REAL-TIME NETWORKS
Controller Area Network

Prof. J.-D. Decotignie
CSEM Centre Suisse d'Electronique et de Microtechnique SA
Jaquet-Droz 1, 2007 Neuchâtel
jean-dominique.decotignie@csem.ch

Outline

- CAN (Controller Area Network) history and use
- Architecture and variants
- Physical layer
- Data link layer
- Temporal behavior
- Response time analysis
- Improving the real-time behavior

©2016, J.-D. Decotignie

History

- 1983: Initial development by Bosch
- 1986: Official introduction of CAN protocol
- 1991: CAN specification 2.0 published
- 1991: first application layer (CAN Kingdom)
- 1992: 2nd AL, CAN Application Layer (CAL) by CiA
- 1992: 1st cars from Mercedes-Benz using CAN
- 1993: becomes an international standard (ISO 11898)
- 1994: modifications for industrial use :DeviceNet by Allen-Bradley and SDS by Honeywell
- 1995: CAN open protocol from CiA (CAN in Automation)
- 2000: Time-triggered comm. protocol for CAN (TTCAN)
- 2012: CAN-FD (payload up to 64B @ bit rate 8 x arbitration)

©2016, J.-D. Decotignie

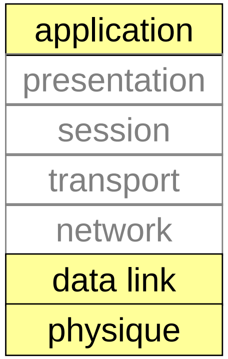
Market

- Initially development for the car industry
  - In competition with others
- Attracted a lot of interest outside initial market
  - Due to availability of inexpensive silicon
- Today:
  - Actually used in cars but not for safety critical functions
  - Some use in the industry
  - Widely available in micro-controllers

©2016, J.-D. Decotignie

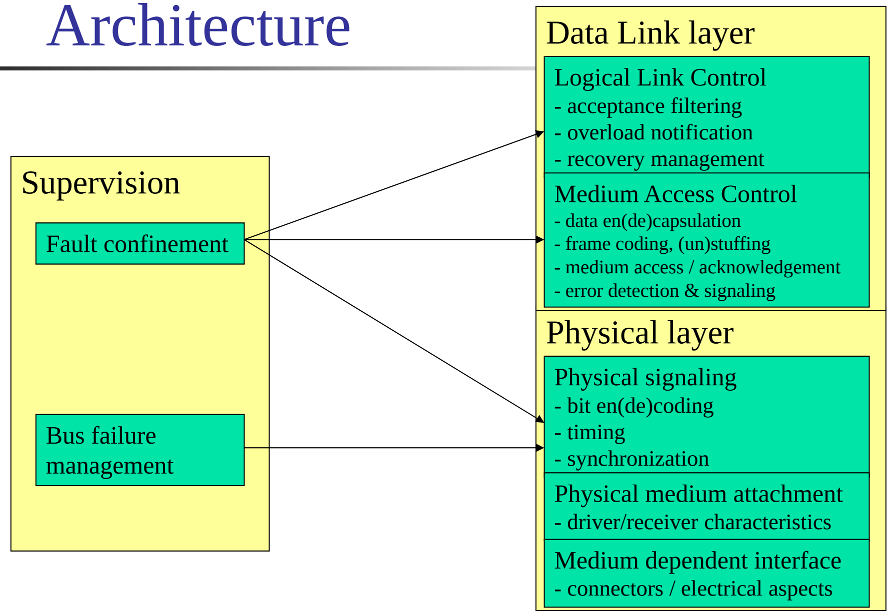
Architecture

- 3 layer collapsed model
- Only first 2 layers are standard
- A few proposals for AL
  - CAL
  - CAN Kingdom
  - DeviceNet
  - CAN Open
  - SAE J1939
  - ….



