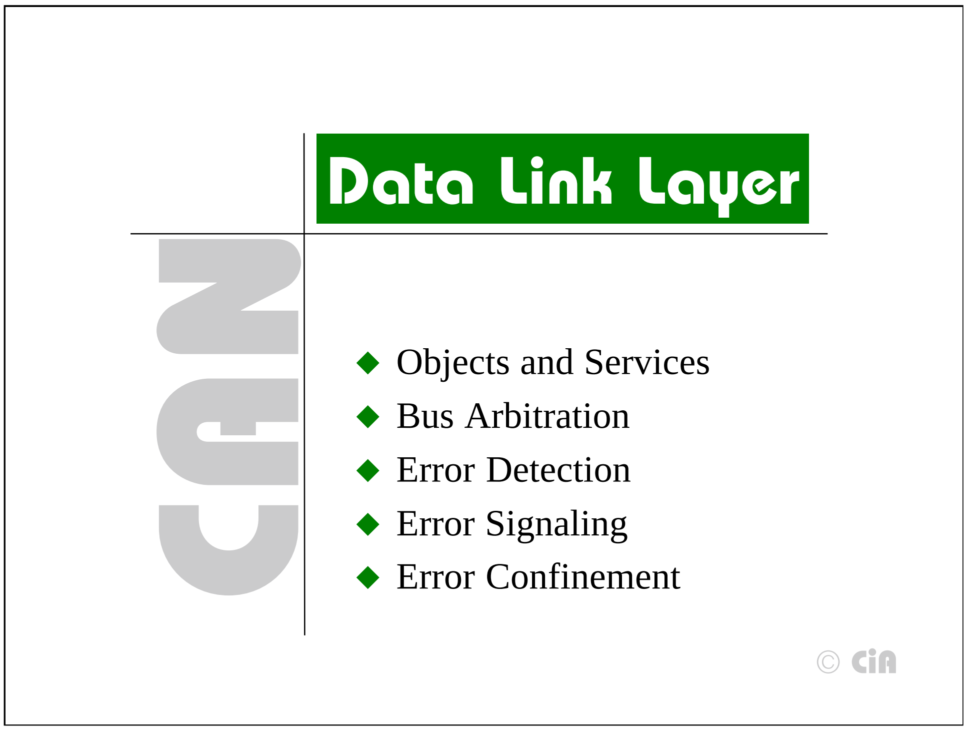
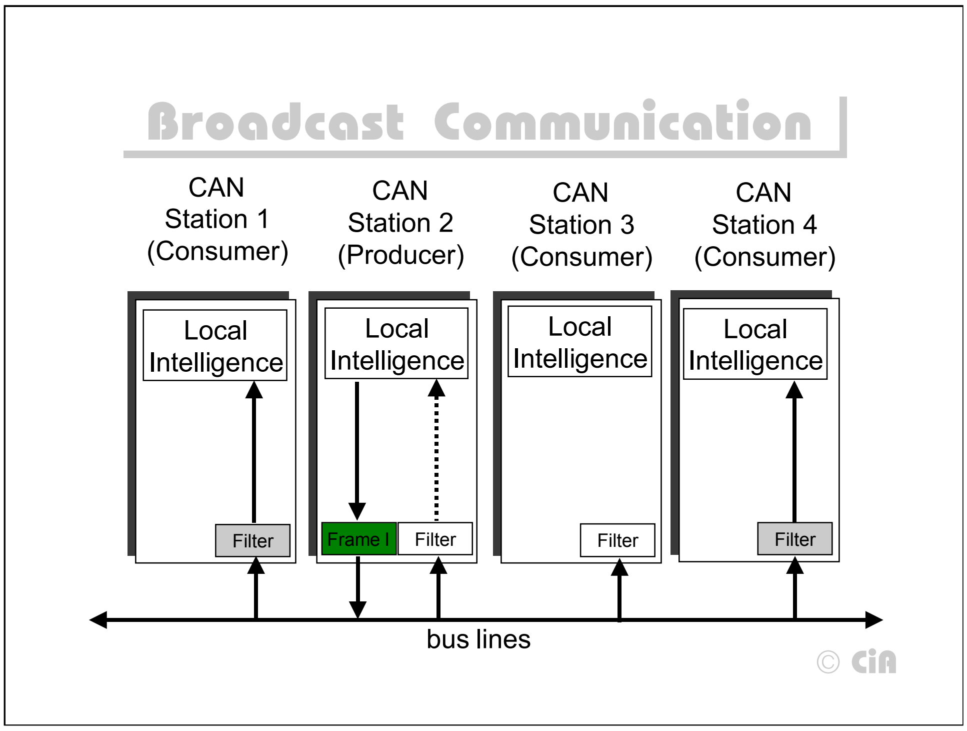


Data Link Layer

- Objects and Services
- Bus Arbitration
- Error Detection
- Error Signaling
- Error Confinement

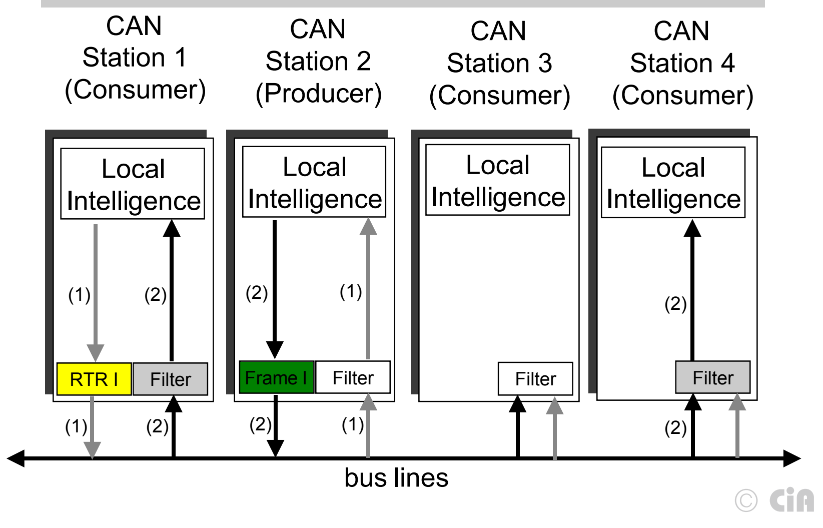
The CAN data link layer is standardized in ISO 11898. The data link layer services are implemented in the Logical Link Control (LLC) and Medium Access Control (MAC) sub-layers of a CAN controller. The LLC provides acceptance filtering, overload notification and recovery management. The MAC is responsible for data encapsulation (de-capsulation), frame coding (stuffing/de-stuffing), medium access management, error detection, error signaling, acknowledgement, and serialization (de-serialization).



The CAN concept of Broadcast Communication means that every station of the network can listen to the frames of the transmitting station (here: station 2). After receiving the frame it is the task of every node to decide if the message has to be accepted or not. So Acceptance Filtering has to be implemented in every CAN node.

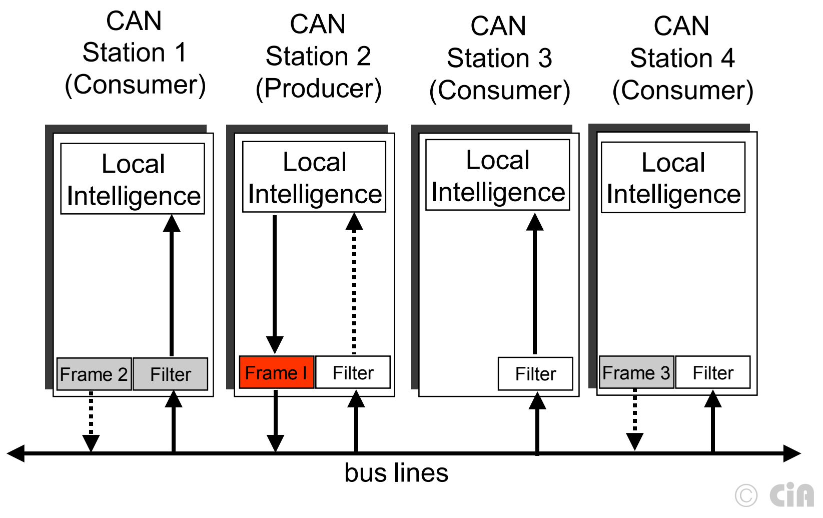
The CAN Broadcast Communication can be compared with a radio station transmitting information about traffic accumulation for vehicle drivers. Every driver has to decide if the messages are important for him dependent on the motorway he wants to use.

Remote Request



Remote Transmission Requests (RTR) are like questions. The node that has the answer will produce in a second communication the requested data. This data frame can be received also by other consumers, which are interested in this object Remote frames and data frames are identified by a specific field, called Identifier.

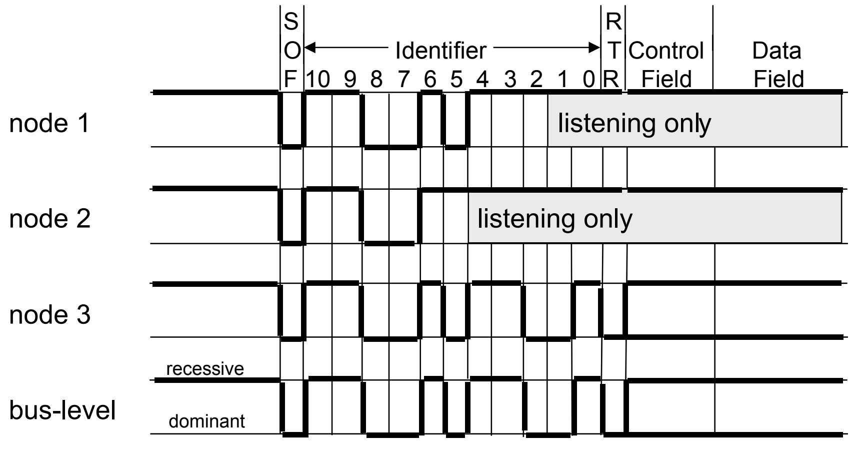
Multiple Bus Access



The CAN protocol allows simultaneous bus access from different nodes. If more than one node is accessing the bus, a arbitration is required. The bus access method used in CAN is a non-destructive, bit-wise arbitration, called Carrier Sense Multiple Access with Collision Detection and Arbitration on Message Priority (CSMA/CD + AMP). The message priority is decoded in the CAN identifier.

When the bus is in Idle state, several nodes can start transmission of a frame. Every node reads back, bit by bit, from the bus during the complete message and compares the transmitted bit value with the received bit value. Per definition the bits with a dominant value overwrites those with a recessive value (this has to be provided by the transceiver).

Bus Arbitration Method

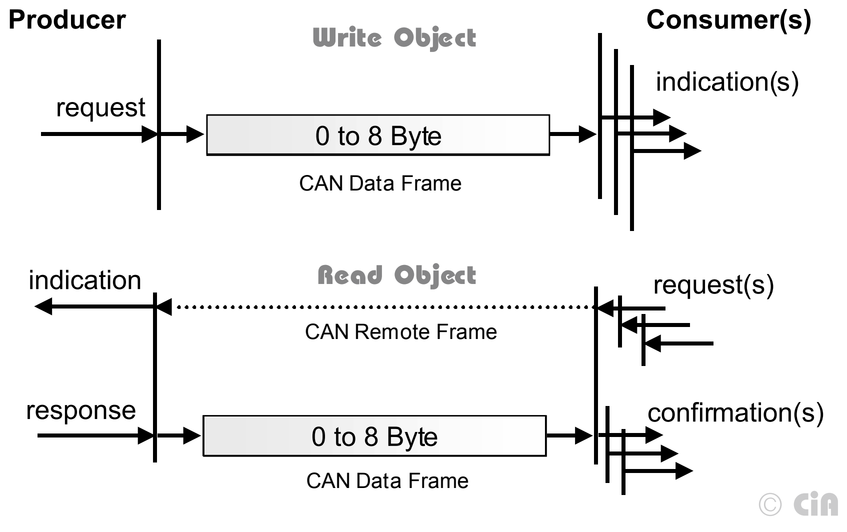


Node 3 wins arbitration and transmits his data.

