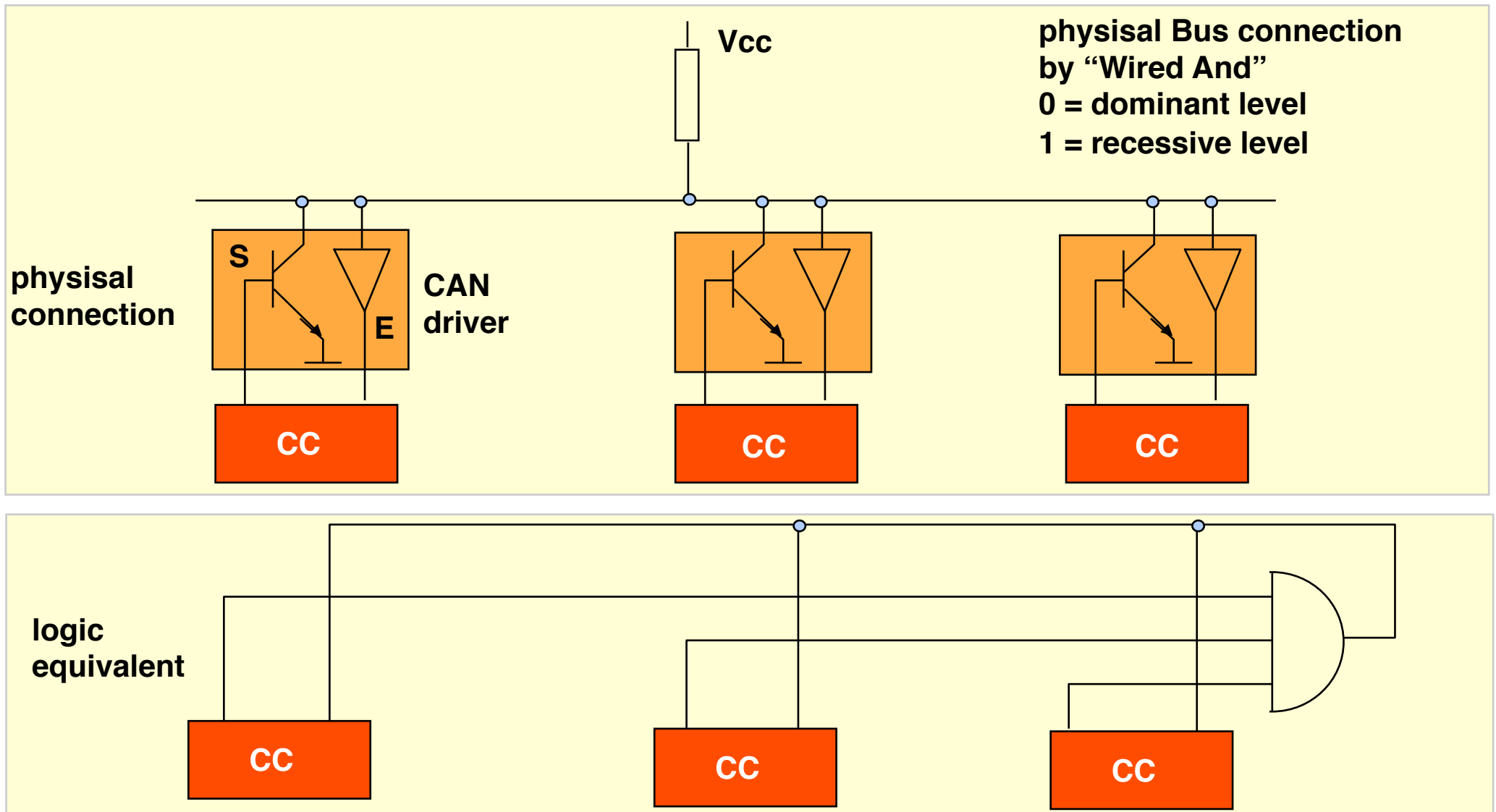
Layers defined by the CAN standard

Data Link Layer

LLC
Acceptance Filtering
Overload Notification
Recovery Management

MAC
Data Encapsulation/ Decapsulation
Frame Coding (Stuffing/Destuffing)
Medium Access Management
Error Detection
Error Signalling
Acknowledge
Serialization /Deserialization

Physical Layer

Bit Encoding / Decoding
Bit Timing
Synchronization

Driver/Receiver Characteristics

Supervisor

Fault Confinement

Bus Failure

LLC = Logical Link Control
MAC = Medium Access Control

CAN differential transmission scheme

CAN node   CAN node   ·········   CAN node

CAN_H

128 Ω   128 Ω

CAN_L

CAN termination resistors

3,5   CAN_H

2,5

1,5   CAN_L

recessive   **dominant**   **recessive**

The CAN physical layer

Vcc

physisal Bus connection
by "Wired And"
0 = dominant level
1 = recessive level

physisal
connection

CAN
driver

S

E

CC
CC
CC

logic
equivalent

CC
CC
CC

CAN Bit Synchronisation

node   node
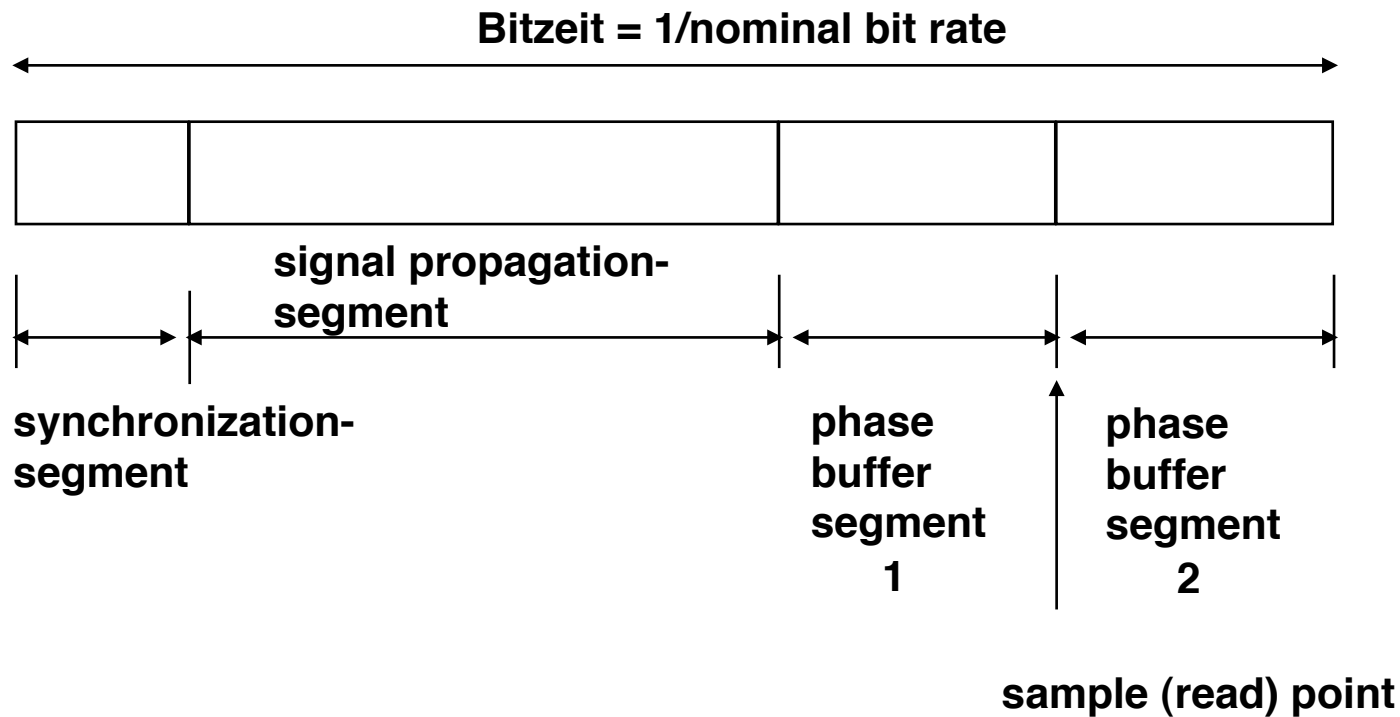
CAN
Bus
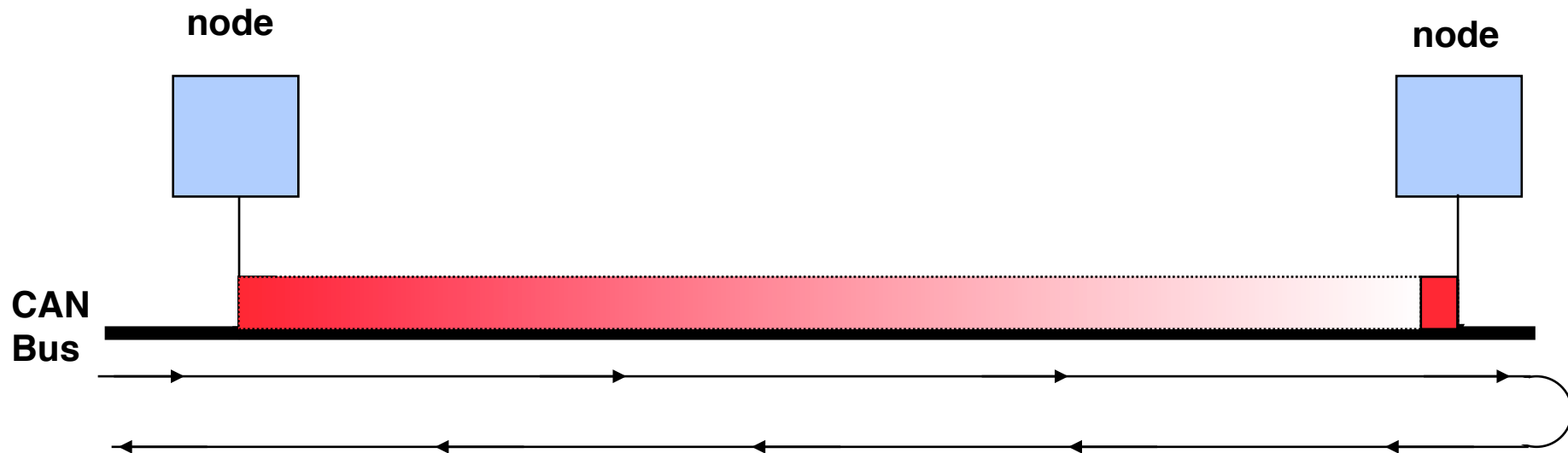
After a certain time, all nodes have seen the value of a bit