©2016, J.-D. Decotignie

Architecture



©2016, J.-D. Decotignie

Variants

- SDS
- DeviceNet
- TT-CAN
- FTT-CAN
- CAN-FD

©2016, J.-D. Decotignie

Physical layer

- Topology
  - Terminated bus
- Number of stations
  - In principle limited to 30 (depends on drivers)
- Medium
  - Twisted pair, single wire (FO possible but not standard)
- Range
  - Signaling speed and propagation speed dependent: 40m at 1Mbit/s
  - Drop length limited to 30 cm
- Signaling and bit encoding
  - 10 kbit/s to 1 Mbit/s, NRZ
  - Up to 8 Mbit/s in payload with CAN-FD

©2016, J.-D. Decotignie

Physical layer (2)

- Synchronization
  - Uses signal edges (implies bit stuffing with NRZ)
    - After 5 consecutive ones, a zero is inserted
    - After 5 consecutive zeros, a one is inserted
    - This rules includes a possible stuffing bit inserted before
- Signals
  - Recessive: logical “1”
  - Dominant: logical “0”
    - When 2 stations compete on a bit by bit basis, the station that emits a dominant bit imposes this level on the bus

©2016, J.-D. Decotignie

Medium Access Control - frame

- RTR: data or request frame
- IDE: normal or extended format
- DLC: data field length
- EOF: End of frame (7 recessive bits)
- Ack: global ack by all connected nodes
- IFS: inter frame silence (3 recessive bits)
- R0: reserved (dominant)

SOF	Identifier	RTR	IDE	R0	DLC	Data	CRC	ACK	EOF	IFS
1	11	1	1	1	4	0..8 x 8	16	2	7	3

©2016, J.-D. Decotignie

Extended addressing

- RTR: data or request frame
- IDE: extended format (recessive)
- DLC: data field length
- EOF: End of frame (7 recessive bits)
- Ack: global ack by all connected nodes
- IFS: inter frame silence (3 recessive bits)
- R0,R1: reserved (dominant)
- SRR: (recessive)

SOF	Identifier	SRR	IDE	Identifier extension	RTR	R1	R0	DLC	Data	CRC	ACK	EOF	IFS
1	11	1	1	18	1	1	1	4	0..8 x 8	16	2	7	3

©2016, J.-D. Decotignie

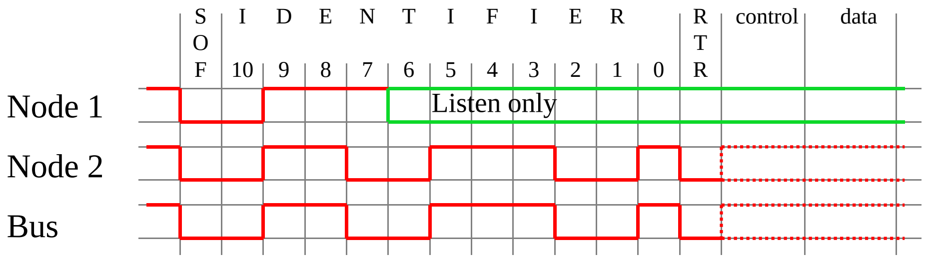
Addressing

- Single 11 or 29 bit identifier par frame
  - If used to identify a node
    - Source (data) or destination (request) of the message
  - Normally used to identify the payload
    - Also called “Broadcast source addressing”
  - A lower value gives higher priority in contention

©2016, J.-D. Decotignie

Medium Access Control

- CSMA with collision resolution
  - Each node observes bus while transmitting
  - If level different from what it has put, withdraws
  - Dominant bit overwrites recessive one



©2016, J.-D. Decotignie

Error detection

- Several means
  - Bit error
    - When what is one the bus is different from what was emitted
      - Except when a recessive bit was emitted during arbitration or ack slot
  - Cyclic Redundancy Check (CRC)
  - Frame check (the frame structure is checked)
  - ACK errors (absence of a dominant bit during the ack slot)
  - Monitoring (each node which transmits also observes the bus level and thus detects differences between the bit sent and the bit received)
  - Bit stuffing (checking adherence to the stuffing rule.)
- A frame is valid for
  - A transmitter if there is no error until the end of EOF
  - A receiver if there is no error until the next to last bit of EOF

