Bus Systems

Automotive Bus Systems - CAN

Prof. Dr. Reinhard Gotzhein, Dr. Thomas Kuhn

Automotive Bus Systems

Learning objectives

- CAN Bus
  - Understand physical principles of CAN bus frame transmissions
  - Understand frame formats and CAN Bus arbitration
  - Understand CAN bus error states and failure modes

- CAN Bus Analysis
  - Know the preequisites for response time analysis
  - Know the difference between sufficient and exact analysis techniques
  - Learn to apply CAN bus scheduling analysis

Automotive Bus Systems

CAN Bus – Introduction

- Controller Area Network (CAN)
  - Standardized communication network
  - Specification publically available
    - http://www.semiconductors.bosch.de/pdf/can2spec.pdf

- Serial data bus developed by Robert Bosch GmbH in the 80s
  - Initially developed for automotive systems and commercial vehicles
    - Today, it is also used in avionics, plant engineering, building automation…
  - Multi master system – multiple nodes may initiate transmissions
  - Reliable and predictable communication
    - Deterministic bus access timing is supported
    - Automatic handling of node and hardware failures
    - Handling of broken cables
  - Supports broadcast and multicast communication
  - Low cost network

Automotive Bus Systems

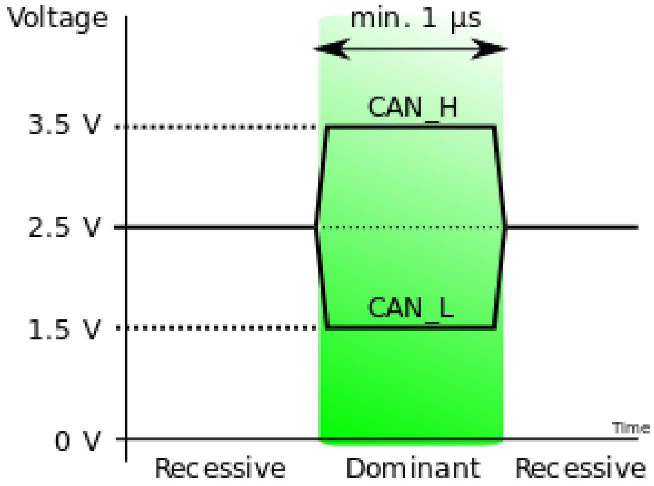
CAN Bus – Hardware

- Simple, differential (copper) wire connection
  - Uses two or three wire copper conenctions
    - CAN_H and CAN_L transmit different voltage levels
    - Optional CAN_GND
  - Cables are twisted to reduce transmission error probability
    - Low speed CAN bus netwoerks may use car-body as ground
    - Fall-back mechanism for automotive low-speed CAN bus
  - Bits are represented by voltage current levels
    - NRZ (Non return to zero) Coding

Automotive Bus Systems

CAN Bus – Physical Layer

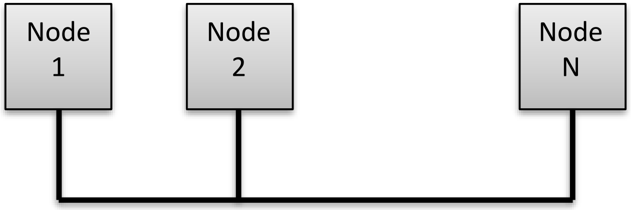
- Transmission of individual bits



Automotive Bus Systems

CAN Bus – Physical Layer

- CAN Bus topology
  - Line topology
  - Physical medium is shared between nodes



- What happens if multiple nodes are transmitting at the same time?
  - CAN Bus applies CSMA/CR
  - Carrier Sense Multiple Access / Collission Resolution
    - Defined behavior to resolve concurrenly pending transmission
    - Based on defined behavior for concurrently transmitted (colliding) bits

Automotive Bus Systems

CAN Bus – Physical Layer

- Concurrent transmission of individual bits
  - One or more nodes may transmit bits at the same time
  - E.g. during bus arbitration phase, when bus access is worked out between nodes

- Remember: CAN Bus supports transmission of two bit levels
  - Logical 0 bits are dominant bits
  - Logical 1 bits are recessive bits
  - When two bit levels are transmitted concurrently, dominant bits overwrite recessive bits
    - CAN Bus implements logical AND for concurrently transmitted bits

- CAN Bus nodes can detect collisions
  - Node transmits bit on CAN bus
  - Node checks bit level on CAN bus and compares with transmitted level