If two or more bus nodes start their transmission at the very same time after having found the bus to be idle, collision of the messages is avoided by the implemented CMSA/CA + AMP bus access method. Each node sends the bits of its message identifier and monitors the bus level. As long as the bits from all transmitters are identical nothing happens.

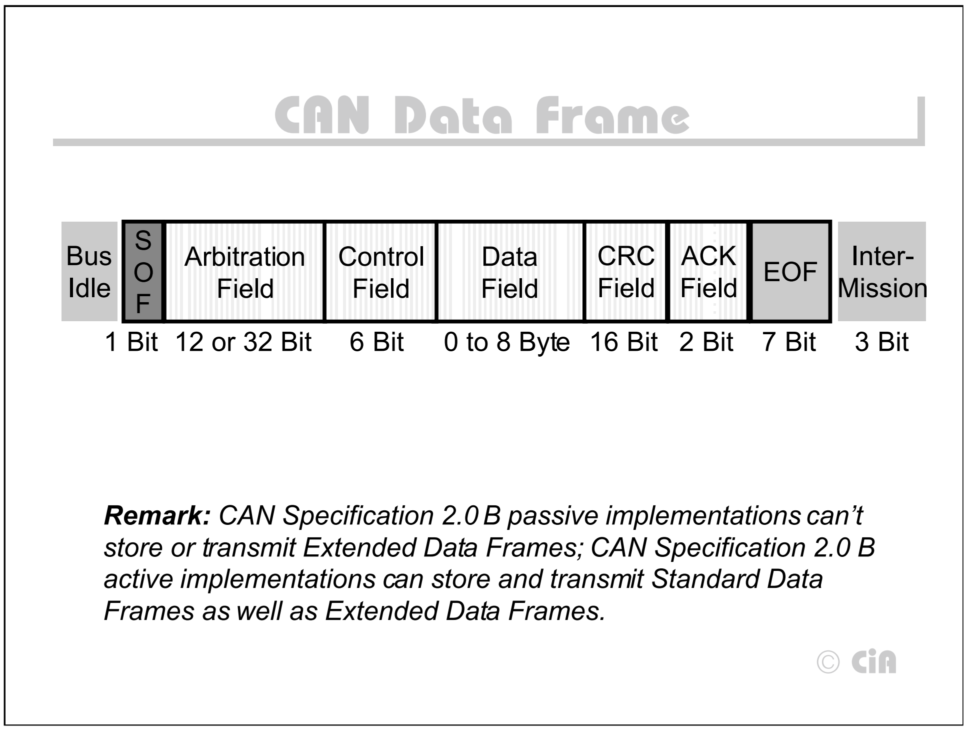
At bit 5 nodes 1 and 3 send a dominant identifier bit. Node 2 sends a recessive identifier bit but reads back a dominant one. Node 2 looses bus arbitration and switches to listening only mode that is transmitting recessive bits. At bit 2 node 1 looses arbitration against node 3. This means that the message identifier of node 3 has a lower binary value and therefore a higher priority than the messages of nodes 1 and 2. In this way the bus node with the highest priority message wins arbitration without loosing time by having to repeat the message. Nodes 1 and 2 will send their messages after node 2 has finished his transmission.

Communication Services



The CAN protocol provides two communication services. This Write Object service transmits a Data Frame from one node (the producer) to one or more receiving nodes (consumers). This don’t implies that one node will accept the message meaning that some one is interested in this information. This service is the classic CAN communication service.

The second communication service is to request a specific message. This Read Object service is initiated by one or more consumers. Therefore these nodes will transmit a so-called Remote Frame. The node, which owns the requested information will transmit the corresponding Data Frame.



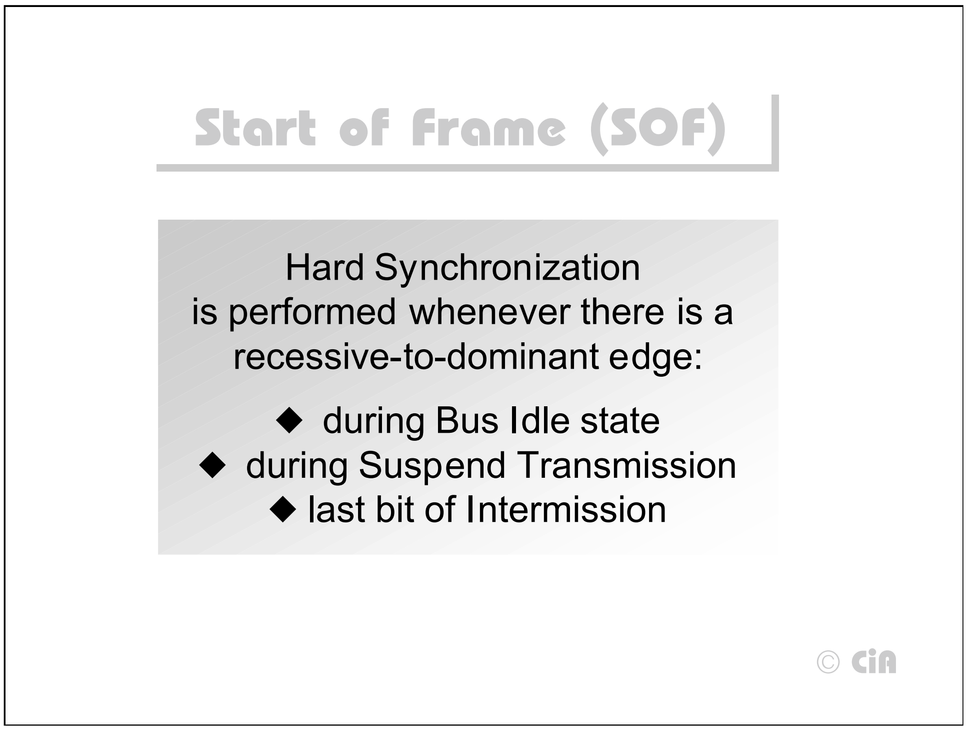
CAN Data Frame

Bus Idle	SOF	Arbitration Field	Control Field	Data Field	CRC Field	ACK Field	EOF	Inter-Mission
	1 Bit	12 or 32 Bit	6 Bit	0 to 8 Byte	16 Bit	2 Bit	7 Bit	3 Bit

Remark: *CAN Specification 2.0 B passive implementations can't store or transmit Extended Data Frames; CAN Specification 2.0 B active implementations can store and transmit Standard Data Frames as well as Extended Data Frames.*

A Data Frame is produced by a CAN node when the node wishes to transmit data or if this is requested by another node. Within one frame up to 8 byte data can be transported.

The Data Frame begins with a dominant Start of Frame (SOF) bit for hard synchronization of all nodes. The SOF bit is followed by the Arbitration Field reflecting content and priority of the message. The next field is the Control Field which specifies mainly the number of bytes of data contained in the message. The Cyclic Redundancy Check (CRC) Field is used to detect possible transmission errors. It consists of a 15-bit CRC sequence completed by the recessive CRC delimiter bit. During the Acknowledgement (ACK) Field the transmitting node sends out a recessive bit. Any node that has received an error free frame acknowledges the correct reception of the frame by sending back a dominant bit. The recessive bits of the End of Frame end the Data Frame. Between two frames there must be an recessive 3-bit Intermission field.



Start of Frame (SOF)

Hard Synchronization
is performed whenever there is a
recessive-to-dominant edge:

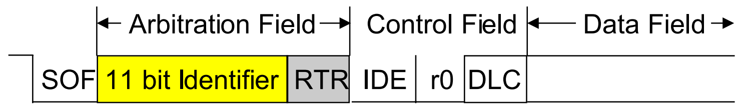
◆ during Bus Idle state
◆ during Suspend Transmission
◆ last bit of Intermission

© CiA

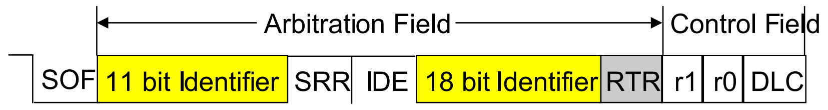
Synchronization rules of the CAN specification requires Hard Synchronization to be performed at every edge from recessive-to-dominant edge during Bus Idle. Additionally, Hard Synchronization is required for each received SOF bit. An SOF bit can be received both during Bus Idle, and also during Suspend Transmission and at the end of Interframe Space. The Bosch CAN Reference Model therefore enables Hard Synchronization not only for Bus Idle state, but also for Suspend state and the last bit of Interframe Space. Any node disables Hard Synchronization if it samples an edge from recessive to dominant or if it starts to send the dominant SOF bit. Since synchronization on edges from dominant to recessive has become obsolete with the upgrade from CAN specification 1.1 to version 1.2, the Bosch CAN Reference Model does not support this kind of synchronization!

Arbitration Field

Standard Frame Format



Extended Frame Format

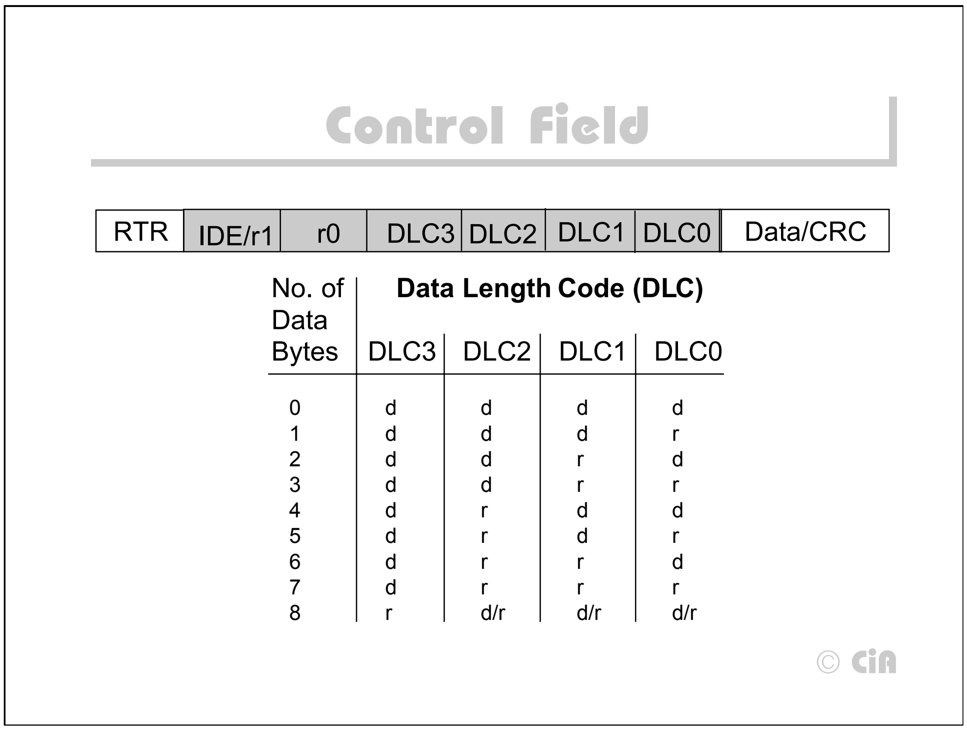


The Identifier´s length in the Standard Format is 11 bit and corresponds to the Base ID in Extended Format. The Identifier is followed by the RTR bit. In Data Frames the RTR bit has to be dominant. Within a Remote Frame the RTR bit has to be recessive. The Base ID is followed by the IDE (Identifier Extension) bit transmitted dominant in the Standard Format (within the Control Field) and recessive in the Extended Format. So the Standard Frame prevails the Extended Frame in case of collision.