©2016, J.-D. Decotignie

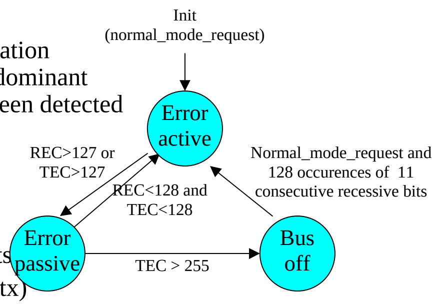
Behaviour in case of error

- In case of stuff, bit, form or acknowledge errors
  - An error flag is started at the next bit
- In case of CRC error
  - An error frame is send after the ack delimiter
- Fault confinement
  - Each time an reception error occurs, REC is incremented
  - Each time a frame is received correctly, REC is decremented
  - Same for the emission errors with TEC
  - The values of TEC and REC may trigger mode changes

©2016, J.-D. Decotignie

Connection modes

- To enforce fault confinement, nodes may be in one of 3 modes
  - Error active
    - Normally takes part to the communication and may send an active error flag (6 dominant consecutive bits) when an error has been detected
  - Error passive
    - Takes part in communication but must not send an active error flag. Instead, it shall send a passive error flag (6 recessive consecutive bits)
    - Some restrictions (silence between 2 tx)
  - Bus off
    - Cannot send or receive any frame.
    - A node is in this state when it is switched of the bus due to a request from a fault confinement entity. May exit from this state only by a user command.



©2016, J.-D. Decotignie

Error frame

- 2 fields: error flag and error delimiter
- Error flag
  - Active: 6 dominant bits
  - Passive: 6 recessive bits
  - As all nodes monitor the bus and the flag violates stuffing rules, they will send error flags too
    - The error flag will last from 6 to 12 bits
- Error delimiter (8 recessive bits)
  - After sending an error flag, a node shall send recessive bits
    - As soon as it senses a recessive bit, it sends 7 recessive bits

©2016, J.-D. Decotignie

Error recovery

- Automatic retransmission
  - of all frames that have lost arbitration
  - of all frames have been disturbed by errors during transmission

©2016, J.-D. Decotignie

Medium Access Control

- All messages are sent in broadcast
- Nodes filter according to their interest
- All messages are acknowledged including by nodes that are not interested by the message
  - Acknowledge just means “message well received by all receivers”
    - It does not mean “intended receiver received it”
- Node that does not receive message correctly sends an error bit sequence
- Node that is too busy may send an overload bit seq.
  - MA_OVLD.request/indication/confirm
  - Same principle as an error frame (overload frame = 6 dominant + 8 recessive bit)

©2016, J.-D. Decotignie

Logical Link Control

- 2 types of services (connectionless)
  - Send Data with no ack
    - L_DATA.request, L_DATA.indication, L_DATA.confirm
    - Uses a data frame
  - Request Data
    - L_REMOTE.request, L_REMOTE.indication, L_REMOTE.confirm
    - Uses a remote frame (same as a data frame but data field is empty)
- Flow control using the overload bit sequence

©2016, J.-D. Decotignie

Temporal analysis

- CAN is essentially unfair
  - Lower identifier frames get priority
  - There exists schemes to overcome this
    - [Cena 2001]
    - DeviceNet to some extent
- However
  - Given a set of traffic needs
  - Given that all nodes comply with the expressed needs
    - It is possible to check whether a network will comply with the requirements

©2016, J.-D. Decotignie

Traffic requirements

- Real-time periodic message streams
  - $M_{p,m}=\{T_m, D_m, C_m, J_m\}$
    - C_m duration of emission of message, T_m period of transfer, D_m relative deadline from beginning of period (absolute deadlines are $d_{n,m} = n\, T_m + D_m$)
    - J_m is the arrival jitter (variability in queuing instants periodicity)
- Real-time sporadic message streams
  - $M_{s,m}=\{T_m, D_m, C_m\}$
    - T_m min. interarrival time, D_m relative deadline from arrival time (absolute deadlines are $d_{n,m} = arr_{n,m} + D_m$)

©2016, J.-D. Decotignie

Response time analysis

- The worst case response time of a message R_m is the longest time between
  - Queuing of the message
  - Arrival time of the message at destination
- A set of traffic requirements can be handled by a CAN networks iif $R_m \leq D_m$ for all message streams
- $R_m = C_m + J_m + I_m$
  - Where I_m is the interference time