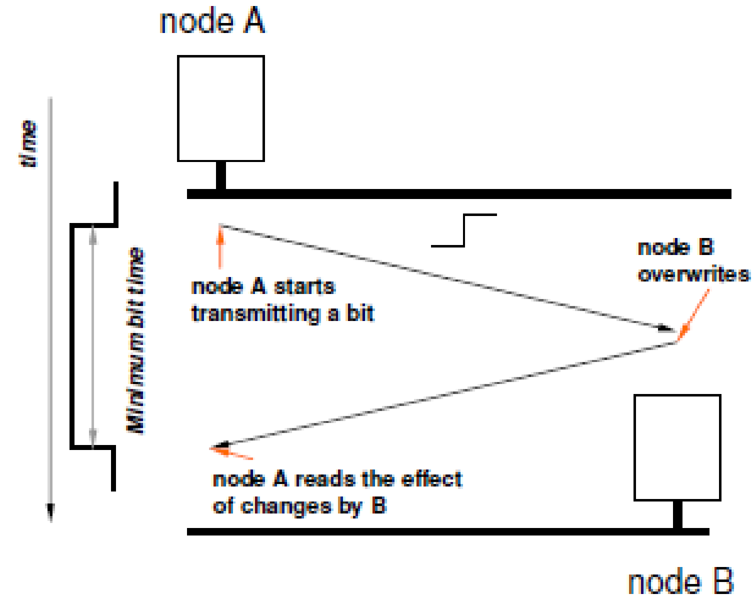
Automotive Bus Systems

CAN Bus – Physical Layer

- CAN Bus collision resolution yields bit timing requirements
  - Bit transmission over CAN requires two times the propagation delay
- Why?
  - Example:

Automotive Bus Systems

CAN Bus – Physical Layer



Automotive Bus Systems

CAN Bus – Physical Layer

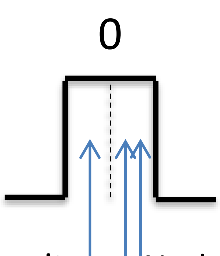
- CAN Bus minimum bit timings and resulting maximum transmission rates
  - Depend on bus length

Bit rate	Bus length
1 Mbit/s	25 m
800 kbit/s	50 m
500 kbit/s	**100 m**
250 kbit/s	250 m
125 kbit/s	500 m
50 kbit/s	1000 m
20 kbit/s	2500 m
10 kbit/s	5000 m

Automotive Bus Systems

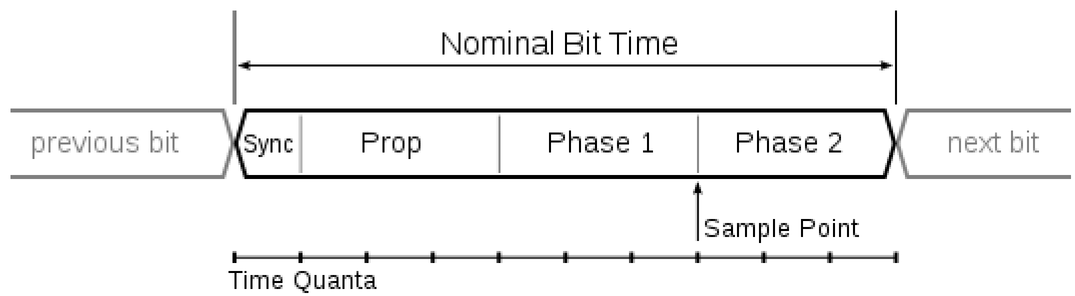
CAN Bus – Physical Layer

- CAN Bus collision resolution requires time synchronization between bus nodes
  - Nodes must transmit bits at the same time with maximum synchronization error
  - All nodes suffer from clock drift
    - Periodic re-synchronization is necessary to counter clock drift



Node transmits bit value, bus voltage level becomes stable

Node samples bit value multiple times



Automotive Bus Systems

CAN Bus – Physical Layer

- CAN Bus collision resolution requires time synchronization between bus nodes
  - Nodes must transmit bits at the same time with maximum synchronization error
  - All nodes suffer from clock drift
    - Periodic re-synchronization is necessary to counter clock drift

Clocks are re-synchronized whenever a change is detected

Automotive Bus Systems

CAN Bus – Physical Layer

- CAN Bus synchronization depends on bit value changes
  - Unlike Manchester coding, NRZ does not support automatic clock synchronization
  - Sufficient amount of bit value changes must be guaranteed by CAN

- Bit Stuffing
  - CAN controllers insert one bit of opposite level after five continuous bits of the same level when transmitting a bit sequence
  - Receiving CAN controllers filter these stuffing bits
  - This guarantees a minimum amount of bit shifts and therefore minimum level of clock synchronization
  - In some cases, bit stuffing is disabled for transmissions, e.g. when signalling error states

Automotive Bus Systems

CAN Bus – Communication Protocol

- CAN Bus protocol
  - Fixed format messages with limited size
    - 0-8 bytes of payload per message

- CAN communication does not require node (or system) addresses
  - Simple network configuration
  - Flexibility – a node can be added at any time
  - Message delivery and routing – the content is identified by an IDENTIFIER field defining the message content
  - Multicast – all messages are received by all nodes that can filter messages based on their IDs
  - Data Consistency – A message is accepted by all nodes or by no node