The Extended Format comprises two sections: Base ID with 11 bit and the Extended ID with 18 bit. The SRR (Substitute Remote Request) bit substitutes the RTR bit and is transmitted recessive. If the SRR is corrupted and transmitted dominant, the receivers will ignore this. But the value is not ignored for bit stuffing and arbitration. Since the SRR bit is received before the DIE bit, a receiver cannot decide instantly whether it receives a RTR or a SRR bit. That means only the IDE bit decides whether the frame is a Standard Frame or an Extended Frame.

The format of the Control Field is similar for Standard Format and Extended Format. The Control Field in Standard Format includes the Data Length Code (DLC), the IDE bit, which is transmitted dominant and the reserved bit r0 also transmitted dominant. The Control Field in Extended Format includes the DLC and two the reserved bits r1 and r0. The reserved bits have to be sent dominant, but receivers accept dominant and recessive bits in all combinations.

The full range of possible identifiers is not required to be implemented. The identifier range 2032 to 2048 are allowed to be used.



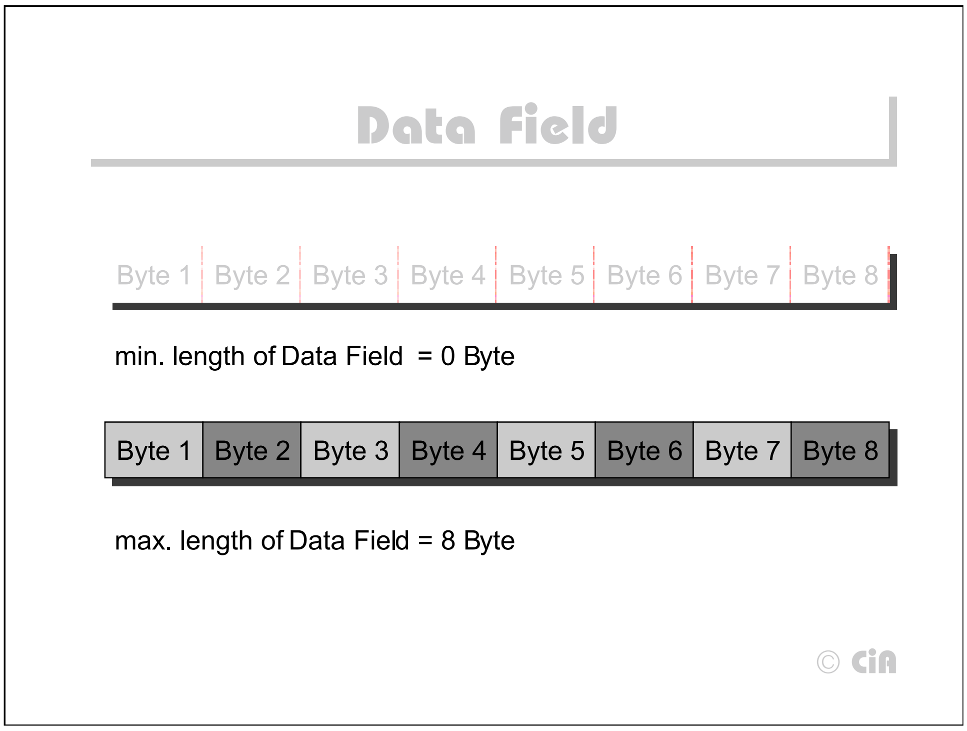
Control Field

RTR	IDE/r1	r0	DLC3	DLC2	DLC1	DLC0	Data/CRC

No. of Data Bytes	Data Length Code (DLC)			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d/r	d/r	d/r

© CiA

The number of bytes in the CAN Data Field is indicated by the Data Length Code (DLC) which is 4-bit wide and is transmitted in the Control Field. The admissible number of data bytes for a data frame ranges from 0 to 8. DLCs in the range of 0 to 7 indicate data filed length of 0 to 7 bytes. All other DLCs indicate that the data filed is 8 bytes long. That means DLCs ranging from 9 to 15 may be used for application-specific purposes. The full range of possible DLCs is not required to be implemented



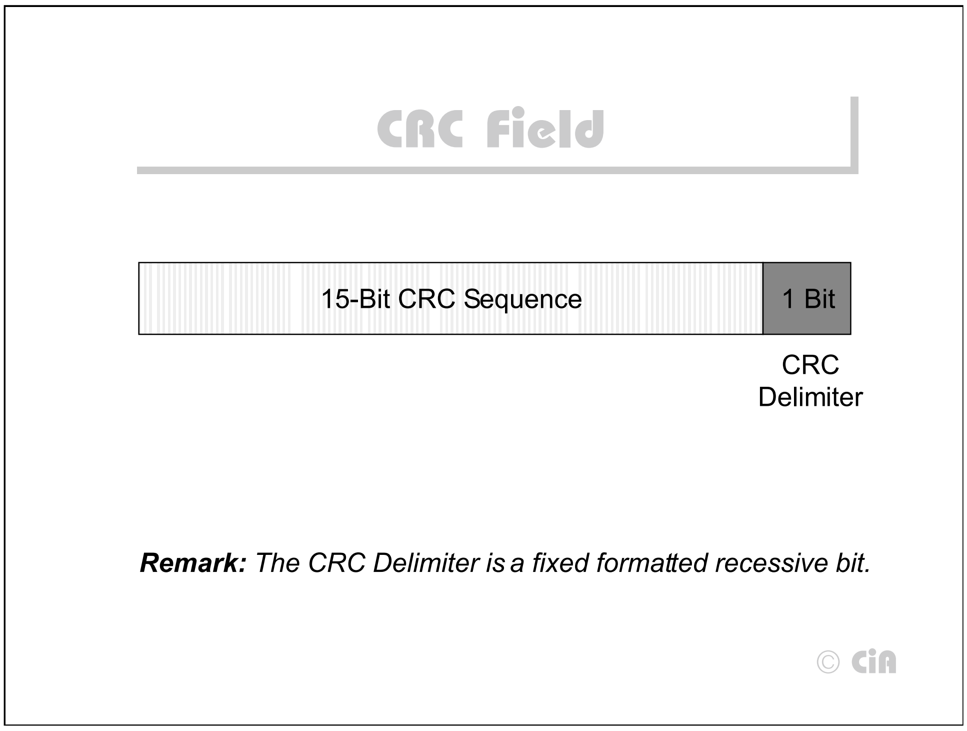
Data Field

min. length of Data Field = 0 Byte

max. length of Data Field = 8 Byte

© CiA

In some application it makes sense to transmit no data. For example, you only like to indicate an event without any variable data. The event is identified by the identifier of the Data Frame, so it may be sufficient to transmit no data field.



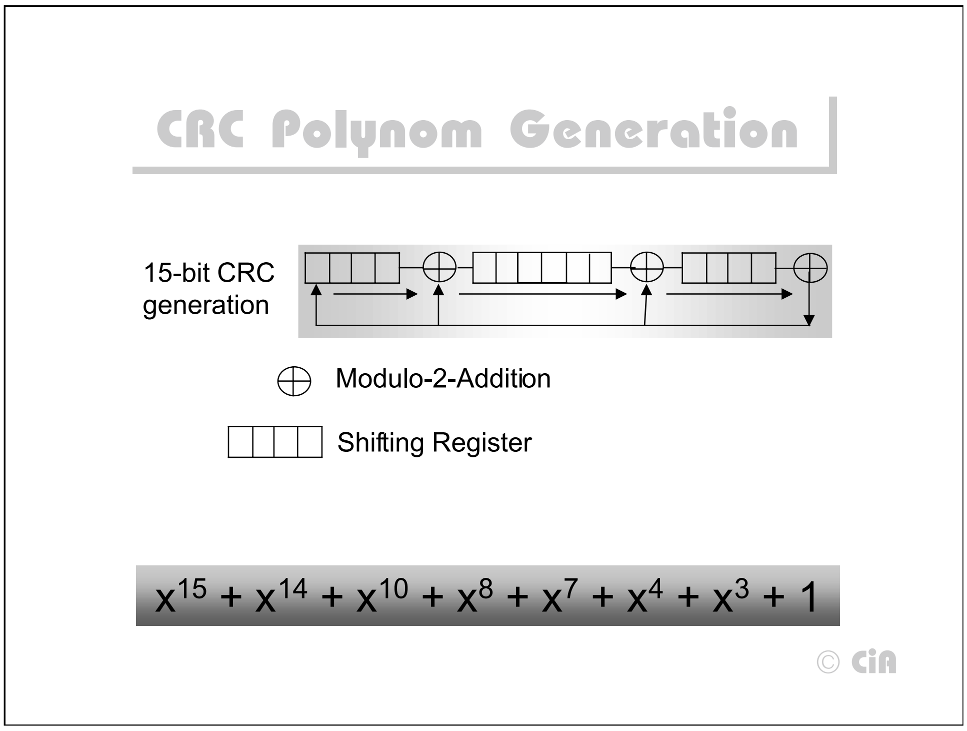
CRC Field

15-Bit CRC Sequence | 1 Bit

CRC Delimiter

***Remark:** The CRC Delimiter is a fixed formatted recessive bit.*

The CRC contains the 15-bit CRC sequence and the recessive 1-bit CRC Delimiter. The frame check sequence is derived from a cyclic redundancy code best suited for frames with bit counts less than 127 bit (BCH Code). The CRC code provides a Hamming distance of 6, meaning that 5 bit errors randomly distributed in the SOF, Arbitration, Control and Data Fields can be detected. In addition, burst errors up to a length of 15 bit can be detected.



CRC Polynom Generation

15-bit CRC generation

⊕ Modulo-2-Addition

Shifting Register

$$x^{15} + x^{14} + x^{10} + x^{8} + x^{7} + x^{4} + x^{3} + 1$$

© CiA

The receivers calculate the CRC in the same way as the transmitter as follows:

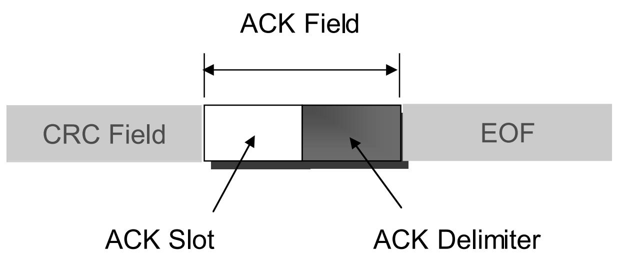
1. The message is regarded as polynom and is divided by the generator polynom: $x^{15} + x^{14} + x^{10} + x^{8} + x^{7} + x^{4} + x^{3} + 1$.

2. The division rest of this modulo2 division is the CRC sequence which is transmitted together with the message.

3. The receiver divides the message inclusive the CRC sequence by the generator polynom.

A CRC error has to be detected, if the calculated result is not the same as that received in the CRC sequence. In this case the receiver discards the message and transmits an Error Frame to request retransmission.

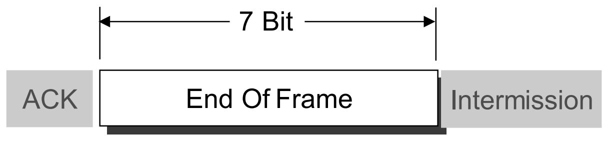
Acknowledge Field



© CiA

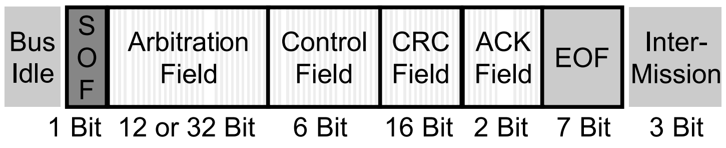
The ACK Field is two bits long and contains the ACK Slot and the ACK Delimiter. The transmitter of a frame transmits both bits of the ACK Field recessive. A Receiver, which has received a valid message correctly, reports this to the Transmitter by sending a dominant bit during the ACK Slot. If the Transmitter detects a positive acknowledge, that is a dominant ACK Slot , the Transmitter knows that in minimum one node has got his message correctly.

