Automotive & Embedded Info

Never Forget Basics Whether its Life or Anything Else ... Basics are Cores. While seeing a Tree how can we forget Seed...

CAN & J1939

CAN:

What is CAN Bus?

Can bus is a message based protocol and designed to allow microcontroller and devices to communicate with each other within a vehicle without a host computer. First its uses was in automotive but now its uses in aerospace, maritime, industrial automation & medical equipment's.

Can bus is one of five protocol used in on-board diagnostics OBD-ii vehicle diagnostics standard.

A modern automobile may have 70 ECU (electronic control unit) for various subsystems. Typically the biggest processor is engine control unit; others are used for transmission, airbag, cruise control, audio system, power window, doors mirror adjustment, battery, and electric power steering, antilock braking and recharging system for hybrid cars etc. Some of these form independent system, but communication among others is essential. A subsystem may need to control actuators or receive feedback from sensors. The can standard was devised to fill this need

CAN Technology: –

Can is a multi-master broadcast serial bus for connecting ECU's. Each node is able to send receive message but not simultaneously.

NODE ELEMENT: –

Each node requires:

1. Central processor unit or host processor
2. Can controller (hardware with synchronous clock)
3. Transceiver.

CPU:-

This CPU decides what received messages mean and which messages it want to transmit itself. The sensor, actuator and control devices can connect to CPU.

CAN Controller:-

Receiving: the can controller stores received bits serially from the bus until an entire message is available which can be fetched by the CPU (usually after that can controller has triggered an interrupt to CPU).

Sending: the CPU store it's transmit message to a can controller which transmits the bits serially onto the bus.

CAN Transceiver: –

Receiving: it adapts s/g level from the bus to the level that a can controller expects and has protective serially onto the bus.

Transmitting: it convert the transmit-bit s/g received from the can controller into an s/g that is sent onto the bus.

Speed and Distance: –

1mbit/sec – length below 40 mtr. 125kbit/sec-length 500 mtr.

Model, Arbitration and Logical AND: –

The CAN data transmission uses a lossless bit-wise arbitration method. This arbitration method requires all nodes on the can network to be synchronized to sample every bit on the can network at the same time. This is why some call can synchronous. Unfortunately the term synchronous is imprecise since the data is transmitted without a clock signal in an asynchronous format.

Can is a differential bus, a carrier sense multiple access/bitwise arbitration (CSMA/CD-BA) scheme implemented: if two or more device start transmitting at the same time, there is a priority based arbitration scheme to decide which one will be granted permission to continue transmitting. The can solution to this is prioritized arbitration, making can very suitable for real time prioritized communication systems.

Can features an automatic arbitration-free transmission. A can message that a node transmitted with highest priority will succeed and the node transmitting the lower priority message will sense this and back off and wait.

This is achieved by can transmitting data through binary model of “dominant” bits and “recessive” bits where dominant is a logical 0 and recessive is logical 1. If one node transmits a dominant bit and another node transmits a recessive bit then the dominant bit “wins” (a logical and

During arbitration each transmitting node monitors the bus state and compares the received with transmitted bit. If a dominant bit is received when a recessive bit is transmitted then node stops transmitting (i.e. it lost arbitration). So if a recessive bit is being transmitted while a dominant bit is displayed, the evidence of collision. A dominant bit is asserted by creating a voltage across the wire while a recessive bit is simply not asserted on the bus. If any node sets a voltage a voltage difference, all nodes will see it. Thus there is no delay to the higher priority messages, and the node transmitting the lower priority message automatically attempts to retransmit six bit clocks after the end of the dominant message. Arbitration is performed during the transmission of the identifier field.

ID ALLOCATION: –

Message ids must be unique on a single can bus, otherwise two nodes would continue transmission beyond the arbitration field. Causing an error.

LAYERS: –

1. Data link layer
2. Physical layer.

CAN BIT TIMING: –

Each node in a can network has its own clock and no clock is sent during data transmission. Synchronization is done by dividing each bit of the frame into number of segment: synchronization, propagation, phase 1, and phase 2.

The length of each phase segment can be adjusted based on network & node condition.

CAN Frame: –

1. Data frame,
2. Remote frame,
3. Error frame,
4. Overload frame.

DATA FRAME: –

Data frame is the only frame for actual data transmission. Two format of data frame: 1. Base frame format 2. Extended frame format.

Base Format:

SOF	IDENTIFIER	RTR	IDE	R0	DLC	DATA	CRC	CRCD	ACK	ACKD	EOF
1	11	1	1	1	4	0-64	15	1	1	1	7

Extended Format:

SOF	I-A	SRR	IDE	I-B	RTR	R0 R1	DLC	DATA	CRC	CRC-D	ACK-S	ACK-D	EOF
1	11	1	1	18	1	1	4	0-64	15	1	1	1	7

REMOTE FRAME: –

It is also possible for a destination node to request the data from the source

by sending a remote frame. Two difference between data frame and remote frame. 1. RTR=0 in DF and RTR=1 in RF. 2. No data field in remote frame.