Automotive Bus Systems

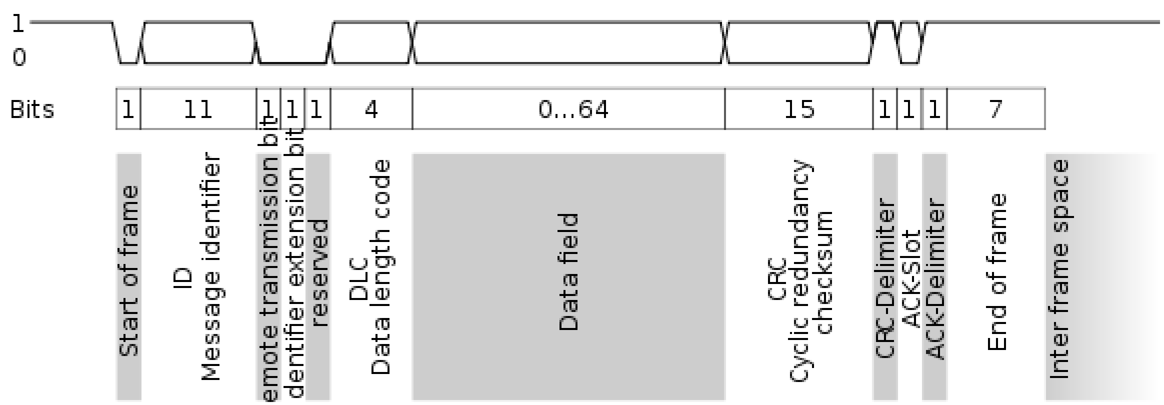
CAN Bus – Communication Protocol

- Frame types
  - DATA FRAME
    - Carries regular data
  - REMOTE FRAME
    - Used to request the transmission of a DATA FRAME with the same ID from another node
  - ERROR FRAME
    - Transmitted by any unit detecting a bus error
  - OVERLOAD FRAME
    - Used to force a time interval in between frame transmissions

Automotive Bus Systems

CAN Bus – Communication Protocol

- Data Frames (Basic Frame Format)



Automotive Bus Systems

CAN Bus – Communication Protocol

- Data Frames (Basic Frame Format)

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier (green)	11	A (unique) identifier for the data which also represents the message priority
Remote transmission request (RTR)	1	Dominant (0) (see Remote Frame below)
Identifier extension bit (IDE)	1	Declaring if 11 bit message ID or 29 bit message ID is used. Dominate (0) indicate 11 bit message ID while Recessive (1) indicate 29 bit message.
Reserved bit (r0)	1	Reserved bit (it must be set to dominant (0), but accepted as either dominant or recessive)
Data length code (DLC)* (yellow)	4	Number of bytes of data (0–8 bytes)
Data field (red)	0–64 (0-8 bytes)	Data to be transmitted (length in bytes dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

Automotive Bus Systems

CAN Bus – Communication Protocol

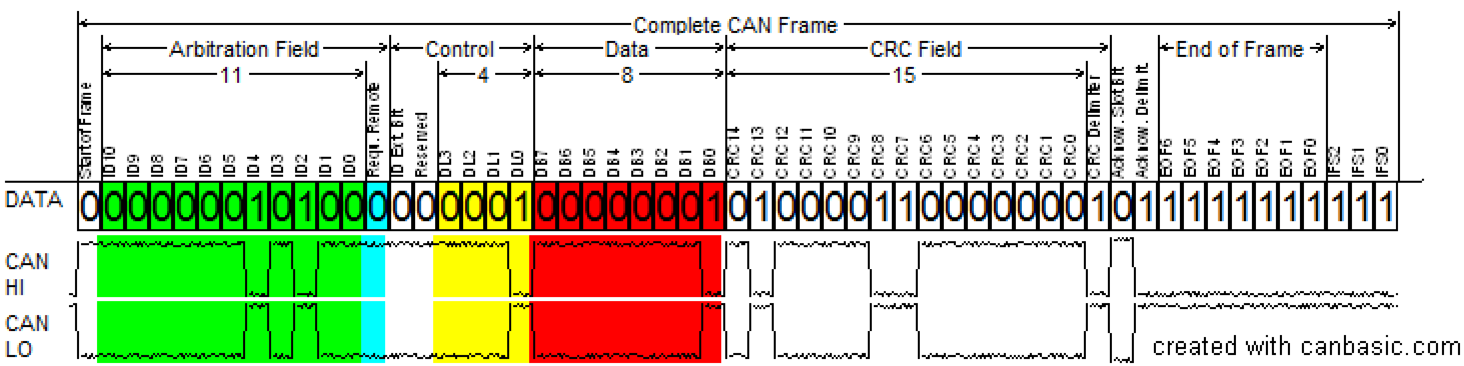
- Data Frames (Extended Frame Format)

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier A	11	First part of the (unique) identifier for the data which also represents the message priority
Substitute remote request (SRR)	1	Must be recessive (1). Optional
Identifier extension bit (IDE)	1	Must be recessive (1). Optional
Identifier B	18	Second part of the (unique) identifier for the data which also represents the message priority
Remote transmission request (RTR)	1	Must be dominant (0)
Reserved bits (r0, r1)	2	Reserved bits (it must be set dominant (0), but accepted as either dominant or recessive)
Data length code (DLC)*	4	Number of bytes of data (0–8 bytes)
Data field	0–64 (0-8 bytes)	Data to be transmitted (length dictated by DLC field)
CRC	15	Cyclic redundancy check (CRC)
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