End of Frame (EOF)



Each Data and Remote Frame is delimited by a flag sequence of seven recessive bits. This EOF was introduced because an Error Frame caused by a global CRC failure should be transmitted within the length of Data or Remote Frame.

CAN Remote Frame

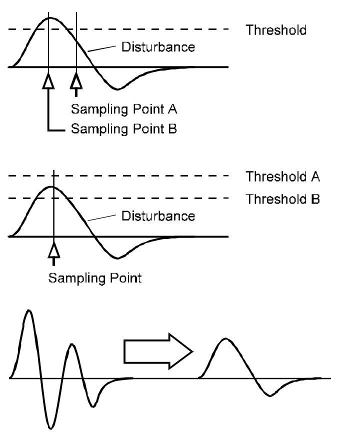


A destination node can request data from the source by sending a Remote Frame with an Identifier that matches the Identifier of the required Data Frame. The appropriate data source node will then send a Data Frame as a response to this remote request.

There are two differences between a Data Frame and a Remote Frame. Firstly the RTR-Bit is transmitted as a dominant bit in the Data Frame and as a recessive bit in the Remote Frame and secondly in the Remote Frame there is no Data Field. In the event of a Data Frame and a Remote Frame with the same identifier begin transmitting at the same time, the Data Frame wins arbitration due to the dominant RTR bit following the Identifier. So the node that transmitted the Remote Frame receives the desired data immediately.

Development of Local Error

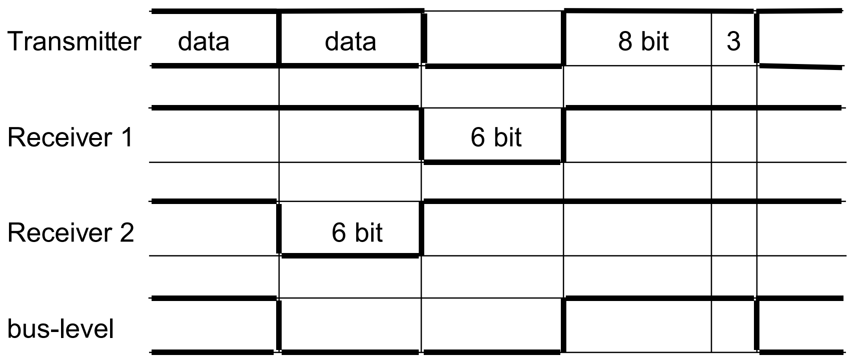
1. Different Sampling Points in different Nodes
2. Different Switching Thresholds in different Nodes
3. Dispersion during Propagation along the Bus Line



Local errors may occur because of several reasons such as sampling points or different switching thresholds or the signal dispersion during propagation along the bus lines. To guarantee network-wide data consistency local errors must be globalize.

In the CAN Data Link Layer protocol the stuff bit rule is used to indicate local errors by transmitting an Error Flag with 6 bits of the same polarity. So each node can detect this global Stuff error and react with sending themselves an Error Flag.

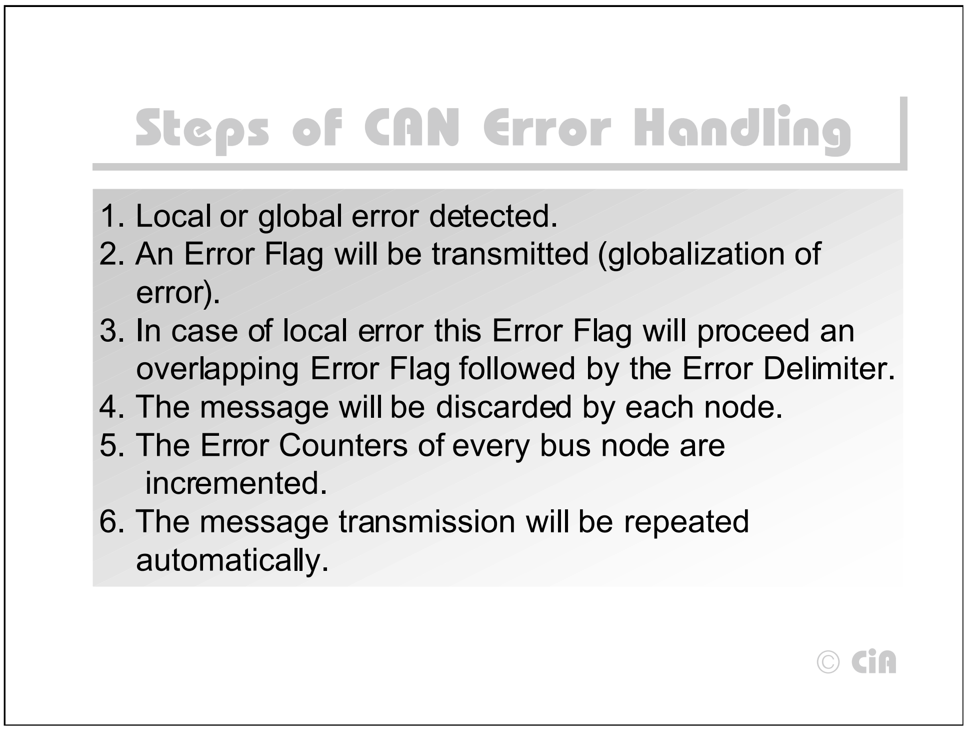
Globalization of Local Errors



The Receiver 2 detects an error and makes it public to the other nodes.

© CiA

The Receiver 2 detects an local error and transmits an Error Flag. At the 6th bit of the Error Flag all other connected stations recognize a violation of the bit stuffing rule and transmit Error Flags by themselves. After the Error Delimiter and Intermission, the transmitter tries again to access the bus to retransmit the corrupted message.



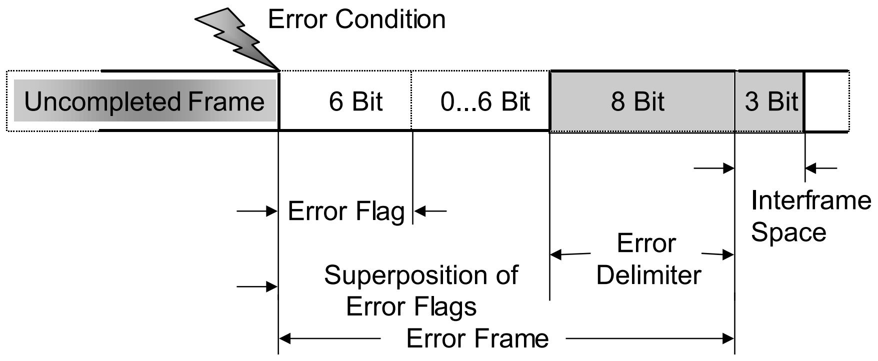
Steps of CAN Error Handling

1. Local or global error detected.
2. An Error Flag will be transmitted (globalization of error).
3. In case of local error this Error Flag will proceed an overlapping Error Flag followed by the Error Delimiter.
4. The message will be discarded by each node.
5. The Error Counters of every bus node are incremented.
6. The message transmission will be repeated automatically.

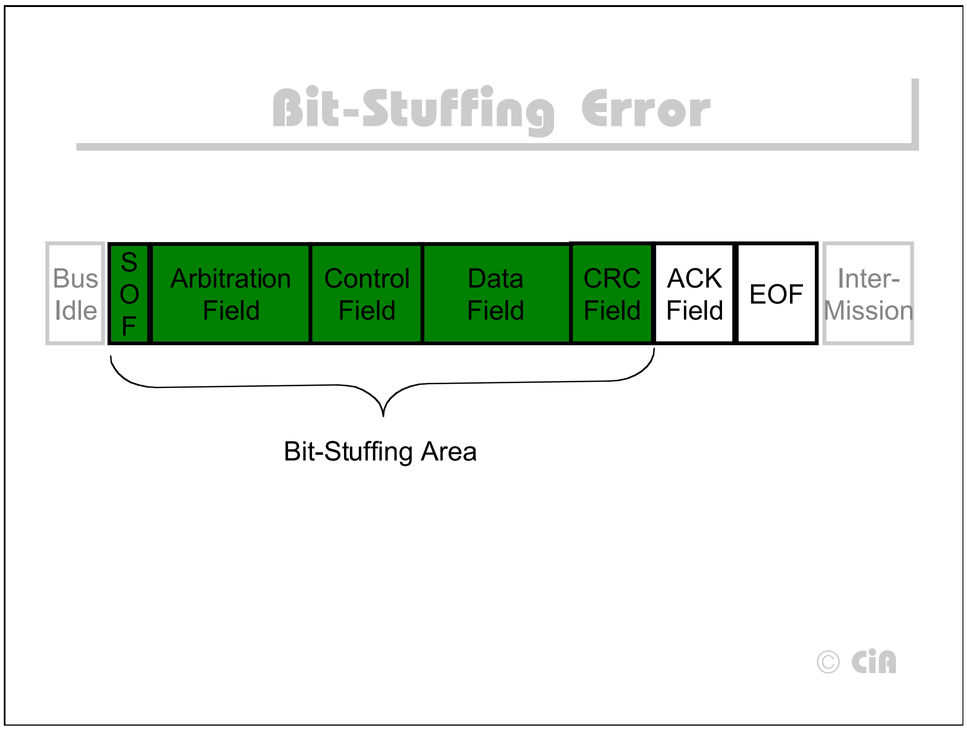
© CiA

By means of CAN Error Handling, network wide data consistency can be guaranteed if no node is in Error Passive state or in Bus Off state. In case of a local error, the failure will be globalized by sending an Error Flag consisting of 6 bits of the same polarity, which proceeds a bit stuffing error. After the Error Frame is finished, the node tries transmit again the discarded message. If no higher prior message like to access the bus, the discarded frame will be transmitted at least after 23 bit times. In the case of an Error Passive node the maximum failure recover time is 31 bit-times because of the suspend transmission (additional 8 bits).

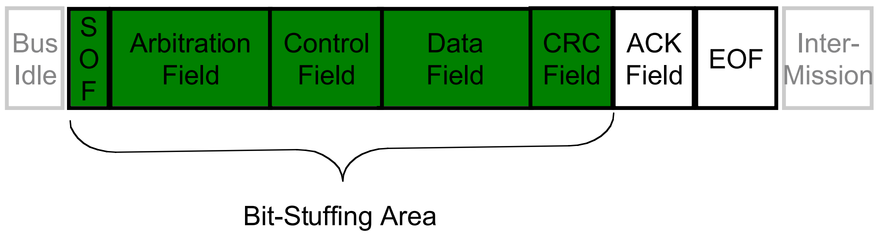
Active Error Frame



An Error Frame is generated by any node that detects a bus error. The Error Frame consists of an Error Flag and an Error Delimiter Field. The dominant bits of the Error Flag overwrite the corrupted data frame and cause a retransmission. Because of the mechanism of Error Globalization a Superposition of Error Flags can occur. The Error Delimiter consists of eight recessive bits and allows the bus nodes to restart bus communications cleanly after an error.