Bit rate dependend on the length of the bus

Bit Monitoring

Bit-timing and bit synchronization

Bitzeit = 1/nominal bit rate

**signal propagation-
segment**

**synchronization-
segment**

**phase
buffer
segment
1**

**phase
buffer
segment
2**

sample (read) point

**Länge der Zeitsegmente werden in Vielfachen einer aus der Oszillatorperiode
abgeleiteten Zeiteinheit (time quantum) spezifiziert:**

synch.-segment	**1**	**time quanta**
sig. propag. seg.	**1...8**	**time quantas**
phase buffer seg. 1	**1...8**	**time quantas**
phase buffer seg. 2	**1...8**	**time quantas**

CAN transfer rates in relation to the bus length

$$T_d = T_{TT\text{-}delay} + T_{line\ delay}$$

$T_{TT\text{-}delay}$ ~ **100 ns**

(driver, transceiver, comparator logic, etc.)

$T_{line\ delay}$ ~ **0,2 m / ns twisted pair**

Bitrate (kBits/s)	max. network extension (m)
1000	40
500	112
300	200
200	310
100	640
50	1300

CAN payload

| payload # of bytes	Std. frame kbits/sec	extended frame kbits/sec
0	--	--
1	71,1	61,1
2	144,1	122,1
3	216,2	183,2
4	288,3	244,3
5	360,4	305,3
6	432,4	366,4
7	504,5	427,5
8	576,6	488,5

The CAN MAC and Logical Link Control (LLC) levels

Frame types and formats:

- **Data Frame**　　　normal data transmision initiated by the sender

- **Remote Frame**　　participant requests frame which is sent with the identical frame ID from some other participant.

- **Error Frame**　　　participant signals an error which it has detected

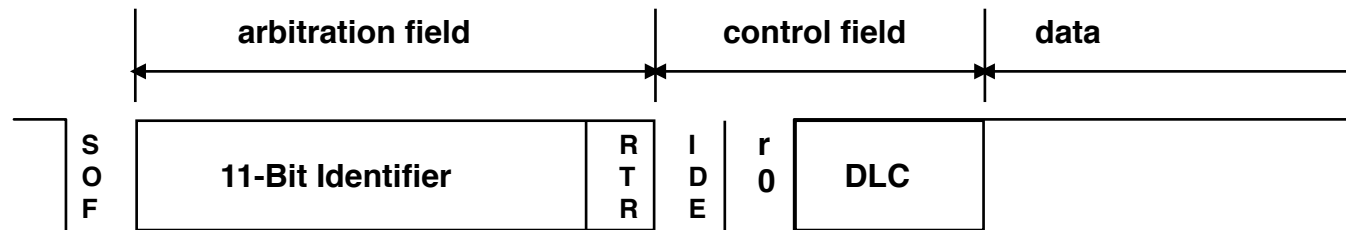- **Overload Frame**　used for flow control. Results in a delayed sending of the subsequent frame.
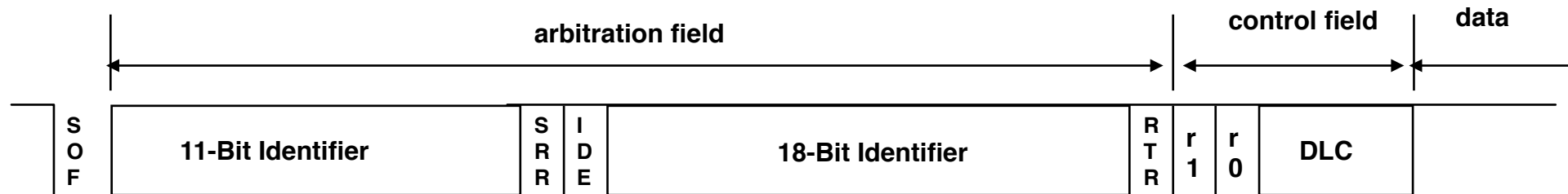
CAN Standard Data Frame
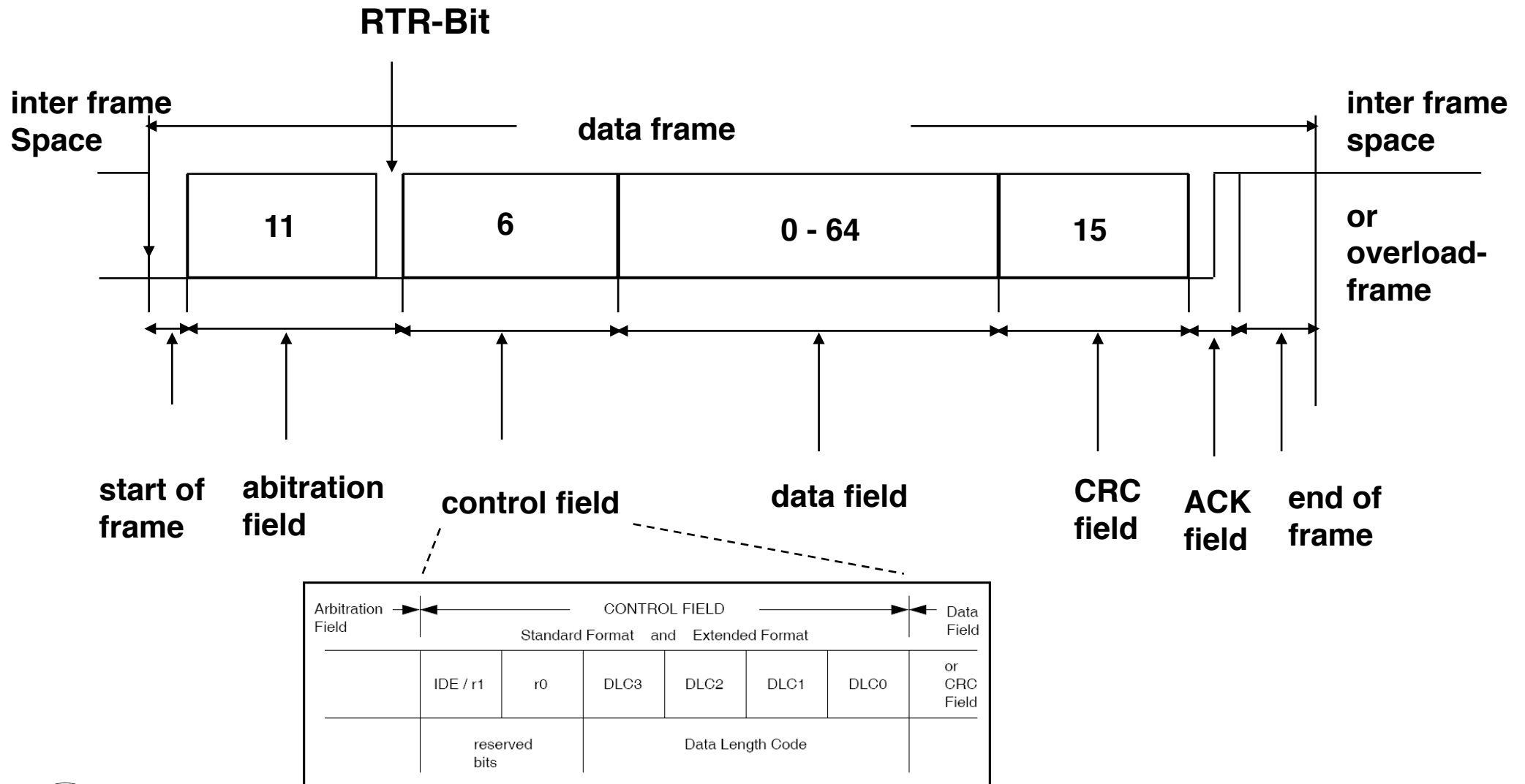
Compatibility between standard and extended frames

Standard Format SF (compatible to CAN Spezifikation 1.2)