Automotive Bus Systems

CAN Bus – Communication Protocol

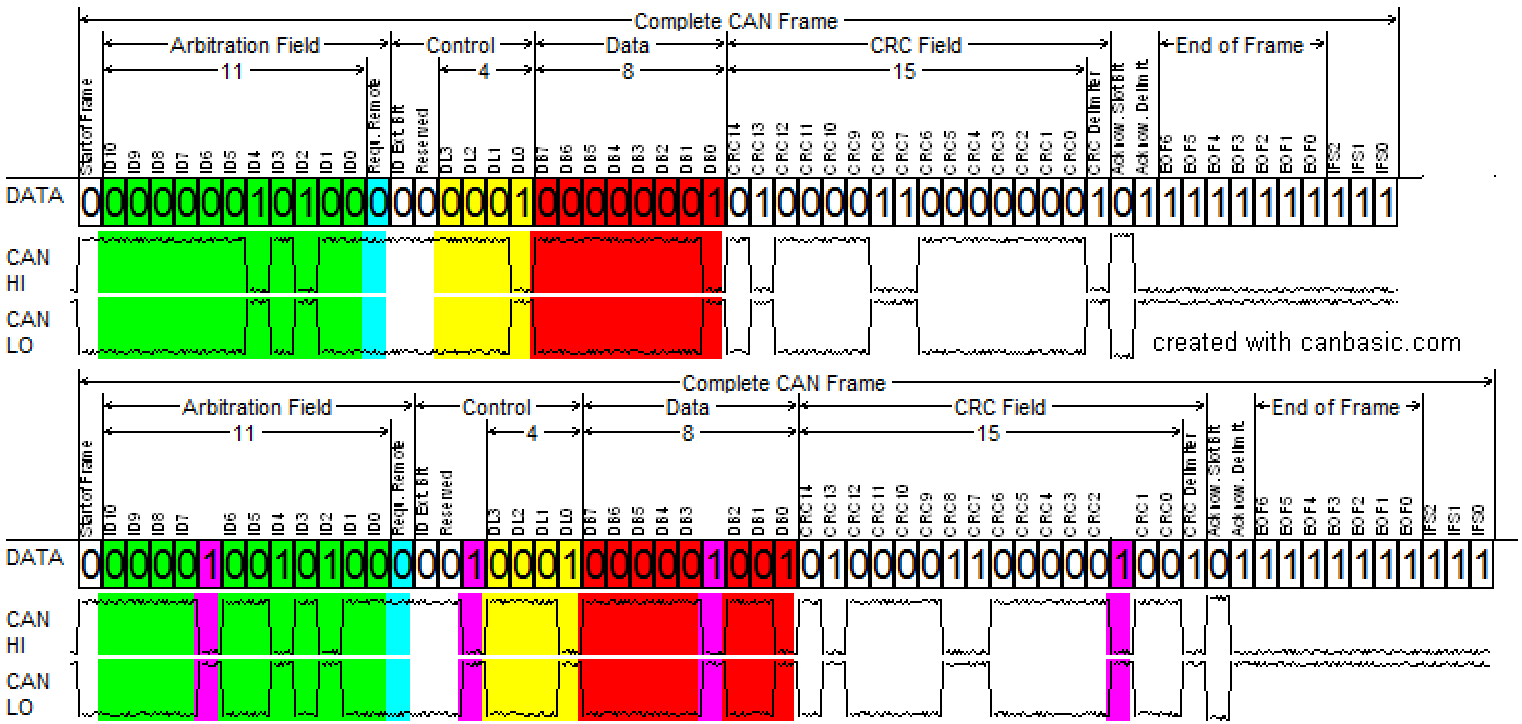
- Bit stuffing
  - Ensures minimum clock resynchronization frequency
  - After each sequence of five consequtive bits of one value a bit of different value is inserted
  - Bit stuffing affects data frames and RTR frames
    - From SOF delimiter to end of CRC field



Automotive Bus Systems

CAN Bus – Communication Protocol

- Bit stuffing (example)



Automotive Bus Systems

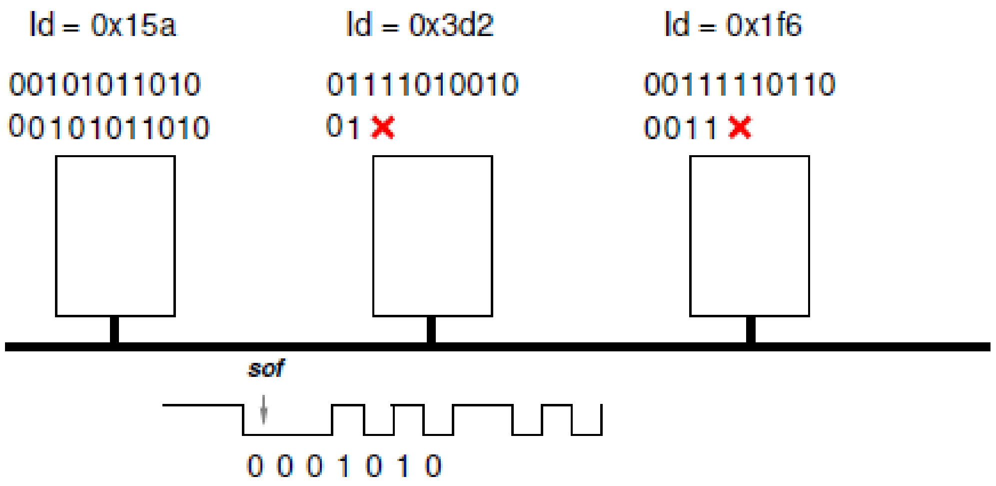
CAN Bus – Communication Protocol

- Bus arbitration sequence
  - Wait for end of current transmission (wait for 6 consecutive recessive Bits)
  - All nodes with pending transmission requests send identifier concurrently
    - Nodes send and concurrently detect bit on bus
    - Watch for mismatch between transmitted/detected signal level Means that a collision with a higher priority message has occurred
    - Back off from bus access, retry later
- Realization of non-preemptive priority scheme
  - Real time guarantees for message with highest priority
    - i.e., message with longest “0”-prefix

Automotive Bus Systems

CAN Bus – Communication Protocol

- Example



Automotive Bus Systems

CAN Bus – Communication Protocol

- RTR Frames
  - Regular data frames without data bit field
  - DLC set to expected number of bits
  - RTR bit set to ‚1‘

- What happens when RTR frame and regular frame with same ID is transmitted concurrently?