©2016, J.-D. Decotignie

Response time analysis

- Response time [Tindell 94&95]

$$R_m = C_m + J_m + I_m \text{ with } I_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{I_m + J_m + \tau_{bit}}{T_j} \right\rceil C_j$$

  - B_m is the transmission of the longest lower priority message + S the duration of the interframe silence
  - Assumes no new message queued before previous sent
- Different from Joseph and Pandya

$$R_m = C_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{R_m}{T_j} \right\rceil C_j$$

©2016, J.-D. Decotignie

Message emission time

- Influence of bit stuffing
  - Normal addressing

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

  - Extended addressing

$$C_m = \left(65 + 8b + \left\lfloor \frac{52 + 8b - 1}{4} \right\rfloor \right) \tau_{bit}$$

    - In the original paper 5 was used instead of 4 in the division. This is a mistake due to a lack of understanding of the stuffing mechanism that also includes stuff bits

©2016, J.-D. Decotignie

Example - SAE benchmark

		Bits	Period [ms]	Typ	D[ms]	source
1,2	Traction battery voltage / current	8/8	100/100	P/P		Battery
3,5	Traction battery temp. average / max	8/8	1000/1000	P/P		Battery
4,6	Auxiliary battery voltage / current	8/8	100/100	P/P		Battery
7,17	Accelerator position / switch	8/2	5/50	P/S	-/20	driver
8,9	Brake pressure master cylinder / line	8/8	5/5	P/P		brakes
10,11	Transaxle lubrication/Trans. clutch line pressure	8/8	100/5	P/P		Transmission
12,18	Vehicle speed / brake switch	8/1	100/20	P/S	-/20	brakes
13,36	Traction battery ground fault / idem test	1/2	1000/1000	P/P		Battery/ V-C
14,28	HI&LO contactor open-close / interlock	4/1	50/50	S/S	5/20	Battery
15,16	Key switch run / start	1/1	50	S/S	20/20	driver
19,20	Emergency brake / shift lever	1/3	50/50	S/S	20/20	driver
22,26	Speed control / brake mode / SOC reset	3/1/1	50/50/50	S/s/s	All 20	driver
21	Motor/trans over temperature	2	1000	P		transmission
23-25	12V Power ack vehicle ctrl / inverter / I-M cont	1/1/1	50/50/50	S/S/s	All 20	battery

©2016, J.-D. Decotignie

Example - SAE benchmark

		Bits	Period [ms]	Typ	D[ms]	source
29,30	High / low contactor control	8/8	10/10	P/P		V-C
31,35	Reverse & 2nd gear clutches / 12V power relay	2/1	50/50	S/S	20/20	V-C
32,33	Clutch pressure ctrl / DC-DC converter	8/1	5/1000	P/P		V-C
34,37	DC-DC converter current ctrl / brake solenoid	8/1	50/50	S/S	20/20	V-C
38,39	Backup alarm / warning lights	1/7	50/50	S/S	20/20	V-C
40	Key switch	1	50	S	20	V-C
42,43	Torque command / torque measured	8/8	5/5	P/P		V-C / I-M C
41,45	Main contactor close / Fwd-rev ack	1/1	50/50	S/S	20/20	I-M C
44,46	Fwd-Rev / idle	1/1	50/50	S/S	20/20	V-C
47,49	Inhibit / Processed motor speed	1/8	50/5	S/P	20/-	I-M C
48,53	Shift in progress / main contactor ack	1/8	50/50	S/S	20/20	V-C
50	Inverter temperature status	2	50	S	20	I-M C
51	Shutdown	8	50	S	20	I-M C
52	Status-malfunction	1	50	S	20	I-M C

©2016, J.-D. Decotignie

Timing analysis [Tindell 95]