Extended Format EF (CAN Spezifikation 2.0)
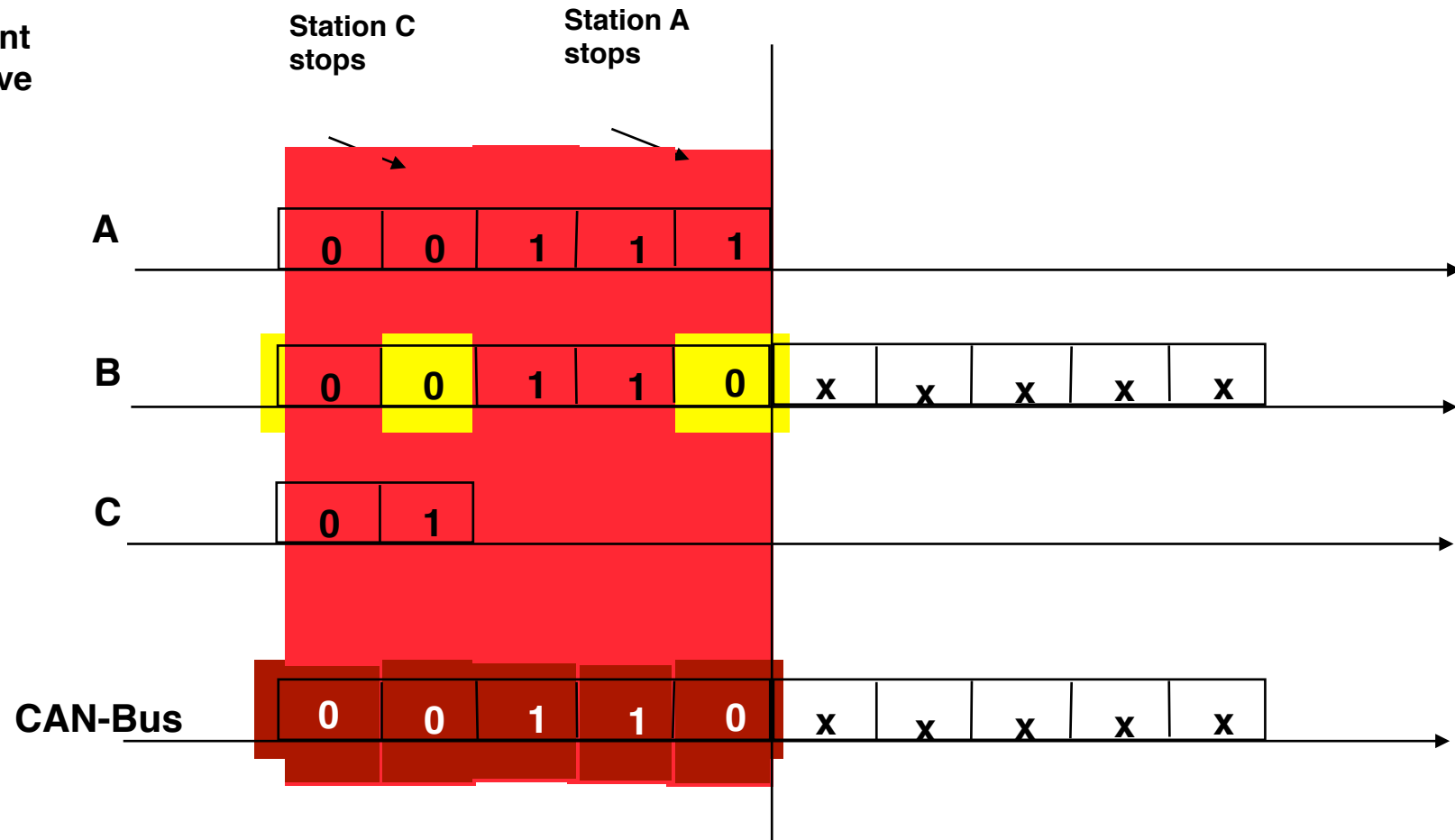

RTR: Remote Transmission Request. In Data Frame: RTR = dominant. In Remote Frame: RTR = recessive.
IDE: Identifier Extension. In the SF this is part of the control field, has a dominant value but is not interpreted.
        In the EF it is part of the addressing field, has a recessive value and causes the format to be recognized as EF.
SRR: Subsitute Remote Request.Always recessive, replaces RTR in the EF for compatibility reasons.
DLC: Data Length Control. 0-8 Byte.
r0, r1:   reserved

Arbitration on a CAN-Bus

0 = dominant
1 = recessive

Station C stops

Station A stops

A

| 0 | 0 | 1 | 1 | 1 |

B

| 0 | 0 | 1 | 1 | 0 | x | x | x | x | x |

C

| 0 | 1 |

CAN-Bus

| 0 | 0 | 1 | 1 | 0 | x | x | x | x | x |

CAN enforces a global priority-based message scheduling

arbitration
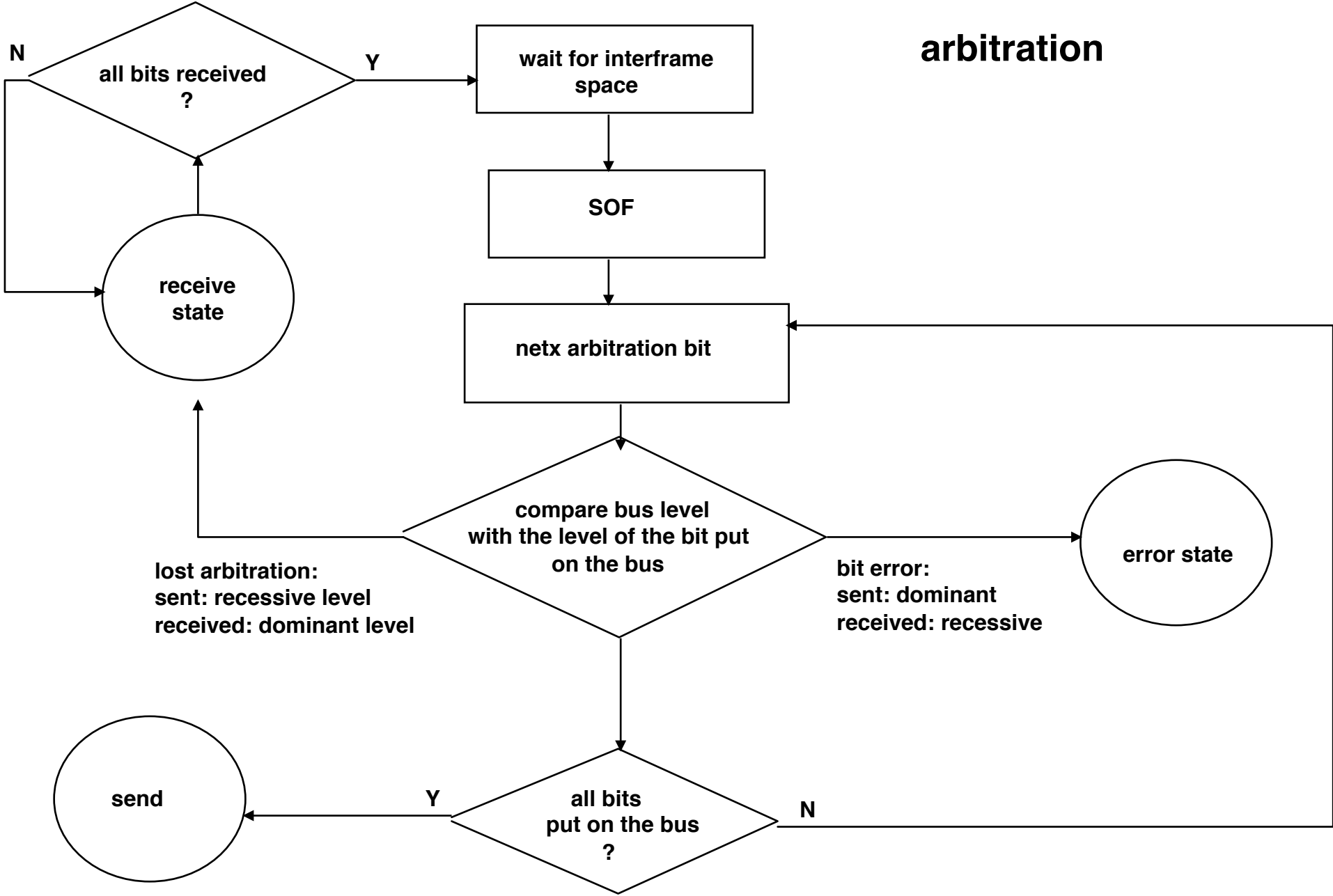
- N / all bits received ? / Y
- wait for interframe space
- SOF
- netx arbitration bit
- receive state
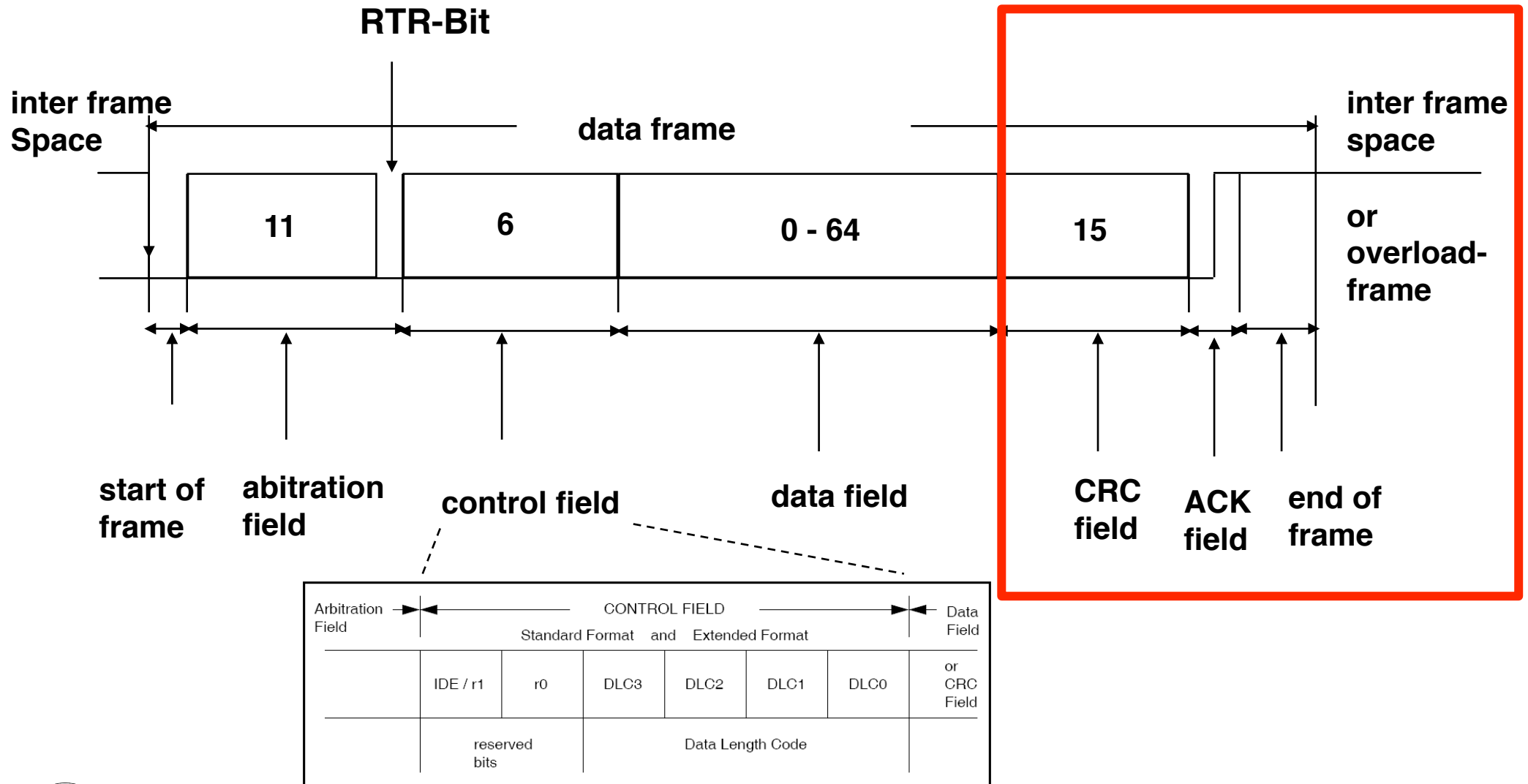- compare bus level with the level of the bit put on the bus
- lost arbitration:
  sent: recessive level
  received: dominant level
- bit error:
  sent: dominant
  received: recessive
- error state
- all bits put on the bus ? / Y / N
- send