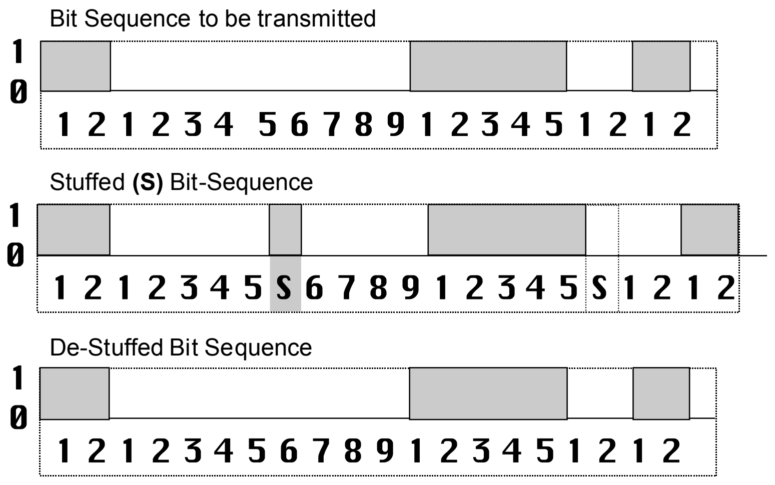
Bit-Stuffing Error



The remaining bit fields of the Data Frame or Remote Frame (CRC Delimiter, ACK Slot and EOF) are of fixed form and are not stuffed. The Error Frame and the Overload Frame are of fixed form as well and are not coded with Bit-Stuffing.

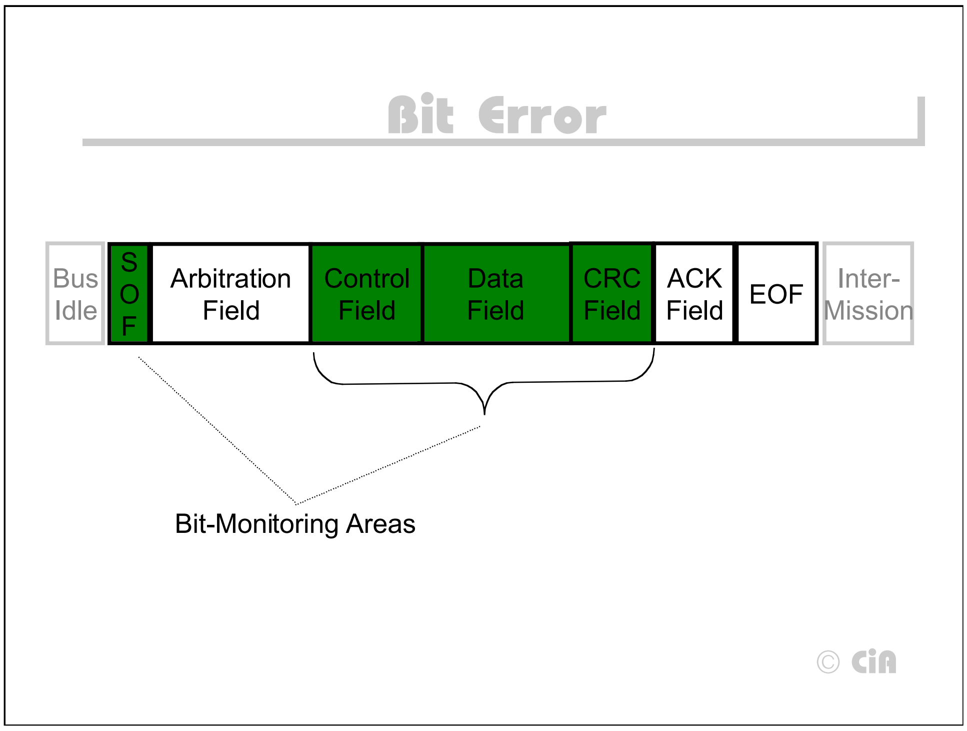
If the a node detects at the time of the 6th consecutive equal bit level during bit-stuffing area of a frame, it will generate an Error Frame starting with the next bit-time.

Bit-Stuffing and De-Stuffing



© CiA

Whenever a transmitter detects five consecutive bits (including stuff bits) of identical value in the bitstream to be transmitted, it automatically inserts a complimentary bit in the bitstream actually being transmitted. This stuff-bit will be removed automatically by the receiver. If more than five consecutive bits with the same polarity are detected between Start of Frame and the CRC Delimiter, the bit-stuffing rule has been violated. A Stuff-Error occurs and an Error Frame is generated. The message is then repeated.

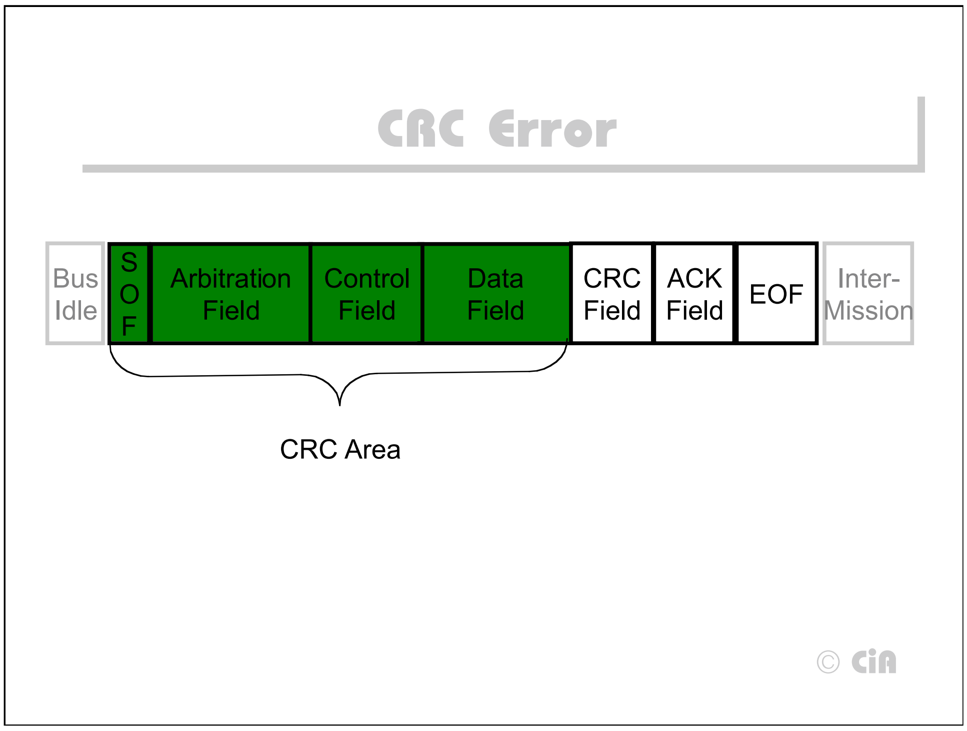


Bit Error

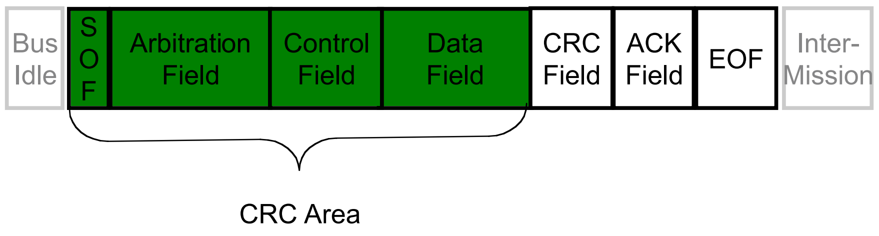
All nodes perform Bit-Monitoring: A Bit Error occurs if a transmitter sends a dominant bit but detects a recessive bit on the bus line or sends a recessive bit but detects a dominant bit on the bus line. An Error Frame is generated and starts with the next bit-time.

When a dominant bit is detected instead of a recessive bit, no error occurs during the Arbitration Field or the Acknowledge Slot because these fields must be able to be overwritten by a dominant bit in order to achieve arbitration and acknowledge functionality.

A Transmitter sending a Passive Error Flag and detecting a dominant bit does not interpret this as a Bit Error.

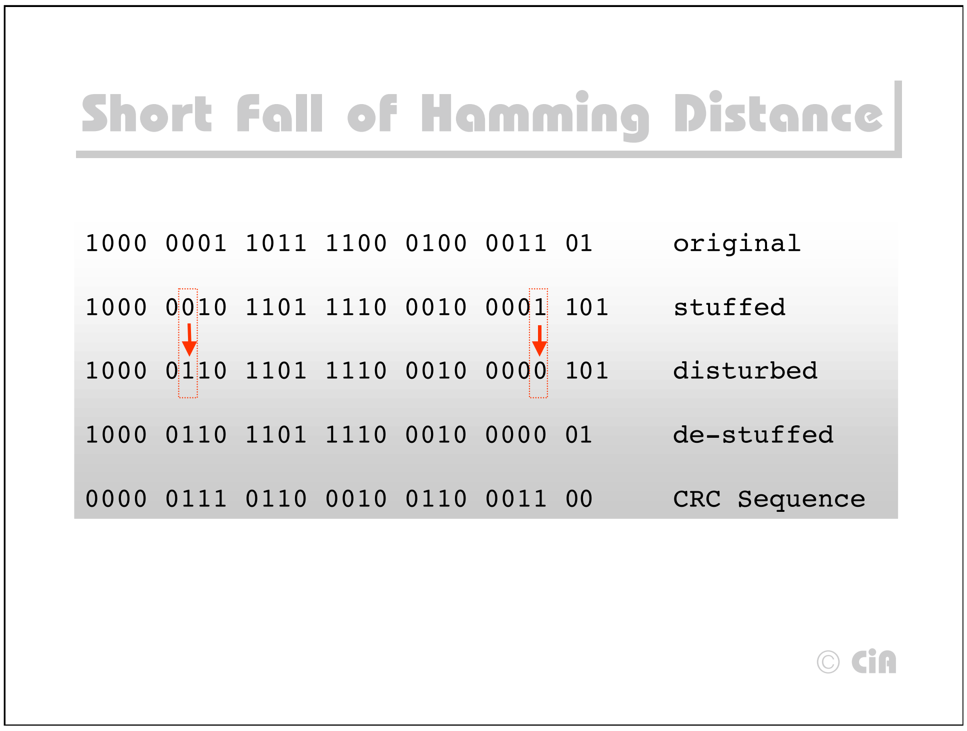


CRC Error



With the Cyclic Redundancy Check the Transmitter calculates a check sum for the CRC bit sequence from the Start of Frame bit until the end of the Data Field. This CRC sequence is transmitted in the CRC Field of the Data or Remote Frame. A CRC Error has to be detected, if the result calculated by the Receiver is not the same as that received in the CRC sequence. In this case the Receiver discards the message and transmits an Error Frame at the bit following the ACK Delimiter, unless an Error Flag for another error condition has already been started.

The CRC checksum will be used for error detection only. It is not used for error correction. The Hamming Distance of this CRC code is theoretically 6. With this it is possible to detect up to 5 single bit errors that are randomly scattered about the message or so-called burst errors up to a length of 15 bit.