Signal number	Byte	Period [ms]	Type	D[ms]	R [ms]	source
14	1	50	S	5	1.544	Battery
8,9	2	5	P	5	2.128	Brakes
7	1	5	P	5	2.632	Driver
43,49	2	5	P	5	3.216	I-M C
11	1	5	P	5	3.72	Trans
32,42	2	5	P	5	4.304	V-C
31,34,35,37-40,44,46,48,53	6	10	P	10	5.192	V-C
23,24,25,28	1	10	P	10	8.456	Battery
15-17,19,20,22,26,27	2	10	P	10	9.04	Driver
41,45,47,50-52	2	10	P	10	9.624	I-M C
18	1	100	S	20	10.128	Brakes
1,2,4,6	4	100	P	100	18.944	Battery
12/10/21	1/1/1	100/100/1000	P/P/P	=T	19.44/19.55/29.19	Br/trans
3,5,13/33,36	3/1	1000/1000	P/P	=T	20.608/29.696	Batt/V-C

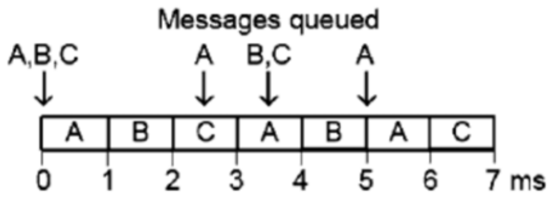
@125 kbit/s

©2016, J.-D. Decotignie

Flaw in Tindell's analysis

- Consider

Message	Priority	Period	Deadline	Tx time
A	3	2.5 ms	2.5 ms	1 ms
B	2	3.5 ms	3.25 ms	1 ms
C	1	3.5 ms	3.25 ms	1 ms



©2016, J.-D. Decotignie

New Analysis

- Use the definition of busy period. Level-m busy period is defined as:
  - It starts at some time t^s when a message of priority m or higher is queued ready for transmission and there are no messages of priority m or higher waiting to be transmitted that were queued strictly before time t^s .
  - It is a contiguous interval of time during which any message of priority lower than m is unable to start transmission and win arbitration.
  - It ends at the earliest time t^e when the bus becomes idle, ready for the next round of transmission and arbitration, yet there are no messages of priority m or higher waiting to be transmitted that were queued strictly before time t^e.
- All messages of priority m or higher, queued strictly before the end of the busy period, are transmitted during the busy period.
- These messages cannot therefore cause any interference on a subsequent instance of message m queued at or after the end of the busy period.
- Maximal busy period start at the critical instant when all message m is queued simultaneously with all higher priority msg and each of these is subsequently queued at highest speed

©2016, J.-D. Decotignie

Where is the flaw ?

- Tindell's analysis

$$R_m = C_m + J_m + I_m \text{ with } I_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{I_m + J_m + \tau_{bit}}{T_j} \right\rceil C_j$$

- Implicitely assumes that level-m busy period will end at or before T_m.
  - Would be true for preemptive scheduling as on completion of message m, no higher priority message would be pending
  - However in CAN, on completion of a message transmission, a higher priority message may be pending
    - Level-m busy period may extend beyond T_m.

©2016, J.-D. Decotignie

New Analysis

- Busy period length $t_m = B_m + \sum_{\forall j \in hep(m)} \left\lceil \frac{t_m + J_m}{T_j} \right\rceil C_j$
- Number of instances of message m that becomes ready for transmission before the end of the busy period

$$Q_m = \left\lceil \frac{t_m + J_m}{T_m} \right\rceil$$

- The longest time from the start of busy period to instance q (q=0 is first one) starting transmission is

$$I_m(q) = B_m + qC_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{I_m + J_m + \tau_{bit}}{T_j} \right\rceil C_j$$

©2016, J.-D. Decotignie

Conclusion on response time analysis

- Takes profit of fixed priority nature of transmission
- Provided all nodes play the game (do not exceed traffic announced)
  - Allows to assess if traffic can be handled in absence of errors

©2016, J.-D. Decotignie

Improving the real-time behaviour

- Provide fairness guarantees
  - MUST and CAN [Cena 01]
- Better predictability of periodic traffic
  - TTCAN [Leen 02]
- Same with handling sporadic traffic
  - FTT-CAN [Almeida 02]
- Comparing CAN and TTCAN in presence of errors
  - [Broster 04]