CAN Standard Data Frame

RTR-Bit

inter frame
Space

data frame

inter frame
space

| 11 | 6 | 0 - 64 | 15 | |

or
overload-
frame

start of
frame

abitration
field

control field

data field

CRC
field

ACK
field

end of
frame

Arbitration Field		CONTROL FIELD					Data Field
		Standard Format and Extended Format					
	IDE / r1	r0	DLC3	DLC2	DLC1	DLC0	or CRC Field
	reserved bits		Data Length Code				

Anonymous acknowledgement of a CAN message

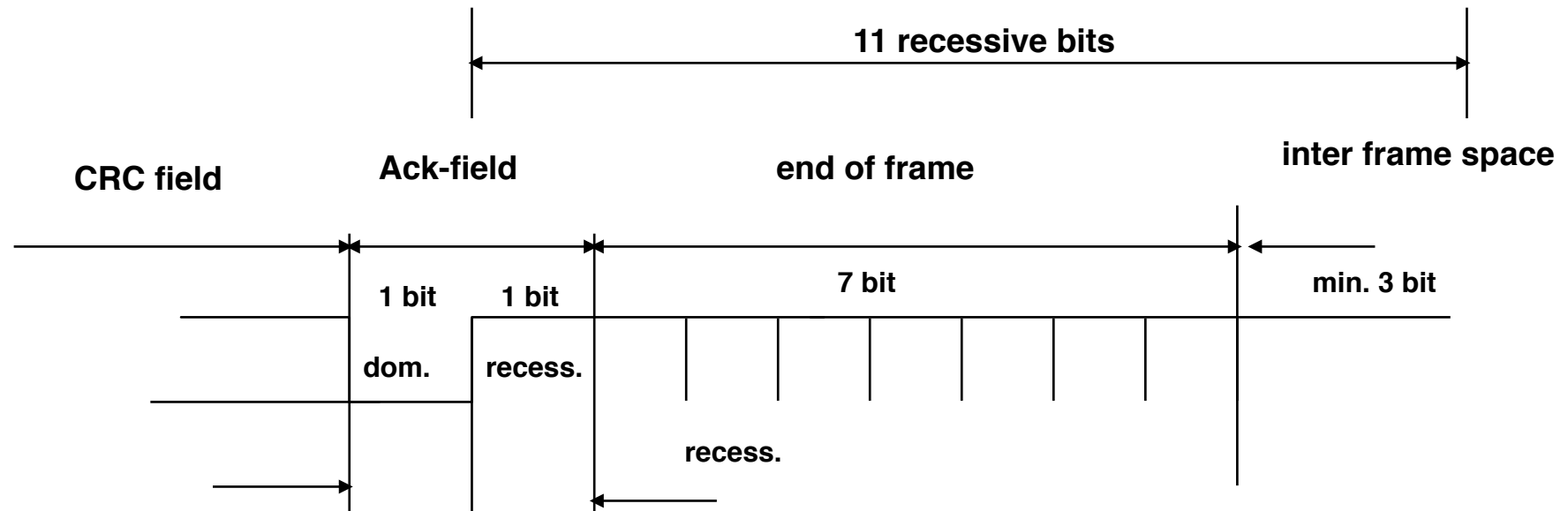| CRC field	acknowledge field	end-of-frame pattern
	1 Bit	1 Bit	7-Bit
	dom.	recess.	recess.

ACK-Slot

ACK delimiter
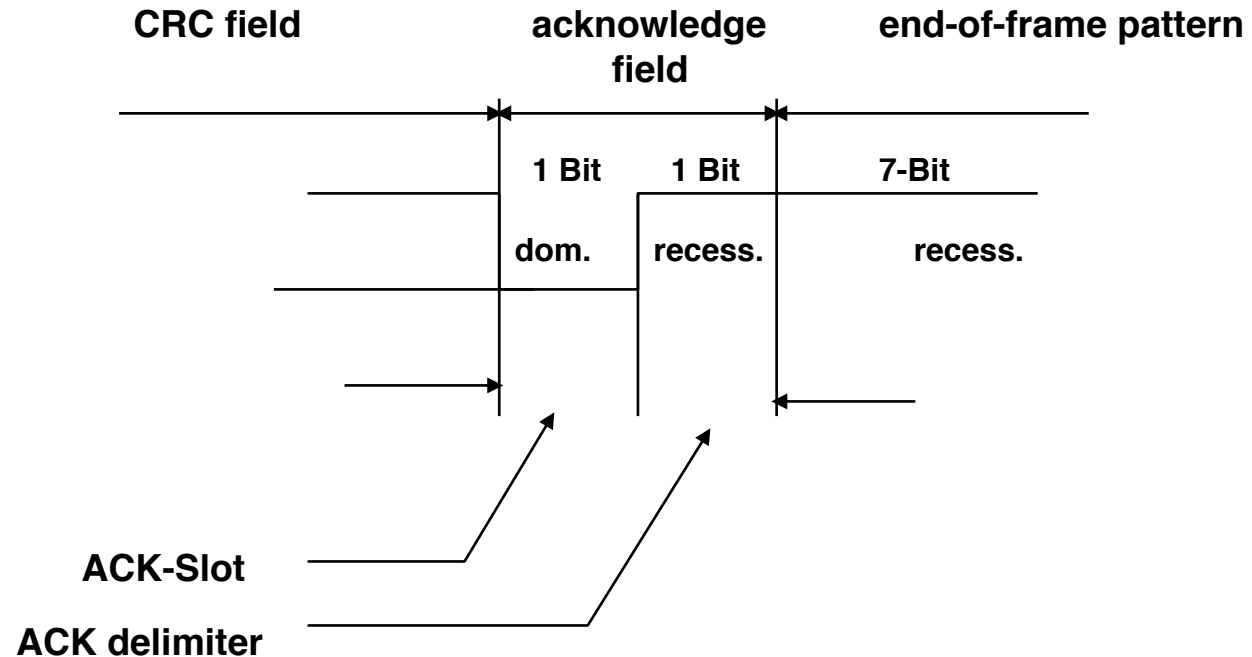
positive anonymous acknowledgement (Broadcast !)
receivers that correctly received a message(a matching CRC sequence) report this in the ack-slot
by superscibing the recessive bit of the sender by a dominat bit. The sender switches to a recessive
level.

➡ **Message is acknowledged by a single correct reception on a correct node.**
➡ **Systemwide data consistency requires additional signalling of local faults.**

Termination sequence of a frame

11 recessive bits

CRC field   Ack-field   end of frame   inter frame space

1 bit   1 bit   7 bit   min. 3 bit

dom.   recess.   recess.

Goals:
1. Detecting AND signalling the error within the actual fame in which it occured
2. Identifying the node which may have caused the error.
3. Creating a systemwide view on the reception state of the message.

Approach: End of frame pattern consisting of 7 recessive bits.
      1. Any error detection is signalled by putting a dominant bit on the bus.
      2. An out-of-sync node, not being aware of the EOF sequence will signal an error at position "6".