ERROR FRAME: –

Consist of two different fields.

1. The first field is given by superposition of error flags (6-12 dominant/recessive bit) contributed from different stations.
2. The following second field is the error delimiter (8 recessive bit.)

Active error flag: six dominant bits-> transmitted by a node detecting an error on the network that is in error state "error active".

Passive error flag: six recessive bits transmitted by a node detecting an active error frame on the network that is in error state "error passive

OVERLOAD FRAME: –

Contains of two bit fields overload flag and overload delimiter. Overload flag consists of six dominant bits. Overload delimiter consist of eight recessive bit.

There are two kinds of overload conditions that can lead to transmission of overload flag.

1. The internal condition of receiver, which require a delay of the next data frame or remote frame.
2. Detection of a dominant bit during intermission.

INTER FRAME SPACING: –

Two frames are separated from preceding frames by bit field called inter frame space. Inter frame space consist of at least three consecutive recessive bits.

Bit Stuffing: –

Can uses a non-return-to-zero protocol, nrz-5, with bit stuffing. The idea behind bit stuffing is to provide a guaranteed edge on the signal so the receiver can resynchronize with the transmitter before minor clock discrepancies between the two nodes can cause a problem. With nrz-5 the transmitter transmits at most five consecutive bits with the same value. After five bits with the same value (zero or one), the transmitter inserts a stuff bit with the opposite state.

In can frames, a bit of opposite polarity is inserted after five consecutive bits of same polarity. This practice is called bit stuffing. Since bit stuffing is used, six consecutive bits of the same type (111111 or 000000) are considered an error.

Long NRZ messages cause problems in receivers. Clock drift means that if there are no edges, receivers lose track of bits. Periodic edges allow receiver to resynchronize to sender clock.

CAN Arbitration: –

Can arbitration is nothing but the node trying to take control on the bus. CSMA/CD-BA+ amp (arbitration on message priority).two bus nodes have got a transmission request. According to this algorithm both network nodes wait until the bus is free (carrier sense). In that case the bus is free both nodes transmit their dominant start bit (multiple access). Every bus node reads back bit by bit from the bus during the complete message and compares the transmitted value with the received value. As long as the bits

was a difference – the arbitration process takes place.

For example, consider three can devices each trying to transmit messages:

- Device 1 – address 433 (decimal or 00110110001 binary) · device 2 – address 154 (00010011010) · device 3 – address 187 (00010111011)

Assuming all three see the bus is idle and begin transmitting at the same time, this is how the arbitration works out. All three devices will drive the bus to a dominant state for the start-of-frame (SOF) and the two most significant bits of each message identifier. Each device will monitor the bus and determine success. When they write bit 8 of the message id, the device writing message id 433 will notice that the bus is in the dominant state when it was trying to let it be recessive, so it will assume a collision and give up for now. The remaining devices will continue writing bits until bit 5, then the device writing message id 187 will notice a collision and abort transmission. This leaves the device writing message id 154 remaining. It will continue writing bits on the bus until complete or an error is detected. Notice that this method of arbitration will always cause the lowest numerical value message id to have priority. This same method of bit-wise arbitration and prioritization applies to the 18-bit extension in the extended format as well.

Sampling Point: –

Sampling point should be from 50 to 100% of total bit time

but approx. 80% of total bit time will be good.

Sync Seg	Prop Seg	Phase-1	Phase-2

CAN Error: –

CAN ERROR PROCESS:-

1. The error is detected by the can controller (a transmitter or receiver)
2. An error frame is immediately transmitted.
3. The message is cancelled at all nodes.
4. The status of can controller updated.
5. The message is re-transmitted. If several controller have message to send, normal arbitration is used.

CAN ERROR DETECTION: –

1. **Bit error: A**. **Bit stuffing error**: as explained above. **B**. **Bit error**: a TX node always reads back the message as it is sending. If it detects a different bit value on the bus that it sent and bit is not part of the arbitration field or in the acknowledgement field, an error is detected.
2. **Message error: A**: **Checksum error**: each RX node checks message for checksum errors **B**. **Frame error**: there are certain predefined bit value (CRC delimiter, ACK delimiter, end of frame, and also the intermission) that must be transmitted at certain within any can message frame. If a receiver detects an invalid bit in one of these positions a form error will be flagged. **C**. **Acknowledgement error:** if a TX node determines that a message has not been acknowledge then an ACK error if flagged.