©2016, J.-D. Decotignie

Updated response time analysis

- Response time [Broster 04]

$$R_m = t_m + J_m \quad \text{with} \quad t_m = B_m + C_m + I_m(t_m) + E_m(t_m)$$

  - $E_m(t)$ is the worst case overhead due to network faults and extra frames occurring before t
  - B_m is the transmission of the longest lower priority message + S the duration of the interframe silence
  - $I_m(t) = \sum_{\forall j \in hp(m)} \left\lceil \frac{t - C_m + J_j + \tau_{bit}}{T_j} \right\rceil (S + C_j)$
  - Assumes no new message queued before previous sent

©2016, J.-D. Decotignie

Taking faults into account

- Here we only consider network faults
  - All nodes are well behaved (according to protocol)
  - Caused by electromagnetic interferences or physical faults
    - It is possible for 2 nodes on the bus to simultaneously read different values from the bus
- Scenarios
  - Error during a data frame $\Rightarrow$ error frame (duration E is transmitted): additional delay = C+E+S
  - Error during error frame: additional delay = E+S
  - Error during bus idle (false start): add. delay = E+S
  - Burst of interferences of duration Z: add. delay = Z+C+E+S

©2016, J.-D. Decotignie

Sporadic fault model

- Faults are always separated by a minimum inter-arrival time T_F.
- n_{burst} is the maximum number of faults that can occur in a succession during a burst

$$E_m(t) = \left(n_{burst} + \left\lceil \frac{t}{T_F} \right\rceil \right) \left(E + \max_{j \in hep(m)} C_j \right)$$

  - hep(m) is the set of messages with priority higher or equal to m

©2016, J.-D. Decotignie

Probabilistic fault model

- Probability that k faults occur in interval t is given by

$$p(F = k, t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

- Each fault causes the maximum length error frame and occurs on the last bit of the longest frame

$$M_m = E + \max_{\forall j \in hep(m)} C_j$$

- The error overhead function is a random distribution
  $E_m = kM_m$ with probability $p(F = k, t)$
- So we get a set of pairs $R_m = <t_m, p(t_m)>$

©2016, J.-D. Decotignie

Analysis

- $R_m = <t_m, p(t_m)>$ gives us the response time in case of k errors (denoted as $R_{m|k}$) and the probability that k faults occurred between times 0 and t_m
- However, this cannot be taken as is:
  - For instance $R_{m|1}$ is the response time when one fault occurred before $R_{m|1}$.
  - However this fault must have occurred before $R_{m|0}$
    - Otherwise we would have had a response time of $R_{m|0}$

$$P(R_{m|1}) = p(1, [0, R_{m|1}]) - P(R_{m|0})\, p(1,]R_{m|0}, R_{m|1}])$$

©2016, J.-D. Decotignie

Scenari

Response Time	Possible Scenarios (Shorthand)	Number of Scenarios
$R_{i\|0}$	[0]	1
$R_{i\|1}$	[10]	1
$R_{i\|2}$	[200], [110]	2
$R_{i\|3}$	[3000], [2100], [2010], [1200], [1110]	5
$R_{i\|4}$	[40000], [31000], [30100], [30010], [21100], [21010], ...	14

Source: I. Broster, 2004

©2016, J.-D. Decotignie

Analysis (2)

- With 2 faults
  - We take the probability to have 2 faults before $R_{m,2}$
  - minus the case in which there is 0 fault before $R_{m,0}$
  - Minus the case in which there is 1 fault before $R_{m,1}$

$$\begin{aligned}P(R_{m|2}) = p(2,[0,R_{m|2}]) \\ - P(R_{m|1})\,p\big(1,]R_{m|1},R_{m|2}]\big) \\ - P(R_{m|0})\,p\big(2,]R_{m|0},R_{m|2}]\big)\end{aligned}$$

©2016, J.-D. Decotignie

Analysis (3)

- For k faults, this generalizes to:

$$P(R_{m|k}) = p(k,[0,R_{m|k}]) - \sum_{j=0}^{k-1} P(R_{m|j})\,p\big(k-j,]R_{m|j},R_{m|k}]\big)$$