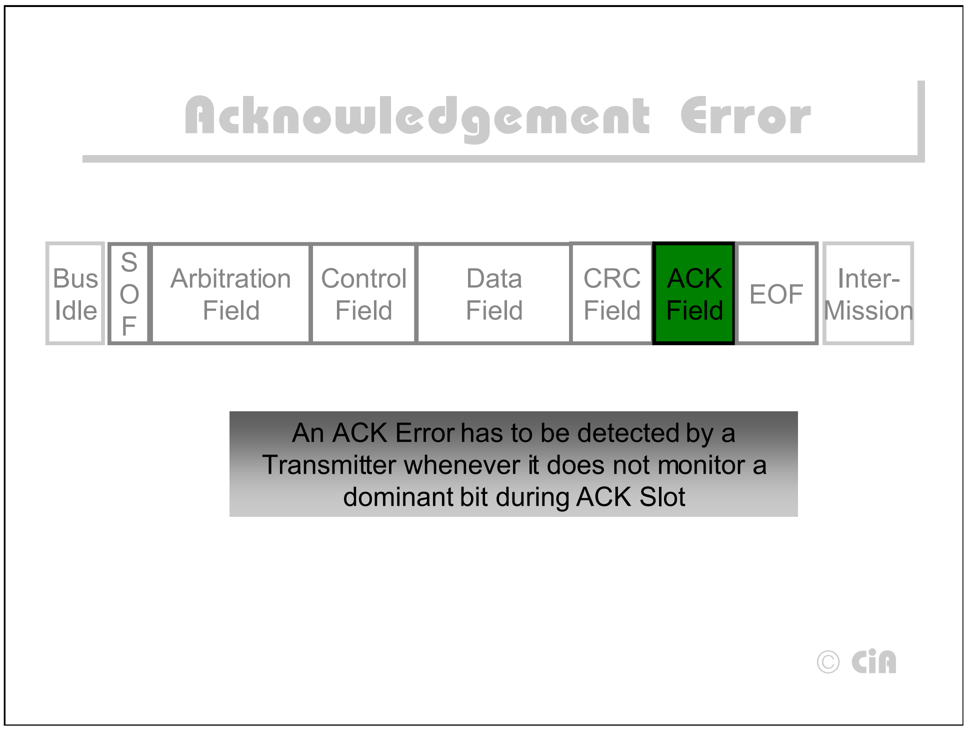
Short Fall of Hamming Distance

1000 0001 1011 1100 0100 0011 01	original
1000 0010 1101 1110 0010 0001 101	stuffed
1000 0110 1101 1110 0010 0000 101	disturbed
1000 0110 1101 1110 0010 0000 01	de-stuffed
0000 0111 0110 0010 0110 0011 00	CRC Sequence

© CiA

Theoretically the CRC polynom implemented in CAN guarantees a Hamming distance of 6 meaning that 5 randomly distributed bit errors will be detected. But there are some very seldom cases possible in which 2 locally bit failures can't be detected by the receiving CAN nodes.

The problem can be overcome by excluding objects with identifiers which produce stuff bits.



Acknowledgement Error

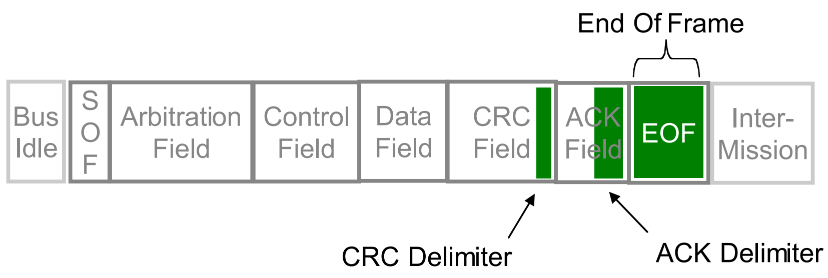
Bus Idle	S O F	Arbitration Field	Control Field	Data Field	CRC Field	ACK Field	EOF	Inter-Mission

An ACK Error has to be detected by a Transmitter whenever it does not monitor a dominant bit during ACK Slot

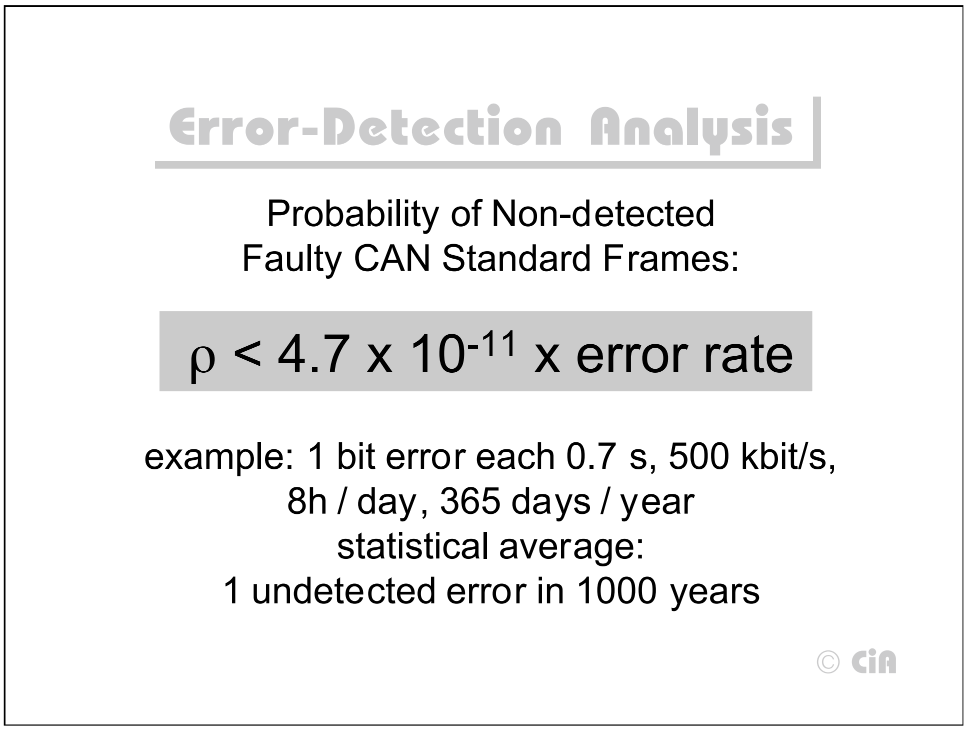
© CiA

With the Acknowledge Error Check the Transmitter checks the Acknowledge Slot, which is transmitted by the transmitting node as a recessive bit, if it contains a dominant bit. If this is the case, at least one other node has received the frame correctly. If not, an Acknowledge Error has occurred and the Transmitter will start in the next bit-time an Error Frame transmission.

Form Error



If a transmitter detects a dominant bit in one of the fix formatted segments CRC Delimiter, ACK Delimiter, End of Frame, a Form Error has occurred and an Error Frame is generated.



Error-Detection Analysis

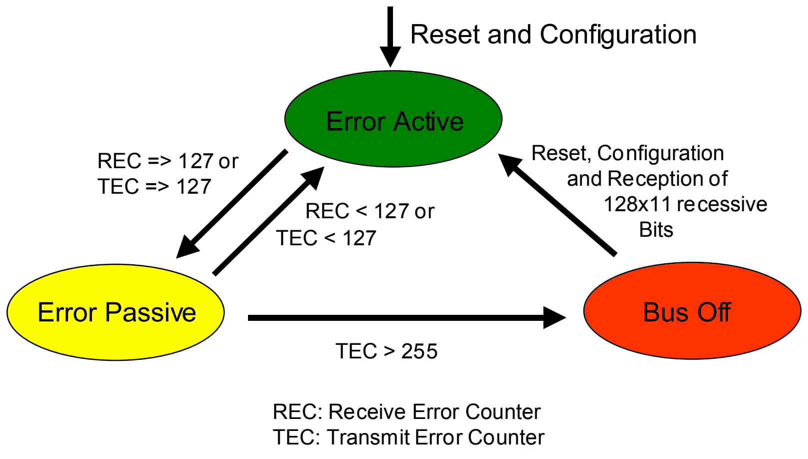
Probability of Non-detected
Faulty CAN Standard Frames:

$$\rho < 4.7 \times 10^{-11} \times \text{error rate}$$

example: 1 bit error each 0.7 s, 500 kbit/s,
8h / day, 365 days / year
statistical average:
1 undetected error in 1000 years

The probability of non-detected faulty messages is subject of several research projects, which are reported in the literature especially on the international CAN Conferences. This probability depends on several parameters such as the average corrupted message rate, which is about 10^{-3} for standard cables. In general, the probability of non-detected faulty messages is higher for Extended Frames as for Standard Frames.

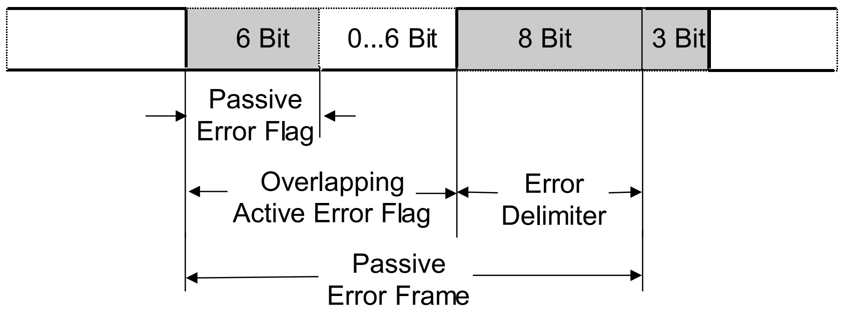
Error States of CAN Node



To distinguish between temporary and permanent failures every CAN controller has two Error Counters: The REC (Receive Error Counter) and the TEC (Transmit Error Counter). The counters are incremented upon detected errors respectively are decremented upon correct transmissions or receptions. Depending on the counter values the state of the node is changed: The initial state of a CAN controller is Error Active that means the controller can send active Error Flags. The controller gets in the Error Passive state if there is an accumulation of errors.

On CAN controller failure or an extreme accumulation of errors there is a state transition to Bus Off. The controller is disconnected from the bus by setting it in a state of high-resistance. The Bus Off state should only be left by a software reset. After software reset the CAN controller has to wait for 128 x 11 recessive bits to transmit a frame. This is because other nodes may pending transmission requests. It is recommended not to start an hardware reset because the wait time rule will not be followed then.

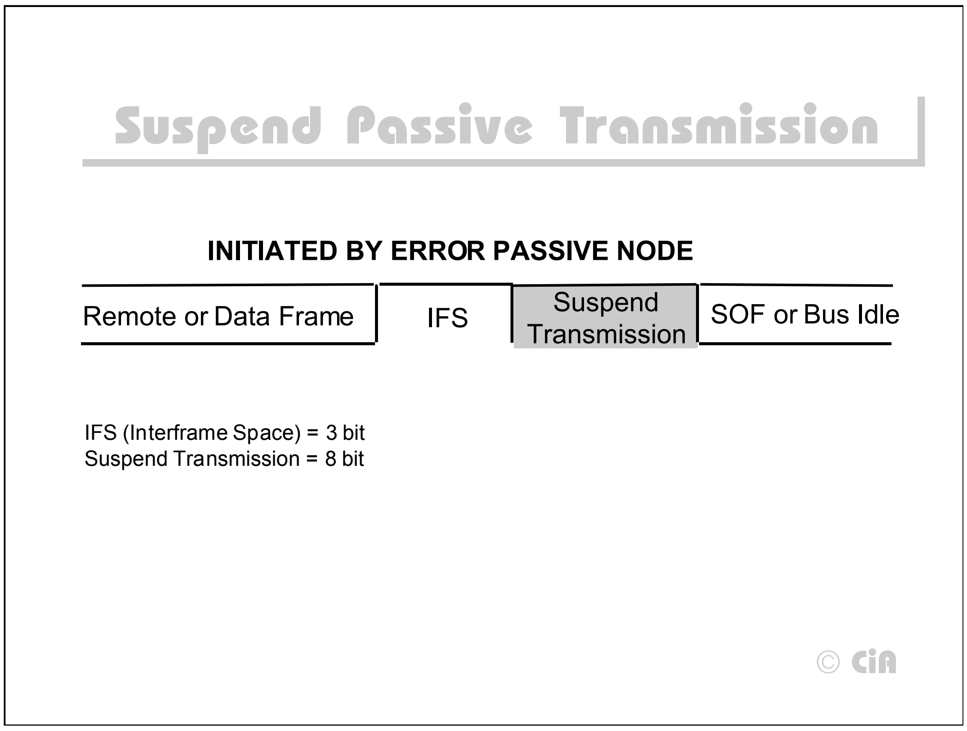
Passive Error Frame



In order to prevent the bus from being blocked by the Error Frames sent from a faulty or heavily disturbed node, the error indication ability can be restricted for this node. An Error Active node can send a dominant Error Flag while an Error Passive node is only able to send a recessive one.

During the transmission of the Error Delimiter a node can detect whether it was the first one in the network reporting the error and thus stopping the data transfer.

Error Passive receivers can no longer interrupt the data transfer as a recessive Error Flag does not influence the bus levels. An Error Passive transmitter can still interrupt its own message by sending a passive Error Flag. Attention, if one Receiver is in error passive mode no data consistency is guaranteed any more!