Interframe Space

Frame   Inter-Frame Space   Frame

sof

I1   I2

Intermission   Bus Idle
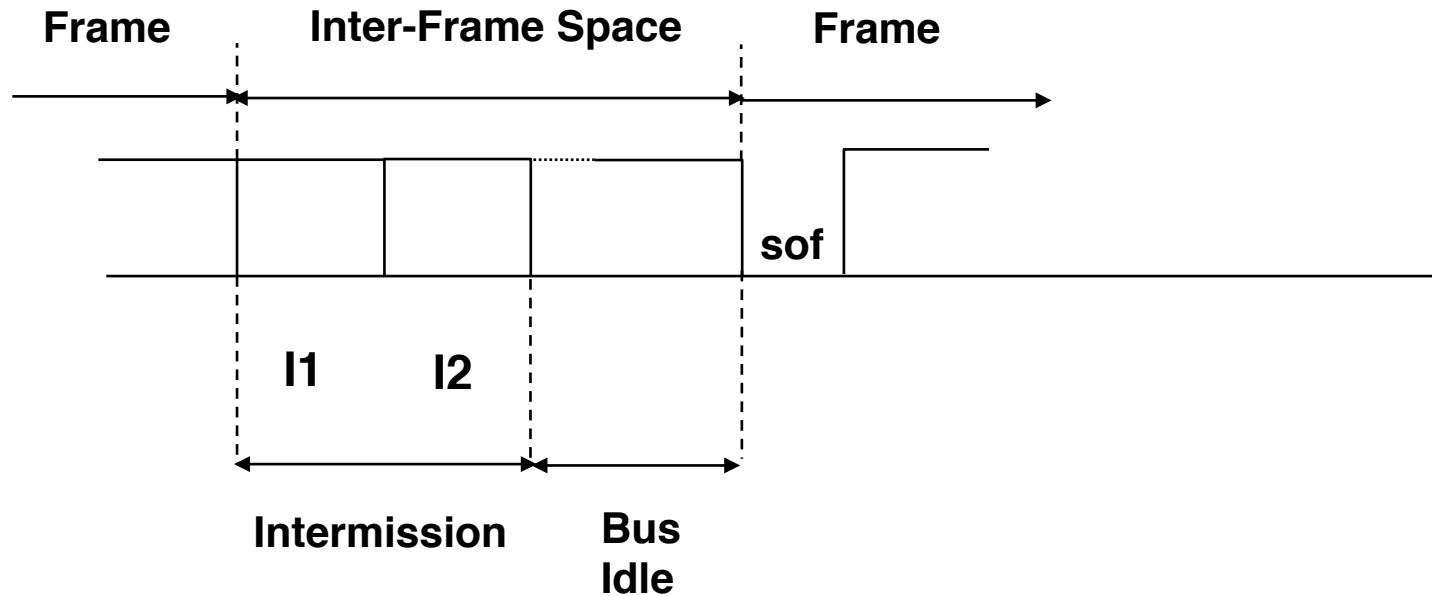
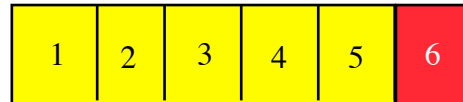**Intermission: no data- or remote Frame may be started**

Intermission 1: active overload Frame may be started
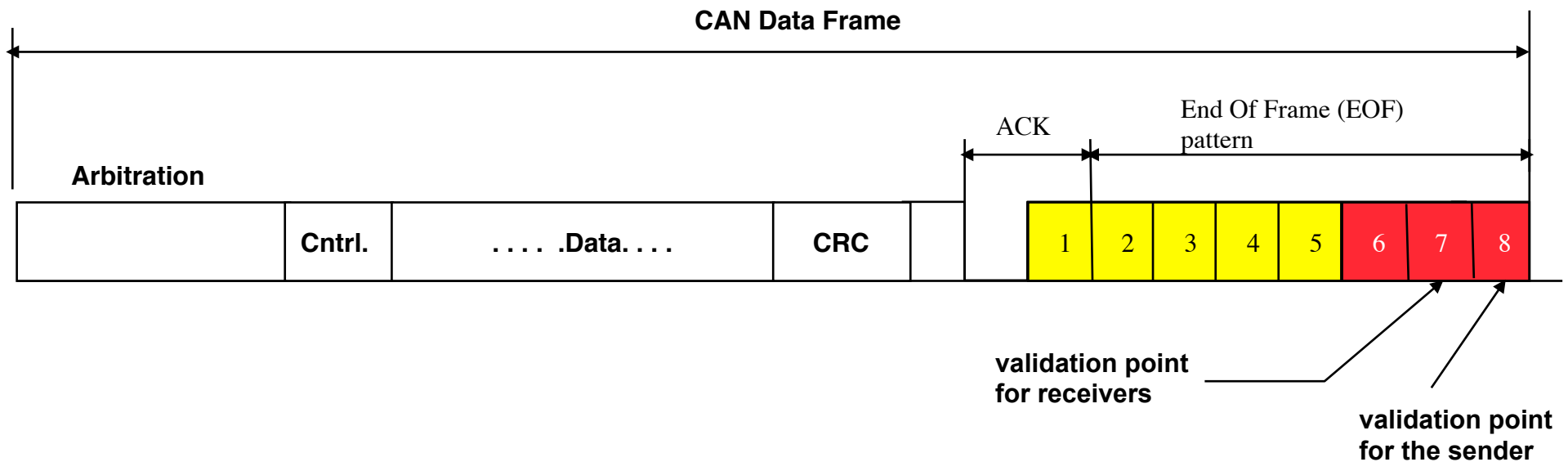**Intermission 2: re-active overload frame (after detecting a dominant bit in I1)**

Error Detection and Error Signalling in CAN

Violation of the Bit-Stuffing Rule:
Used for Error Detection and Signalling

1	2	3	4	5	6

Bit-Stuffing enforces the following rule:

A sequence of 5 identical bit levels
is followed by a complementary bit level

CAN Data Frame

ACK

End Of Frame (EOF) pattern

Arbitration

		Cntrl.	. . . . .Data. . . .	CRC			1	2	3	4	5	6	7	8

validation point
for receivers

validation point
for the sender

Error detection

1.) Monitoring: Sender compares the bit sent with the bit actually on the bus.

      Type of faults: local sender faults

      Error detection: sender based

2.) Cyclic Redundancy Check:

      Type of faults: 5 arbitrarily distributed faults in the code word,
                            burst error max. length 15.

      Error detection: receiver based

3.) Bitstuffing:

      Type of faults: transient faults, stuck-at-faults in the sender

      Error detection: receiver based

4.) Format control:

      Type of faults: the specified sequence of fields is violated.

      Error detection: receiver based

5.) Acknowledgment:

      Type of faults: no acknowledge

      Error detection: sender based, sender assumes local fault.

Risk of undetected errors

Bit monitoring: **An error will not be detected if**
- the sender is correct and monitoring doesn't detect an error
- all other nodes receive the same bit pattern which is different from that of the sender and contains a non-detectable error.

Bit-stuffing: **double errors within 6 bits will not be detected**

CRC: **difference between frame sent and received is a multiple of the generator polynome.**

Frame errors: **the frame is shortened or additional bits are added. At the same time a correct end-of-frame sequence is generated.**