CAN Bus – Error and fault Containment

- There are 5 types of error
  - BIT ERROR
    - The sender monitors the bus. If the value found on the bus is different from the one that is sent, then a BIT ERROR is detected
  - STUFF ERROR
    - Detected if 6 consecutive bits of the same type are found
  - CRC ERROR
    - Detected by the receiver if the received CRC field does not match the computed value
  - FORM ERROR
    - Detected when a fixed format field contains unexpected values
  - ACKNOWLEDGEMENT ERROR
    - Detected by the transmitter if a dominant value is not found in the ACK slot

Automotive Bus Systems

CAN Bus – Error and fault Containment

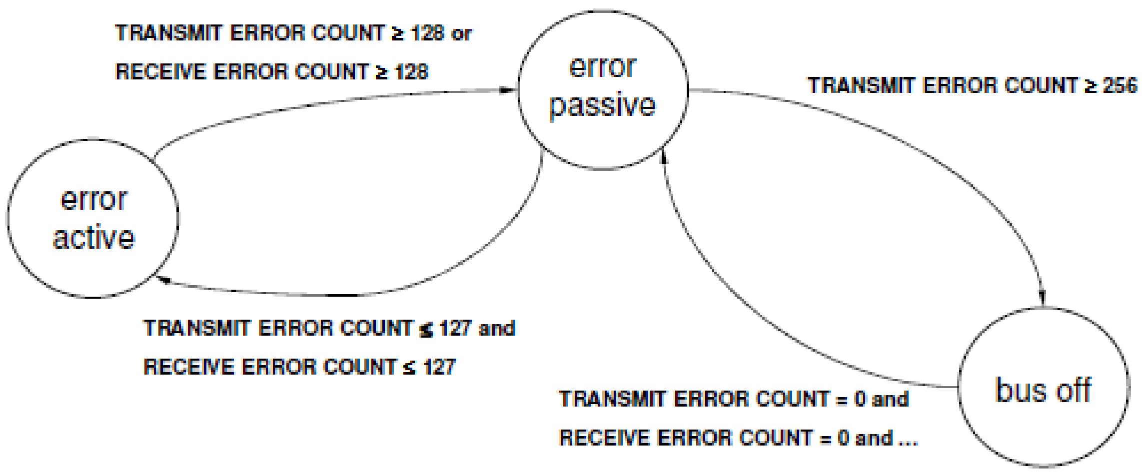
- A station detecting an error transmits an ERROR FRAME.
  - For BIT, STUFF, FORM, ACKNOWLEDGEMENT errors, it is sent in the immediately following bit.
  - For CRC it is sent after the ACK DELIMITER

- Error Flag
  - Consists of two different fields:
    - The first field is given by the superposition of ERROR FRAME
      - 6–12 dominant/recessive bits, transmitted by different stations.
    - The following second field is the ERROR DELIMITER
      - 8 recessive bits
  - There are two types of error flags:
    - Active Error Flag: 6 dominant bits – Transmitted by a node detecting an error on the network that is in error state "error active“
    - Passive Error Flag: 6 recessive bits – Transmitted by a node detecting an active error frame on the network that is in error state "error passive"

Automotive Bus Systems

CAN Bus – Error and fault Containment

- Each node can be in 3 states:
  - Error active: normal, initial state
  - Error passive: limited error signalling and transmission features
  - Bus off: cannot influence the bus anymore



Automotive Bus Systems

CAN Bus – Discussion

- Which relevant properties for modern bus systems are addressed by CAN?
  - Reliability, Predictability, Scalability, Robustness?

- How are these addressed?

Automotive Bus Systems

CAN Bus – Scheduling analysis

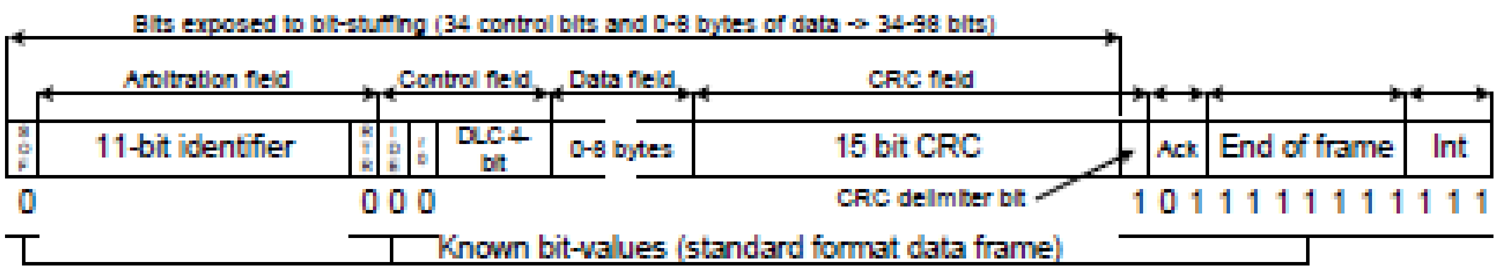
- Enables prediction of
  - Network transmission delays
  - End-to-end transmission delays