- With

$$P(R_{m|0}) = p(1,[0,R_{m|0}])$$

- The probability of a deadline failure is then given by:

$$p_m(R_m > D_m) = 1 - \sum_{\forall j | R_{m|j} < D_m} p(R_{m|j})$$

©2016, J.-D. Decotignie

Case study

- SAE benchmark difficult to use because very tight
- Mobile robot message set [Borster 04]
  - bus at 256 kbit/s, 41% use, λ=30 faults/s

Priority	Length [ms]	Period [ms]	Deadline [ms]	WCRT [ms]	Probability of failure	signal
6	0.288	2	2	0.828	$1.5\ 10^{-5}$	Motor control
5	0.328	4	4	1.168	$1.6\ 10^{-9}$	Wheel 1
4	0.328	4	4	1.508	$8.7\ 10^{-8}$	Wheel 2
3	0.528	8	8	2.048	$2.7\ 10^{-9}$	Radio In
2	0.248	12	12	2.608	$2.1\ 10^{-12}$	Proximity
1	0.528	240	240	2.32	$< 10^{-20}$	Logging

©2016, J.-D. Decotignie

Summary

- Efficient for sporadic traffic
  - No control / non destructive distributed access
- Low response time in case of low traffic
- Periodicity difficult to achieve without jitter and some additional control
- A lot of care for error detection and handling
- Provides spatial consistency (global ack)
- Multicast easily implemented (broadcast source addr)
- Bounds can be derived for all kind of traffic
  - Under some assumptions
- MAC essentially unfair but there exists improvements

©2016, J.-D. Decotignie

References

- Standard
  - ISO 11898, « Road Vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communications », 1993.
- Sources of information www.can-cia.org
- Response time analysis
  - K. Tindell, A. Burns and A. J. Wellings, Calculating controller area network (can) message response times, Control Engineering Practice, vol. 3(8), pp. 1163-69, 1995.
  - K. Tindell et al. « Analyzing real-time communications: Controller Area Network (CAN) » , 1994
  - R. Davis et al., « Controller Area Network (CAN) schedulability analysis: Refuter, revisited and revised », Real-Time Systems, vol. 35, pp. 239-272, 2007.

©2016, J.-D. Decotignie

References (2)

- Response time analysis in case of errors
  - N. Navet, Y. -Q. Song and F. Simonot, Worst-case deadline failure probability in real-time applications distributed over controller area network , Journal of Systems Architecture, vol. 46 (7), pp. 607-617.
  - I. Broster, A. Burns, G. Rodriguez-Navas, Comparing real-time communication under electromagnetic interference, ECRTS 2004. pp. 45 – 52
  - Broster, I.; Burns, A.; Rodriguez-Navas, G.; Probabilistic analysis of CAN with faults, 23rd IEEE Real-Time Systems Symposium, 3-5 Dec. 2002, pp. 269 – 278
  - L. Pinho et al., « Integrating inaccessibility in response time analysis of CAN networks », Proc. 3rd Workshop on Factory Comm. Systems, Porto, Sept. 2000, pp. 77-84.

©2016, J.-D. Decotignie

References (3)

- improvements
  - G. Cena, A. Valenzano, “Integrating the CAN and MUST access techniques in a single fieldbus”, Proc. 8th int. conf. on Emerging Technologies and Factory Automation, 15-18 Oct. 2001, pp.231 – 239.
  - G. Leen, D. Heffernan, “TTCAN: a new time-triggered controller area network”, Microprocessors and microsystems, Vol. 26, Issue 2, Pages 51-94 (17 March 2002), Pages 77-94.
  - L. Almeida et al.,”The FTT-CAN protocol: why and how”, IEEE Transactions on Industrial Electronics, vol. 49 , Issue: 6 , Dec. 2002, pp.1189 - 1201 .
  - SDS: http://content.honeywell.com/sensing/prodinfo/sds/
  - I. Broster et al., « Comparing real-time communications under electromagnetic interference », Proc. 16th Euromicro Conf. On real-time systems, Catania, Italy, July 2002, pp.
  - CAN-FD: http://www.can-cia.org/can-knowledge/can/can-fd/

©2016, J.-D. Decotignie