Suspend Passive Transmission

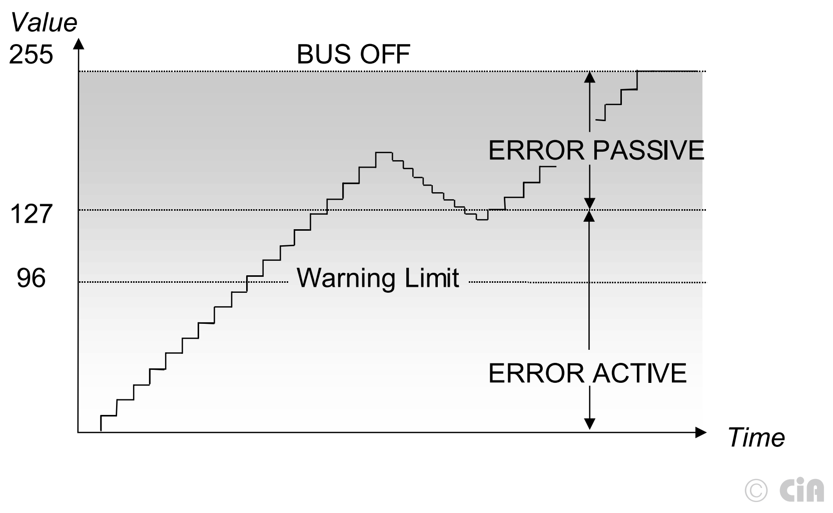
INITIATED BY ERROR PASSIVE NODE

Remote or Data Frame
IFS
Suspend Transmission
SOF or Bus Idle

IFS (Interframe Space) = 3 bit
Suspend Transmission = 8 bit

To avoid blocking the bus by a disturbed node sending high priority messages a transmission delay was introduced for nodes in Error Passive state. After transmission an Error Passive node must wait 3 + 8 recessive bits before starting transmission again (additional wait state).

Transmit Error Counter



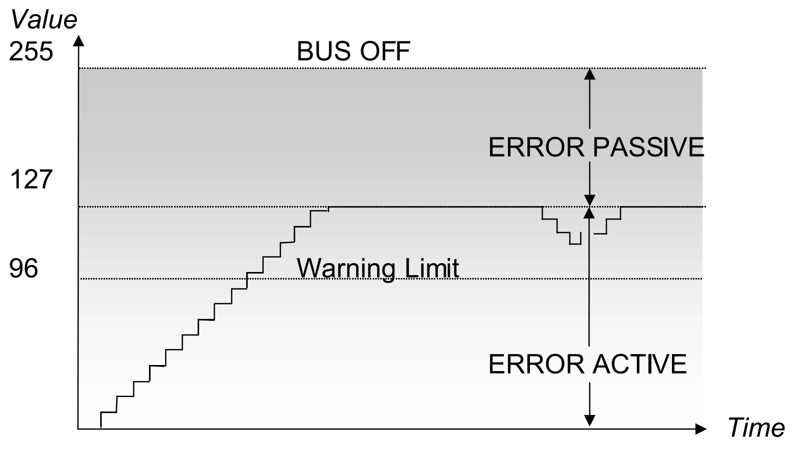
0 - 96: The node is in the Error Active state below the warning limit. Exceeds the counter a value of 96 the CAN controller generates a warning (set error flag, generating interrupt).

97 - 127: The node is Error Active. If the counter is in this area it must be assumed that the bus is heavily disturbed.

128 - 255: In this area the node is Error Passive. Unfortunately some of the CAN controllers do not inform the microcontroller about this change of state.

> 255: After reaching this counter value the node is switched to Bus Off state. In this state it does not take part in bus activities. The microcontroller is informed by an interrupt on this change of states.

Receive Error Counter

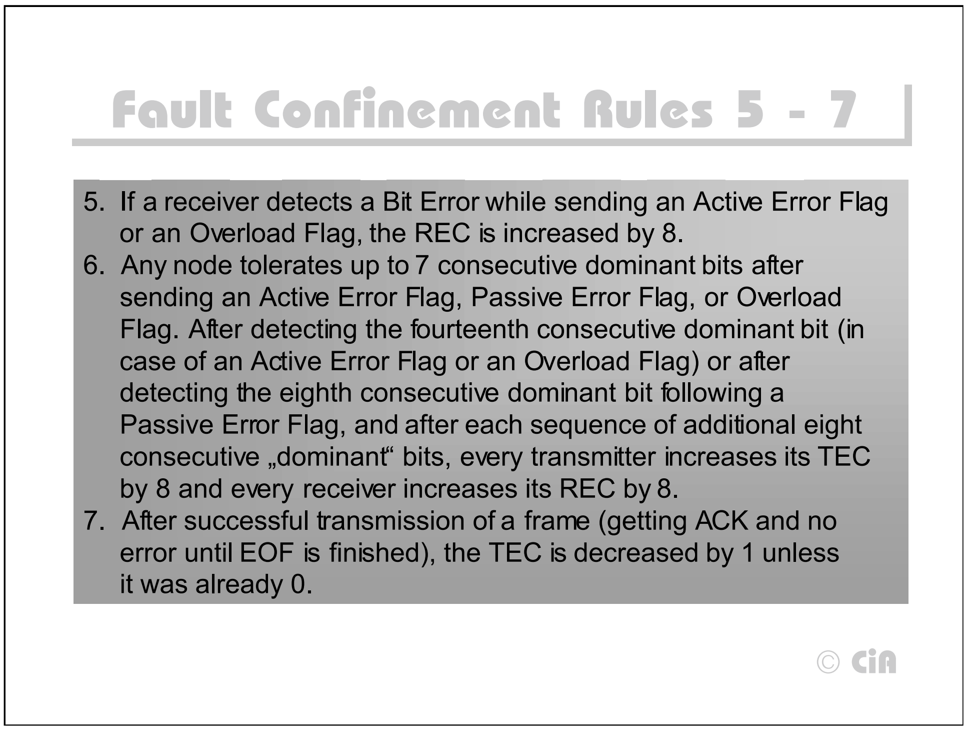


Theoretically, the CAN Fault Confinement rules could increment REC's value over all limits when a receiver in Error Passive mode detects additional errors without receiving any error-free message. This cannot be implemented in hardware and the REC's value is limited by its actual number of digits. In the Bosch Reference Model, the REC has a resolution of 8 bits. So REC cannot be increased in Passive Error mode.

Fault Confinement Rules 1 - 4

1. When a receiver detects an error, the REC will be increased by 1, except when the detected error was a Bit Error during the sending of an Active Error Flag or an Overload Flag.
2. When a receiver detects a dominant bit as the first bit after sending an Error Flag, the REC will be increased by 8.
3. When a transmitter sends an Error Flag, the TEC is increased by 8.
   Exception 1: If the transmitter is Error Passive and detects an ACK Error because of not detecting a dominant ACK and does not detect a dominant bit while sending its Passive Error Flag.
   Exception 2: If the transmitter sends an Error Flag because a Stuff Error occurred during arbitration, and should have been recessive, and has been sent as recessive but monitored as dominant.
4. If the transmitter detects a Bit Error while sending an Active Error Flag or an Overload Frame, the TEC is increased by 8.

© CiA



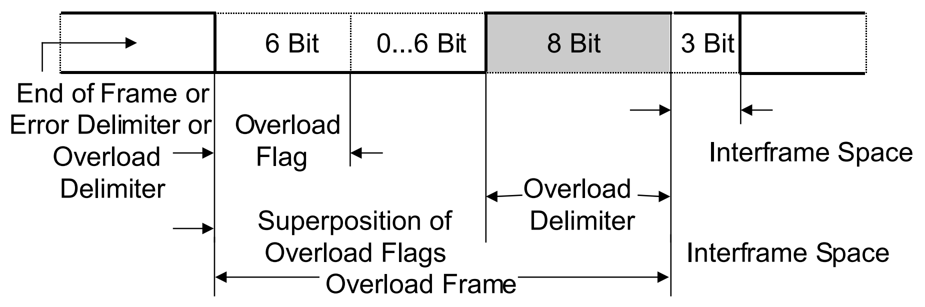
Fault Confinement Rules 5 - 7

5. If a receiver detects a Bit Error while sending an Active Error Flag or an Overload Flag, the REC is increased by 8.
6. Any node tolerates up to 7 consecutive dominant bits after sending an Active Error Flag, Passive Error Flag, or Overload Flag. After detecting the fourteenth consecutive dominant bit (in case of an Active Error Flag or an Overload Flag) or after detecting the eighth consecutive dominant bit following a Passive Error Flag, and after each sequence of additional eight consecutive „dominant“ bits, every transmitter increases its TEC by 8 and every receiver increases its REC by 8.
7. After successful transmission of a frame (getting ACK and no error until EOF is finished), the TEC is decreased by 1 unless it was already 0.

Fault Confinement Rules 8 - 12

8. After the successful reception of a frame (reception without error up to the ACK Slot and the successful sending of the ACK bit), the REC is decreased by 1, if it was between 1 and 127. If the REC was 0, it stays 0, and if it was greater than 127, then it will be set to a value between 119 and 127.
9. A node is Error Passive when the TEC equals or exceeds 128, or when the REC equals or exceeds 128. An error condition letting a node become Error Passive causes the node to send an Active Error Flag.
10. A node is Bus Off when the TEC is greater than or equal to 256.
11. An Error Passive node becomes Error Active again when both the TEC and the REC are less than or equal to 127.
12. A node which is Bus Off is permitted to become Error Active (no longer Bus Off) with its error counters both set to 0 after 128 occurrences of 11 consecutive recessive bits have been monitored on the bus.

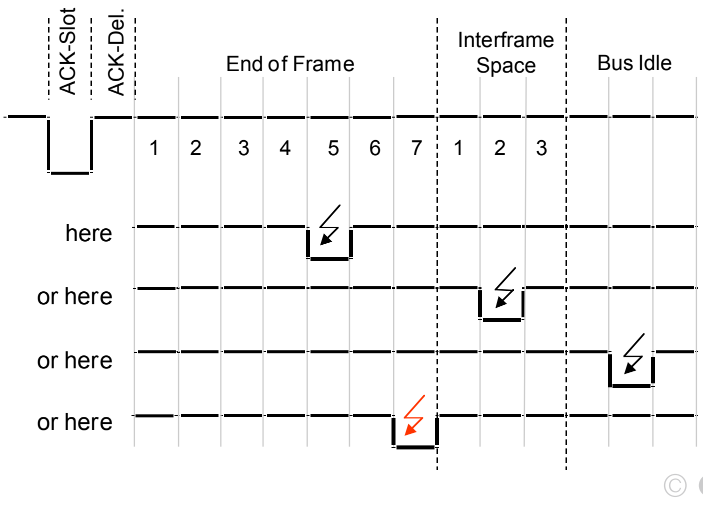
Overload Frame



An Overload Frame can be generated by a node if due to internal conditions the node is not yet able to start reception of the next message or if during Intermission one of the first two bits is dominant. Another overload condition is that a message is valid for receivers, even when the last bit of EOF is recei ved as dominant. Therefore, this dominant bit is not regarded as an error. On the other hand, the fixed-form bit filed EOF contains an illegal bit and the receiver of the dominant bit may have lost synchronization, which requires a reaction. The Reference CAN Model follows the example of a dominant bit as the last bit of Error or Overload Delimiter which are responded with an Overload Frame.

With an Overload Frame the transmitter is requested to delay the start of the next transmission. The Overload Frame is identical to an Active Error Frame. The only difference is that an Overload Frame does not increase the error counters (see error confinement) and does not causes a retransmission of a frame. Every node may transmit consecutively only 2 Overload Frames.

Local Errors in EOF



If one of the EOF (End of Frame) bits 1 to 6 are detected locally as dominant bits the node will send an Error Flag to globalize this failure.