- Network transmission delay
  - Time that passes from queuing a CAN bus message at one node until this message is received by target node(s) of that message

- End-to-end transmission delay
  - Time that passes from transmitting a request messge until a response message is required
  - Requires additional consideration of task scheduling

CAN Bus – Scheduling analysis – Network transmission delay

- Frame size
  - Number of payload bytes s_i in CAN frame is $s_i \in [0,8]$
  - Number of total bits in CAN frame before bit stuffing
    - $c_i = v + 13 + 8s_i$
    - Variable v represents header bits
      - $v = 34$ for CAN standard format
      - $v = 54$ for CAN extended format
    - 13 additional bits after CRC are not exposed to bit stuffing



Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Frame size
  - $c_i = v + 13 + 8s_i$ is frame size before bit stuffing
  - Bit stuffing adds in the worst case b_i bits
    - $b_i = \left\lfloor \frac{v+8s_i-1}{4} \right\rfloor$
  - Total worst case frame size is therefore: $f_i = v + 13 + 8s_i + \left\lfloor \frac{v+8s_i-1}{4} \right\rfloor$
  - Why do we calculate the worst case frame size?

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Transmission time
  - Bit transmission time is t_{bit}
  - Frame transmission time C_i is $C_i = \left(v + 13 + 8s_i + \left\lfloor \frac{v+8s_i-1}{4} \right\rfloor\right) t_{bit}$

- Transmission time of CAN standard frame is
  - $C_i = (55 + 10s_i)t_{bit}$ $\quad$ $(v = 34)$

- Transmission time of CAN extended frame is
  - $C_i = (80 + 10s_i)t_{bit}$ $\quad$ $(v = 54)$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Scheduling analysis context assumptions (1)
  - For the analysis presented here, frames are assumed to be generated in a periodic manner, i.e., with a fixed amount of time between any instances of the same frame. Hence, each frame is associated with a frame period denoted T_i, i.e., the time between the generation of two instances of the same frame.
  - This temporal representation also models sporadic frame arrivals, i.e., frames that are not generated periodically but have a known minimum inter-arrival time between instances of the same frame.
  - The sporadic frames are simply represented by their worst-case arrival pattern, corresponding to periodic frames with periods (T_i) equal to the minimum inter-arrival time of the corresponding sporadic frame.

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Scheduling analysis context assumptions (2)
  - As frames are generated by software running on the nodes in the system, additional time needs to be taken into account to allow for queuing and interrupt latencies.
  - In fact, the time between the creation of the frame in the software at the node, and the time when the frame is available in the CAN communication adapter, is called the **queuing jitter**. This time is assumed to be bounded by J_i.
  - Each frame *i* has an associated frame-relative deadline, denoted D_i. A frame relative deadline is the time it must have completed before transmission, relative to the start of the frame period T_i.
  - The frame worst-case response time R_i is the longest time needed for any instance of frame *i* to finish frame transmission, relative to the start of the period T_i. The system is said to be schedulable if $R_i \leq D_i \;\; \forall i$.

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- A communication system is described as a set of streams φ
  - These are transmitted via CAN bus
  - Each stream represents a series of CAN bus frames of type i

- Each Stream $S_i \in \varphi$ is defined by the tuple (P_i, T_i, J_i, C_i)
  - P_i is the frame priority, defined by the frame identifier
  - T_i is the frame period
  - J_i is the frame queuing jitter
  - C_i is the worst case transmission time of frame type i

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Sufficient response time analysis
  - Pessimistic test
    - The frame is schedulable if the test holds,
    - If the test fails, the frame still might be schedulable
    - Does not work if multiple instances of frame type i are queued at the same time
  - Start analysis for frame type i at critical instant for frame type i
  - Critical instant: All frames of priority P_i and higher are simultanously queued

  - Worst case response time R_i for frame type i
    - $R_i = J_i + w_i + C_i$
      - J_i is the frame queuing jitter
      - C_i is the worst case transmission time of frame type i
      - w_i is the worst case queuing delay for frame type i

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Sufficient response time analysis
  - Worst case queuing delay w_i represents the worst case time a frame has to wait in the transmit queue before it will be transmitted
  - Calculation of worst case queuing delay w_i
    - Recursive calculation: Start with $w_i^0 = C_i$
    - Calculate next iteration $w_i^{n+1} = B^{MAX} + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i^n + J_j + t_{bit}}{T_j} \right\rceil C_j$
    - Terminate calculation when
      - $w_i^{n+1} = w_i^n$ and/or $J_i + w_i^{n+1} + C_i > D_i$
    - B^{MAX} is transmission time of longest possible CAN frame
    - The set $hp(i)$ contains the frame types j with higher priority than type i