Unruh, Mathony und Kaiser:"Error Detection Analysis of Automotive Communication Protocols", SAE International Congress, Nr. 900699, Detroit, USA, 1990
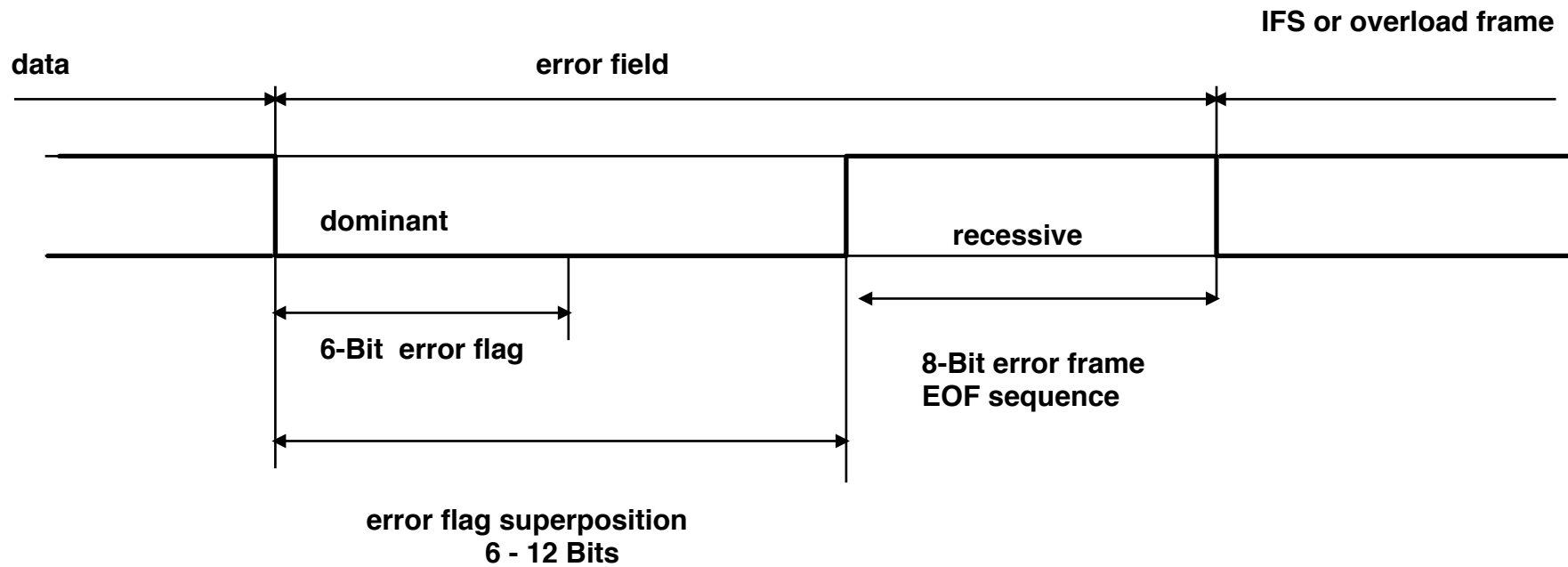
Scenario:
nodes: 10, Bit error rate: $2 \cdot 10^{-2}$, message error rate: 10^{-3}
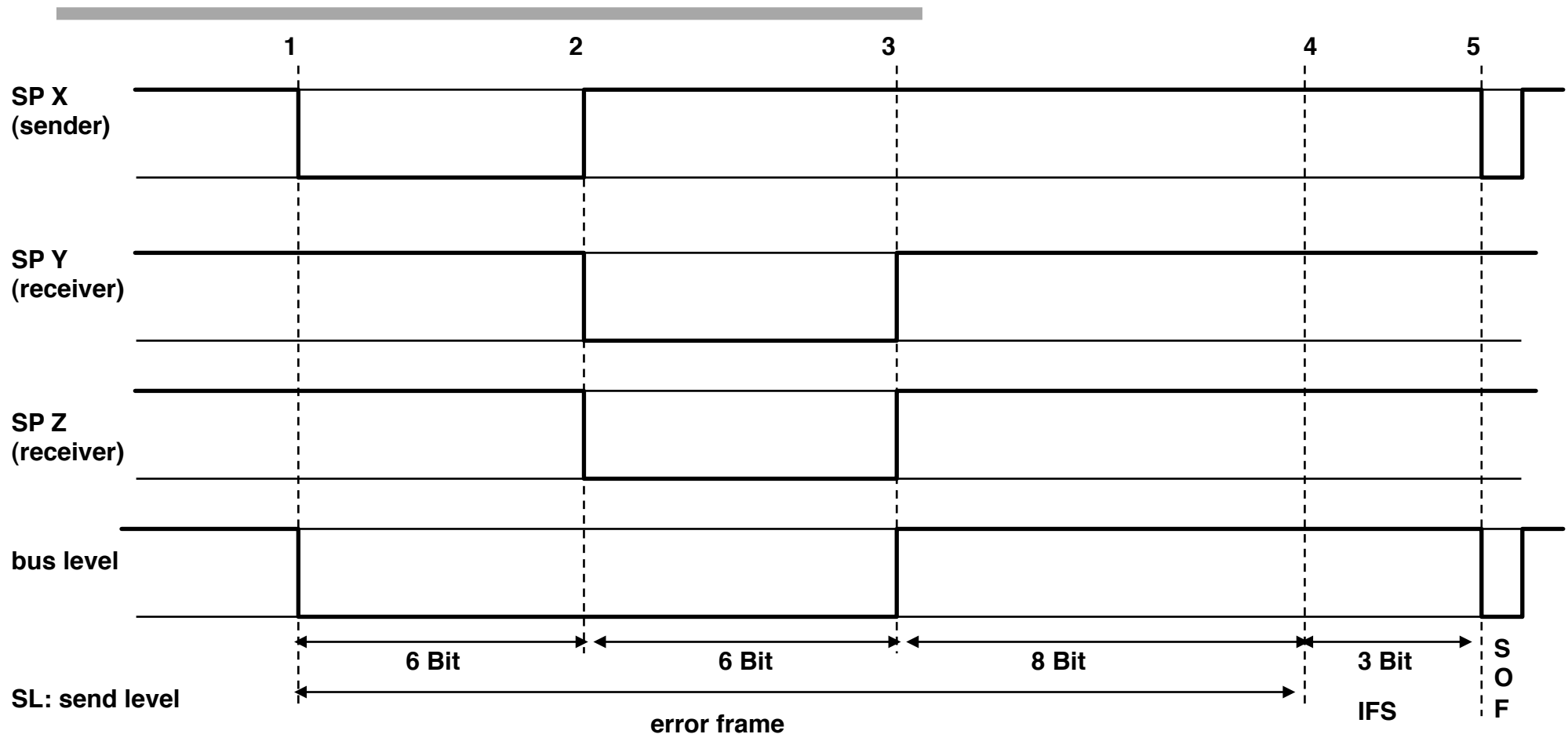**risk of undetected errors: $4,7 \cdot 10^{-14}$**
When the number of nodes increase, the probability of undetected errors decreases.

CAN error frame

IFS or overload frame

data

error field

dominant

recessive

6-Bit error flag

8-Bit error frame
EOF sequence

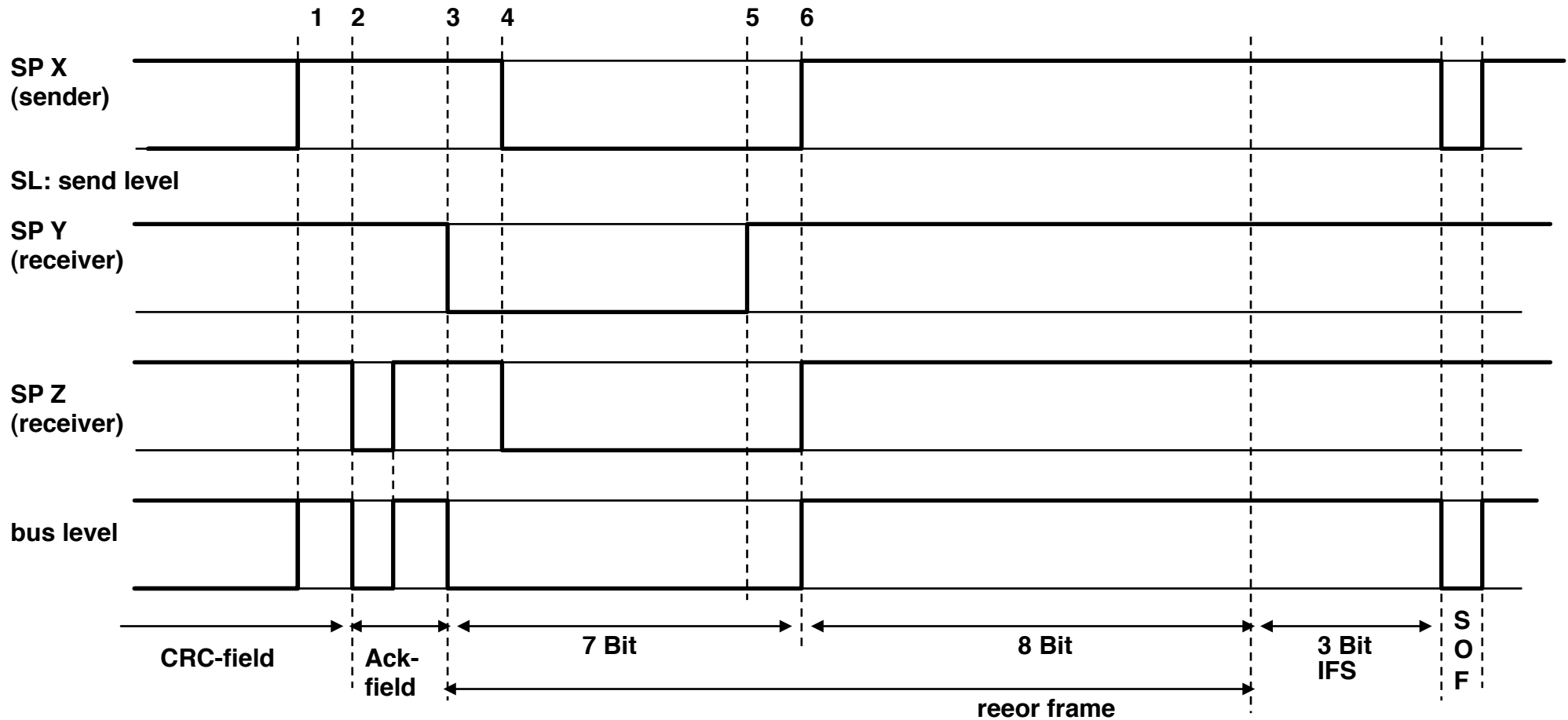error flag superposition
6 - 12 Bits

Error frame resulting from a sender fault

time to re-transmit a faulty message frame: min. error recovery time: 23 bit times

Error frame resulting from a receiver fault

	1	2	3	4	5	6	

SP X (sender)

SL: send level

SP Y (receiver)

SP Z (receiver)

bus level

CRC-field   Ack-field   7 Bit   8 Bit   3 Bit IFS   S O F

reeor frame

time to re-transmit: min. error recovery time: 20 bit times

Enforcing fault confinement and a "Fail Silent" behaviour

Problem: Faulty component may block the entire message transfer on the CAN-Bus.