The CAN specification reads as follows:

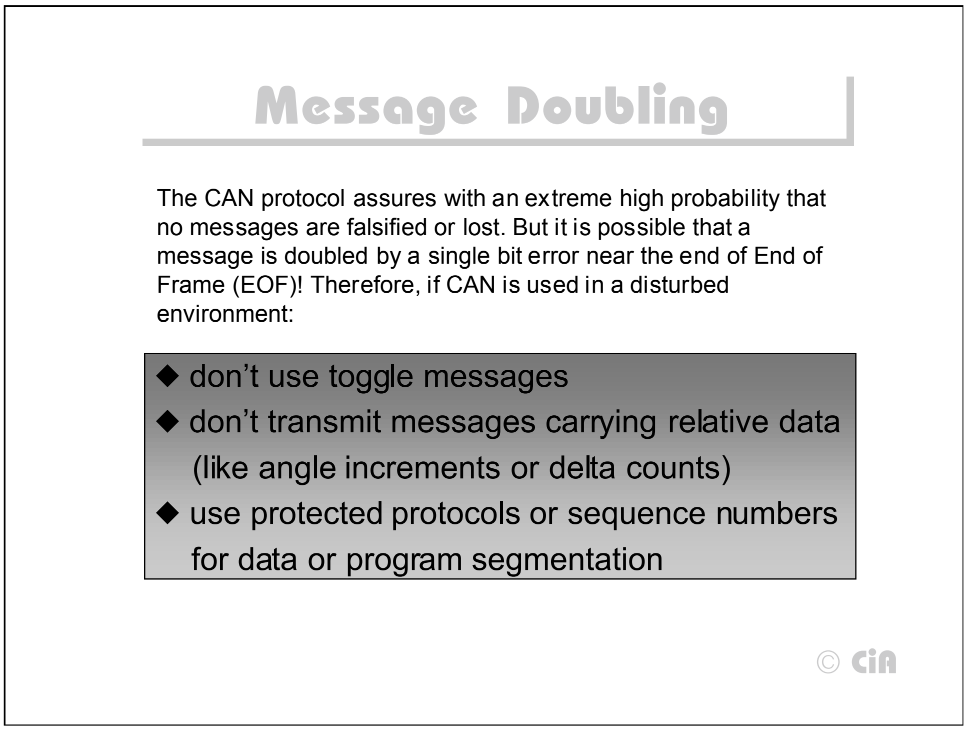
The point of time at which a message is taken to be valid is different for the transmitter and the receivers of the message.

• Transmitter:The message is valid for the transmitter, if there is no error until the end of End of Frame. If a message is corrupted, retransmission will follow automatically.

• Receiver: The message is valid for the receiver, if there is no error until the last but one bit of End of Frame.

A Receiver, which has sampled a dominant value during the 7th EOF bit do not regard this as an error. On the other hand, the fixed-form bit contains an illegal bit and the Receiver may have lost synchronization, therefore an Overload Frame is transmitted.

As the Receiver can inform the sender at the earliest during the next bit time, it’s obvious that the Transmitter must wait one additional bit time for his message validation. This is independent of which bit is defined as the validation bit. It does not help to introduce (one or more) additional bits at the end of the message frame.



Message Doubling

The CAN protocol assures with an extreme high probability that no messages are falsified or lost. But it is possible that a message is doubled by a single bit error near the end of End of Frame (EOF)! Therefore, if CAN is used in a disturbed environment:

- ◆ don’t use toggle messages
- ◆ don’t transmit messages carrying relative data (like angle increments or delta counts)
- ◆ use protected protocols or sequence numbers for data or program segmentation

© CiA

If a Transmitter samples a dominant bus level during the last bit of EOF it must retransmit that message. The dominant bus level can result from:

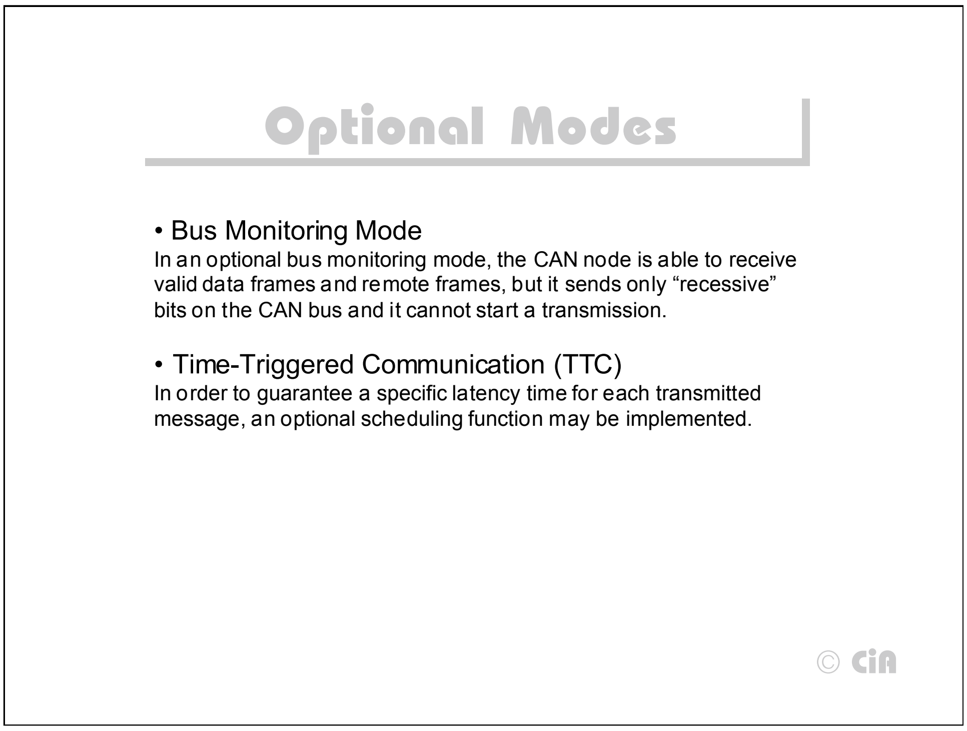
1. a receivers Error Flag reporting a local error in the last but one bit of EOF
2. a disturbance of the last bit of EOF

In case 1 it’s likely that there are other receivers which have not been affected by the local error and therefore have already accepted the first message.

In case 2 all receivers have already accepted the first message. After the retransmission they have got the same message twice.

ISO 11898 Review

• Restructured documents (part 1 and part 2)
• DLCs larger than 8 are allowed
• SRR with dominant value is ignored
• Overload frame transmission in case of dominant last bit of EOF
• Hard Synchronization at every edge from recessive-to-dominant
• No restriction of the identifier range (0 to 2047)
• Optional Bus monitoring mode
• Optional time-capturing (triggering at SOF and last bit of EOF)



Optional Modes

• Bus Monitoring Mode
In an optional bus monitoring mode, the CAN node is able to receive valid data frames and remote frames, but it sends only “recessive” bits on the CAN bus and it cannot start a transmission.

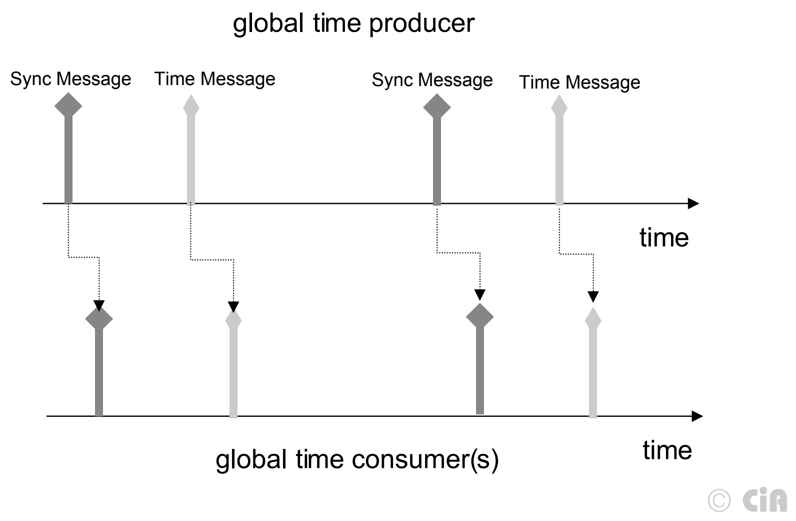
• Time-Triggered Communication (TTC)
In order to guarantee a specific latency time for each transmitted message, an optional scheduling function may be implemented.

© CiA

If the optional bus monitoring mode is supported, the MAC sub-layer is required to transmit “dominant” bits (ACK bit, overload flag, active error flag), but the bit is rerouted internally so that the MAC sub-layer monitors these “dominant” bits, although the CAN bus may remain in recessive state.

The use of the time-triggered communication option requires a single-shot mode. In this mode the CAN node do not transmit overload and error flags. The automatic retransmission is disabled.

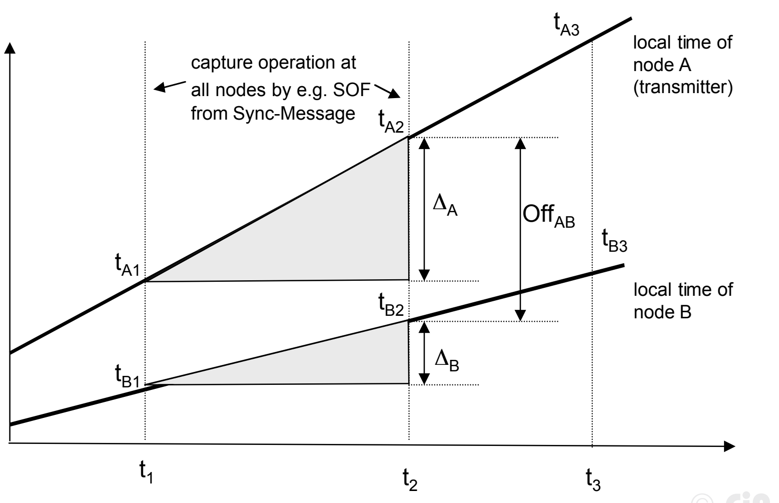
TTC Option



The time-triggered communication (TTC) option describes the prerequisites needed for the synchronization of all nodes in the network. With the synchronization of node any message can be transmitted at a specific time slot, where it has not to compete for the bus with other messages thus providing predictable latency times by avoiding the loss of arbitration. In order to synchronize the activities of the nodes within the network a common reference point is needed. The SOF bit or the sample point of the last bit of EOF of any message is used as the reference point. The individual presence of a single message at a time is referred to as frame scheduling. Based on the synchronization of the nodes the TTC facilitates also the establishment of a global time system in higher-layer protocols. The hardware needed to establish TTC is included between LLC and MAC.

Any node that supports TTC option needs to provide a time base, which is a cyclic up counter of at least 16 bit with either an internal or an external clock. Any message received or transmitted invokes a capture of the time base taken at the SOF recognition of the respective message or at the sample point of the last bit of EOF. After successful message reception, the capture value is provided to the CPU for at least one message and it is readable unitl the next message is received. It has to be possible to generate at least one programmable event trigger from the above mentioned time base. The trigger should be freely programmable by the CPU in the range of at least 0 to (2^{16}-1) x timer clocks.

Global Time System



Transformation Equations

$t_A = \Delta_{AB} (t_B - t_{B2}) + Off_{AB}$ transformation from node B into node A

$t_B = \Delta_{BA} (t_A - t_{A2}) + Off_{BA}$ transformation from node A into node B

with $\Delta_{AB} = \dfrac{1}{\Delta_{BA}} = \dfrac{\Delta_A}{\Delta_B} = \dfrac{(t_{A2} - t_{A1})}{(t_{B2} - t_{B1})}$

$Off_{AB} = - Off_{BA} = t_{A2} - t_{B2}$