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example

Frame i	P_i	T_i	D_i	C_i
1	1	187.5	187.5	75
2	2	262.5	262.5	75
3	3	262.5	262.5	75

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution

$B_{MAX} = 75$

$w_1^0 = 75 \qquad w_1^1 = 75$

$R_1 = 0 + 75 + 75 = 150$

$w_2^0 = 75 \qquad w_2^1 = 75 + 75 = 150 \qquad w_2^2 = 75 + 75 = 150$

$R_2 = 0 + 150 + 75 = 225$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Exact response time analysis
  - Exact worst case response time $\overline{R_i}$ for frame type i sent on stream S_i is found within level-i busy period
  - Individual frame response time has to be calculated for every instance of frame type i within ist busy period
  - Worst case response time $\overline{R_i}$ is experienced by one or multiple instances of frame type i. Deriving of the worst case response time for a frame type requires the calculation of the worst case response time $\overline{R_i}(\mathrm{q})$, 1≤q ≤ n of all of its frame instances 1..n.
  - Analysis starts again at critical instant

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Calculation of level-i busy period for frame type i
  - Recursive calculation
  - Starts with $t_i^0 = C_i$, ends when $t_i^{n+1} = t_i^n$
  - $t_i^{n+1} = B_i + \sum_{\forall k \in hep(i)} \left\lceil \frac{t_i^n + J_k}{T_k} \right\rceil C_k$
  - B_i is maximum blocking time due to lower priority CAN frames
  - Set $hep(i)$ contains frame types with higher or equal priority to i

- Number of frame instances Q_i of frame i that become ready before the end of the busy period
  - $Q_i = \left\lceil \frac{t_i + J_i}{T_i} \right\rceil$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Calculation of worst case response time for frame type i
  - Worst case response time must be derived for each frame instance $(0 \ldots Q_i - 1)$
  - Assuming q being the index of the currently processed frame instance of frame type i, worst case response time $\bar{R_i}(\mathrm{q})$ for said frame instance is:

  - $\bar{R_i}(\mathrm{q}) = J_i + \overline{w_i}(q) - qT_i + C_i$

- Variable $\overline{w_i}(q)$ represents effective queuing time of instance q for frame type i
  - Calculated by recurrence relation (see next slide)
    - $\overline{w_i^{n+1}}(\mathrm{q}) = B_i + qC_i + \sum_{j \in hp(i)} \left\lceil \frac{\overline{w_i^n}(q) + J_j + t_{bit}}{T_j} \right\rceil C_j$
  - Initial value $\overline{w_i^0}(q) = 0$
  - Terminate if $\overline{w_i^{n+1}}(q) = \overline{w_i^n}(q)$ and/or $J_i + \overline{w_i^{n+1}}(q) - qT_i + C_i > D_i$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Calculation of worst case response time for frame type i
  - The worst case response time $\overline{R_i}$ for frame i is found as following:
  - $\overline{R_i} = \max\limits_{q=0\ldots Q_i-1}\left(\overline{R_i(q)}\right)$

- In contrast to the sufficient response time test, this test also works for frames with deadlines greater than their period

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example

Frame *i*	P_i	T_i	D_i	C_i
1	1	187.5	187.5	75
2	2	262.5	262.5	75
3	3	262.5	262.5	75

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 1

Calculate length of busy period for Frame 1:

You need to consider Frame type 1

$$t_1^0 = C_t = 75 \qquad t_1^1 = 75 + \left\lceil \frac{75+0}{187{,}5} \right\rceil \cdot 75 = 150 \qquad t_1^2 = 75 + \left\lceil \frac{150+0}{187{,}5} \right\rceil \cdot 75 = 150$$

Calculate number of concurrent frame instances for Frame 1:

$Q_1 = \left\lceil \frac{150}{187{,}5} \right\rceil = 1$ → One concurrent frame instance

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 1

Calculate effective queuing times for each instance of Frame 1:

Frame instance 1:

$\overline{w}_1^0(0) = 0$ $\overline{w}_1^1(0) = 75 + 0 = 75$ $\overline{w}_1^2(0) = 75 + 0 = 75$

Calculate worst case response times for each frame instance:

Frame instance 1:

$\overline{R_1}(0) = 0 + 75 - 0 \cdot T_1 + 75 = 150$

Calculate worst case response time for all instances of Frame type 1:

$R_1 = \max(150) = 150$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 2

Calculate length of busy period for Frame 2:

You need to consider Frame type 1 and 2

$$t_2^0 = C_t = 75$$

$$t_2^1 = 75 + \left\lceil \frac{75+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{75+0}{262{,}5} \right\rceil \cdot 75 = 225$$

$$t_2^2 = 75 + \left\lceil \frac{225+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{225+0}{262{,}5} \right\rceil \cdot 75 = 300$$

$$t_2^3 = 75 + \left\lceil \frac{300+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{300+0}{262{,}5} \right\rceil \cdot 75 = 375$$

$$t_2^4 = 75 + \left\lceil \frac{375+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{375+0}{262{,}5} \right\rceil \cdot 75 = 375$$

Calculate number of concurrent frame instances for Frame 2:

$$Q_2 = \left\lceil \frac{375}{262{,}5} \right\rceil = 2$$

→ Two concurrent instances need to be considered

CAN Bus – Scheduling analysis – Network transmission delay

Calculate effective queuing times and Response times for each instance of Frame 2:

Worst case queuing time for frame instance 1:

$$\overline{w}_2^0(0) = 0 \qquad \overline{w}_2^1(0) = 75 + 0 + \left\lceil \frac{0+0+1}{187,5} \right\rceil \cdot 75 = 150$$

$$\overline{w}_2^2(0) = 75 + 0 + \left\lceil \frac{150+0+1}{187,5} \right\rceil \cdot 75 = 150$$

Worst case response time for frame instance 1:

$$\overline{R_2}(0) = 0 + 75 - 0 \cdot T_2 + 150 = 225$$

Worst case queuing time for frame instance 2:

$$\overline{w}_2^0(1) = 0 \qquad \overline{w}_2^1(1) = 75 + 75 + \left\lceil \frac{0+0+1}{187,5} \right\rceil \cdot 75 = 225$$

$$\overline{w}_2^2(1) = 75 + 75 + \left\lceil \frac{225+0+1}{187,5} \right\rceil \cdot 75 = 300$$

$$\overline{w}_2^3(1) = 75 + 75 + \left\lceil \frac{300+0+1}{187,5} \right\rceil \cdot 75 = 300$$

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 2

Worst case response time for frame instance 2:

$$\overline{R_2}(1) = 0 + 75 - 1 \cdot 262{,}5 + 300 = 112{,}5$$

Calculate worst case response time for all instances of Frame type 2:

$$R_2 = \max(225; 112{,}5) = 225$$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 3

Calculate length of busy period for Frame 3:

You need to consider Frame type 1, 2 and 3

$$t_2^0 = C_t = 75$$

$$t_2^1 = 0 + \left\lceil \frac{75+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{75+0}{262{,}5} \right\rceil \cdot 75 + \left\lceil \frac{75+0}{262{,}5} \right\rceil \cdot 75 = 225$$

$$t_2^2 = 0 + \left\lceil \frac{225+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{225+0}{262{,}5} \right\rceil \cdot 75 + \left\lceil \frac{225+0}{262{,}5} \right\rceil \cdot 75 = 300$$

$$t_2^3 = 0 + \left\lceil \frac{300+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{300+0}{262{,}5} \right\rceil \cdot 75 + \left\lceil \frac{300+0}{262{,}5} \right\rceil \cdot 75 = 450$$

$$t_2^4 = 0 + \left\lceil \frac{450+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{450+0}{262{,}5} \right\rceil \cdot 75 + \left\lceil \frac{450+0}{262{,}5} \right\rceil \cdot 75 = 525$$

$$t_2^5 = 0 + \left\lceil \frac{525+0}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{525+0}{262{,}5} \right\rceil \cdot 75 + \left\lceil \frac{525+0}{262{,}5} \right\rceil \cdot 75 = 525$$

Calculate number of concurrent frame instances for Frame 3:

$$Q_2 = \left\lceil \frac{525}{262{,}5} \right\rceil = 2$$

→ Two concurrent instances need to be considered

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 3

Calculate effective queuing times and Response times for each instance of Frame 2:

Worst case queuing time for frame instance 1:

$$\overline{w}_2^0(0) = 0$$

$$\overline{w}_2^1(0) = 0 + 0 + \left\lceil \frac{0+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{0+0+1}{262{,}5} \right\rceil \cdot 75 = 150$$

$$\overline{w}_2^2(0) = 0 + 0 + \left\lceil \frac{150+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{150+0+1}{262{,}5} \right\rceil \cdot 75 = 150$$

Worst case response time for frame instance 1:

$$\overline{R_3}(0) = 0 + 75 - 0 \cdot T_2 + 150 = 225$$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Example solution – Frame type 3

Worst case queuing time for frame instance 2:

$$\overline{w}_2^{0}(0) = 0$$

$$\overline{w}_2^{1}(1) = 0 + 75 + \left\lceil \frac{0+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{0+0+1}{262{,}5} \right\rceil \cdot 75 = 225$$

$$\overline{w}_2^{2}(1) = 0 + 75 + \left\lceil \frac{225+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{225+0+1}{262{,}5} \right\rceil \cdot 75 = 300$$

$$\overline{w}_2^{3}(1) = 0 + 75 + \left\lceil \frac{300+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{300+0+1}{262{,}5} \right\rceil \cdot 75 = 375$$

$$\overline{w}_2^{4}(1) = 0 + 75 + \left\lceil \frac{375+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{375+0+1}{262{,}5} \right\rceil \cdot 75 = 450$$

$$\overline{w}_2^{5}(1) = 0 + 75 + \left\lceil \frac{450+0+1}{187{,}5} \right\rceil \cdot 75 + \left\lceil \frac{450+0+1}{262{,}5} \right\rceil \cdot 75 = 450$$

Worst case response time for frame instance 2:

$$\overline{R_3}(1) = 0 + 75 - 1 \cdot T_2 + 450 = 0 + 75 - 262{,}5 + 450 = 262{,}5$$

$$R_2 = \max(225; 262{,}5) = 262{,}5$$

Automotive Bus Systems

CAN Bus – Scheduling analysis – Network transmission delay

- Enables calculation of deterministic worst case execution time bounds
  - Requires knowledge about all message types and worst case transmission schedules on all network nodes
  - Scheduling analysis must be repeated whenever message types and message transmission schedules are modified

- CAN bus scheduling analysis only considers network transmission delays