Assumption: 1. A faulty node detects the error first.
                  2. frequently being the first which detects an error --> local fault in the node

approach: error counter for receive and transmit errors. If error was first detected by the node, the counter is increased by 8-9.
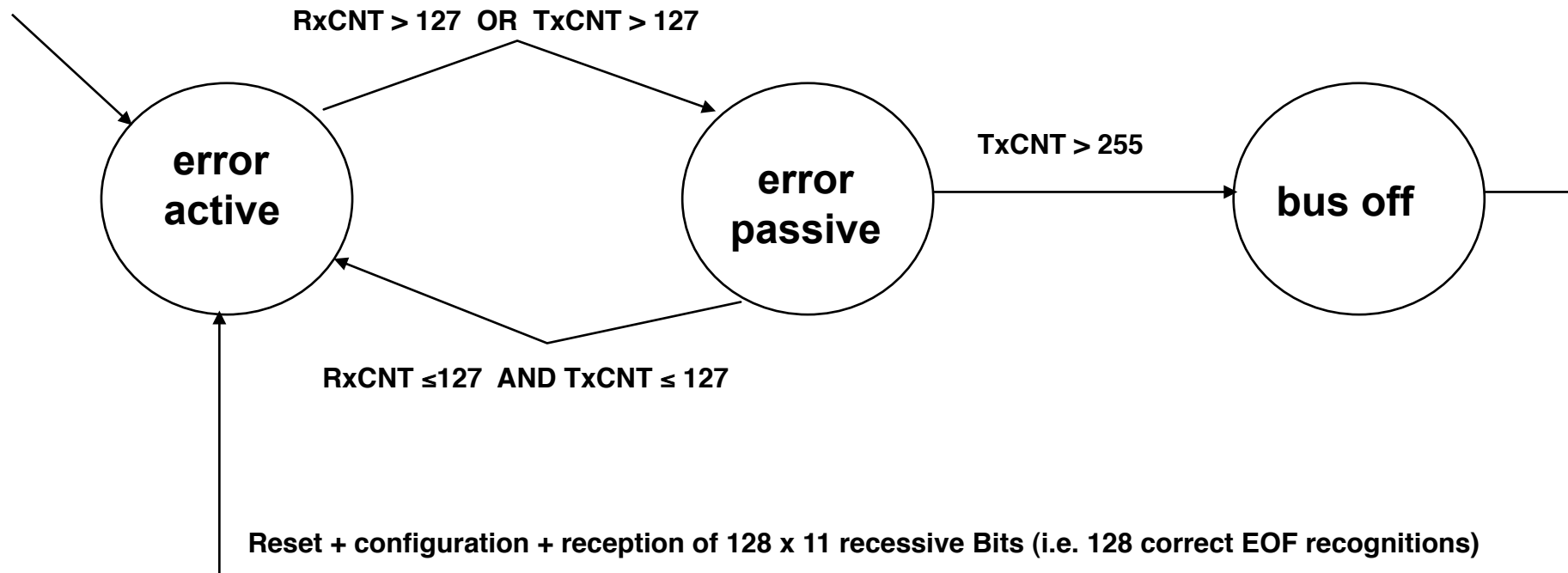
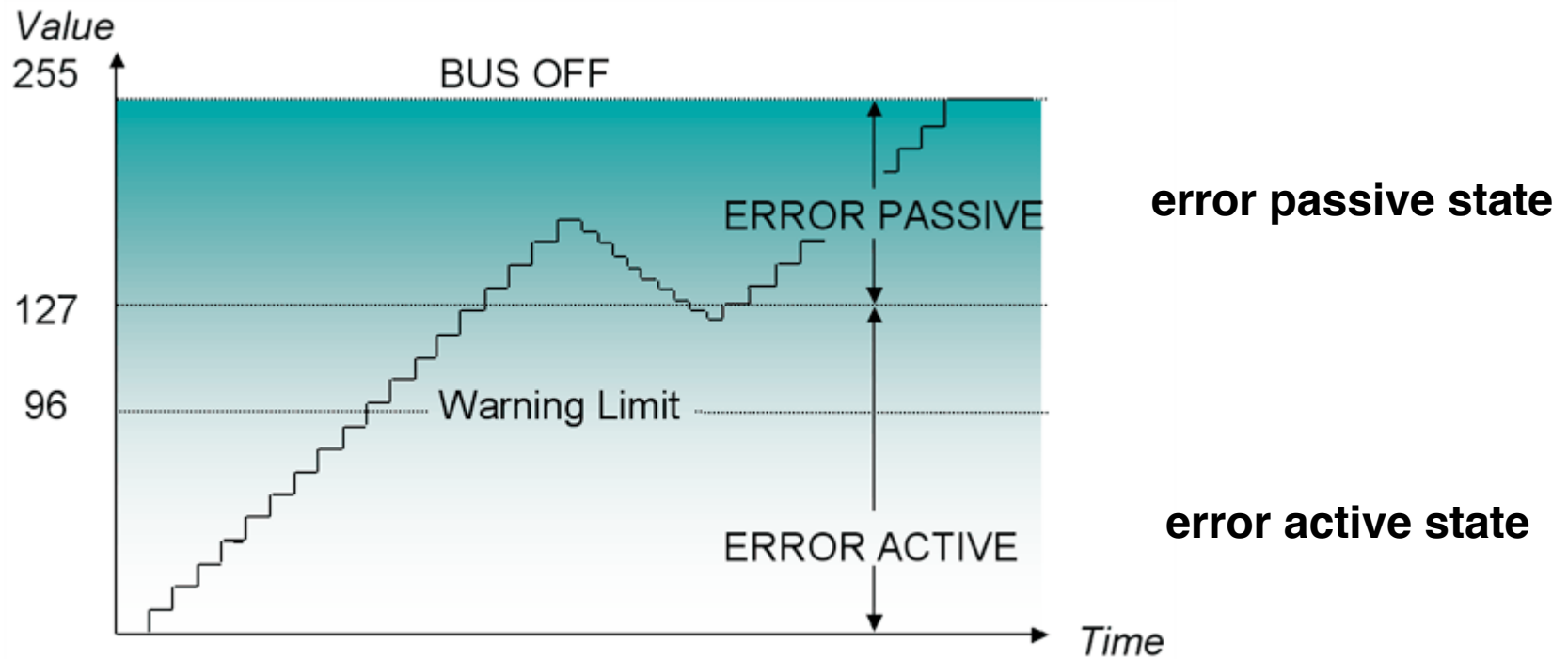# Enforcing fault confinement and a "Fail Silent" behaviour

States of a CAN node:
- error active
- error passive
- bus off

RxCNT: Value of the receive counter
TxCNT: Value of the transmit counter

RxCNT > 127 OR TxCNT > 127

(error active) → (error passive)

TxCNT > 255

(error passive) → (bus off)

RxCNT ≤127 AND TxCNT ≤ 127

Reset + configuration + reception of 128 x 11 recessive Bits (i.e. 128 correct EOF recognitions)
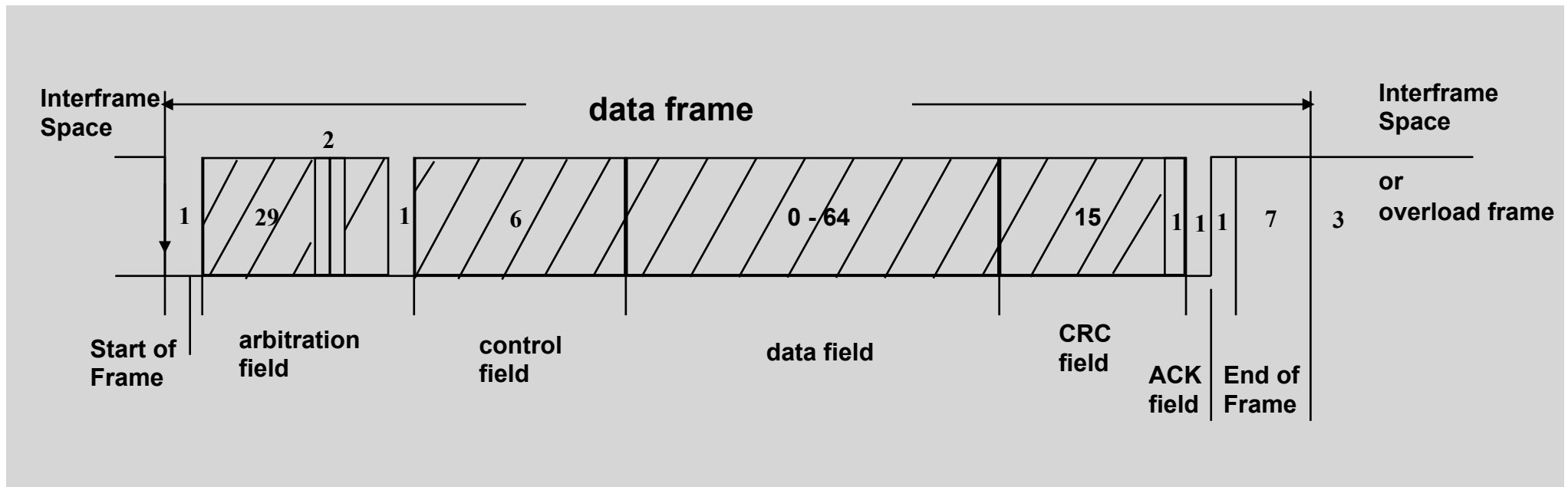
CAN bus Error Handling - Transmit Error Counter

Analysis of CAN inaccessibility

CAN Data Frame

longest possible message:
Format-Overhead: 67 bit times
Data: 64 bit times
Bitstuffing (max): 23 bit times

total: 154 bit times

CAN Inaccessibility Times*

Data Rate 1 Mbps , Standard Format

Scenario	t_{inacc} (µs)	
Bit Errors	155.0	← worst case
Bit Stuffing Errors	145.0	single
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	
Transmitter Failure	2480.0	← worst case
Receiver Failure	2325.0	multiple

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

Predictability of various Networks*

Worst Case Times of Inaccessibility*	t_{inacc} (ms)	
ISO 8002/4 Token Bus (5 Mbps)	139.99	Token-based Protocols
ISO 8002/5 Token Ring (4 Mbps)	28278.30	
ISO 9314 FDDI (100 Mbps)	9457.33	
Profibus (500 kbps)	74.80	
CSMA/CD	unbounded	CSMA Protocols
CSMA/CA	stochastic	
CAN-Bus (1Mbps)	2.48	

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

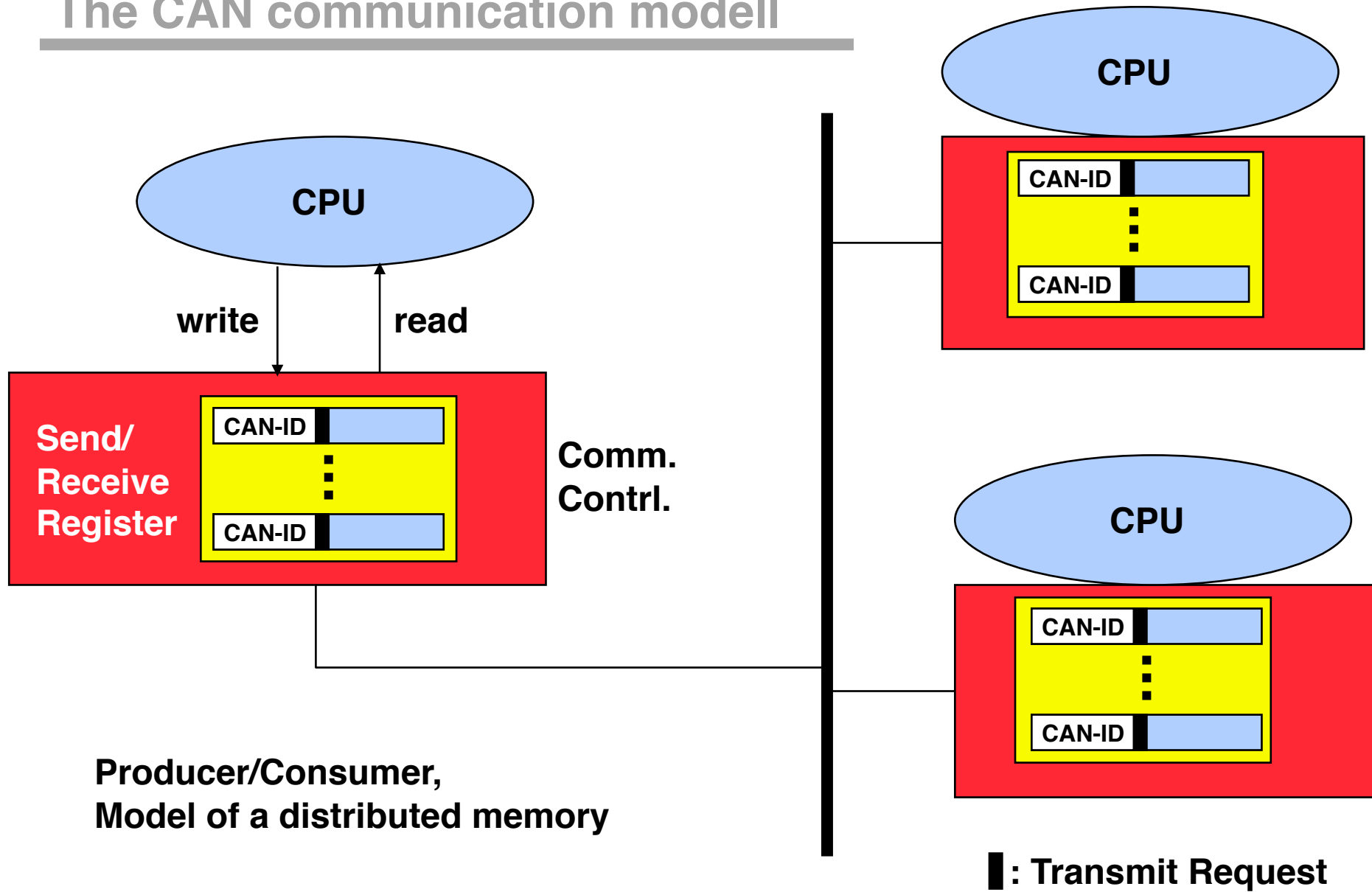# High level issues

Routing: How does a message reach a receiver ?

 CAN: Broadcast, message subjects

Filtering: How can the receiver only receive those messages selectively in which it is interested in ?
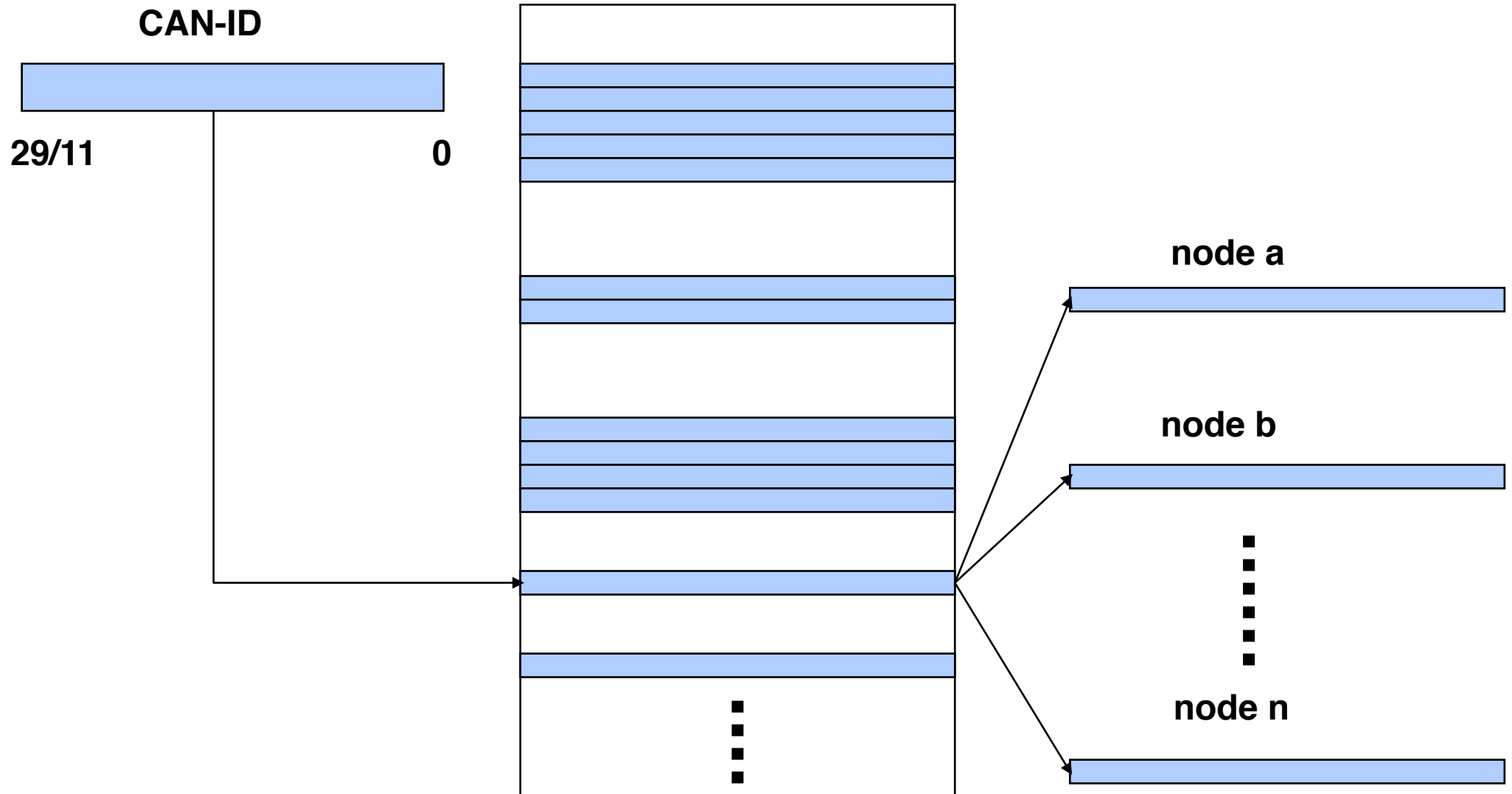
CAN: message filters

The CAN communication modell

CPU

write   **read**

Send/ Receive Register

CAN-ID

CAN-ID

Comm. Contrl.

CPU

CAN-ID

CAN-ID

CPU

CAN-ID

CAN-ID

Producer/Consumer, Model of a distributed memory

▐ **: Transmit Request**

The CAN communication modell

CAN-ID

29/11　　　　　　　　　　　　**0**

node a

node b

node n

CAN-Bus Properties (summary)

➡️ **Event-triggered communication with low latency**

➡️ **Priority-based arbitration with collision resolution for guaranteed throughput**

➡️ **error handling:**
   anonymous positive acknowledge
   negative ack. in case of an error (system wide messaging)
   identification of faulty nodes
   immediate synchronization and retransmission

➡️ **content-based addressing with a high flexibilitx (system elasticity)**