CAN ERROR MODES: –

1. Error active: the normal operating mode for a controller. Messages can be received and transmitted. On detecting an error an active error flag is sent.
2. Error passive: a mode entered when the controller has frequent problems transmitting or receiving messages. Messages can be received and transmitted. On detecting an error while receiving, a passive error flag is sent.
3. Bus off: entered if the controller has serious problems with transmitting

messages. No messages can be received or transmitted until the can controller is reset by the host microcontroller or processor.

The mode of the controller is controlled by two error counter the TX error counter and the RX error counter.

The can controller is in active mode if TX_COUNT<=127 &RX_COUNT<=127.

Passive mode is used if (TX_COUNT>127 or RX_COUNT>127) &TX_COUNT<=255.

BUSOFF is entered if TX_COUNT>255.

Once the can controller has entered bus off state, it must be reset by uc in order to be able to continue operation. In addition, this is only allowed after the reception of 128 occurrences of 11 consecutive recessive bits.

CAN ERROR SIGNALLING: –

When an error is detected by a node it sends an error flag on the bus. This prevents any other node from accepting the message and ensures consistency of data throughout the network.

The active error flag consists of six low bits, and is used if the node transmitting the error frame is in active error state. As low is dominant all other nodes will detect bit stuffing violation and send their own error flags? After this, nodes that want to transmit (including the one sending the interrupted message) will start to do so. As usual, the node whose message has the highest priority will win arbitration and send its message.

If the can controller is in error passive mode the error frame will consist of six passive (high) bits. Since the error flag only consists of passive bits, the bus is not affected. If no other node detected an error, the message will be sent uninterrupted. This ensures that a node having problems with

receiving cannot block the bus.

How to recover from can bus off? Once the can controller has entered bus off state, it must be reset by uc in order to be able to continue operation. In addition, this is only allowed after the reception of 128 occurrences of 11 consecutive recessive bits.

CANFD: –

CAN with Flexible data rate.

Benefits: – 1. Increase Bit Rate 2. Increase Payload

Use Case: – 1. Fast Software Download 2. Avoid splitting of long CAN messages. 3. Higher Bandwidth. 4. Baud Rate by limited dimension.

Control Bit Definition: –EDL- Extended data length, r0 and r1- reserved bit, BRS- Bit rate switch, ESI- Error State indicator.

CAN-FD Base Frame: –

SOF	Identifier	R1	IDE	EDL	R0	BRS	ESI	DLC	Data	CRC	ACK	EOF
1	11 bit	1bit	1bit	1bit	1bit	1bit	1bit	4 bit	0-64 bytes	21 bit	2 bit	1 bit

CAN Bit timing: –

Each node in a can network has its own clock, and no clock is sent during data transmission. Synchronization is done by dividing each bit of the frame into a number of segments:

1. Synchronization
2. Propagation time
3. Phase 1 and phase 2.

Each segment consist of a specific, programmable number of time quanta. The length of time quantum Tq=brp/fsys. Fsys=fosc/2.

1. The synchronization segment sync_seg is that part of the bit time where edges of the can bus level are expected to occur; the distance between an edge that occurs outside of sync_seg and the sync_seg is called the phase error of that edge. This segment is always one time quantum long.
2. This part of the bit time is used to compensate physical delay times within the network. These delay times consist of the signal propagation time on the bus and the internal delay time of the can nodes. The segment size is programmable between 1 and 8 time quanta.
3. The phase buffer segments (phase_seg1 and phase_seg2) and the synchronization jump width (SJW) are used to compensate for the oscillator tolerance. The phase buffer segments may be lengthened or shortened by synchronization.

Synchronizations occur on edges from recessive to dominant, their purpose is to control the distance between edges and sample points. Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of sync_seg, otherwise the distance between edge and the end of sync_seg is the edge phase error, measured in time quanta. If the edge occurs before sync_seg, the phase error is negative, else it is positive.

Two types of synchronization exist: hard synchronization and

resynchronization.

Hard Synchronization:

After a hard synchronization, the bit time is restarted with the end of sync_seg, regardless of the edge phase error. Thus hard synchronization forces the edge which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time.

BIT Resynchronization:

Resynchronization leads to a shortening or lengthening of the bit time such that the position of the sample point is shifted with regard to the edge.

When the phase error of the edge which causes resynchronization is positive, phase_seg1 is lengthened. If the magnitude of the phase error is less than SJW, phase_seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

When the phase error of the edge which causes resynchronization is negative, phase_seg2 is shortened. If the magnitude of the phase error is less than SJW, phase_seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.

When the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of hard synchronization and resynchronization are the same. If the magnitude of the phase error is larger than SJW, the resynchronization cannot compensate the phase error completely, an error of (phase error – SJW) remains.

J1939:

What is J1939?

J-1939 used in commercial vehicle for communication. J1939 uses can as a physical layer. Support both peer to peer & broadcast communication. Fix bit rate 250 kbps. Can transmit up to 1765 bytes of data using transport protocol (TP). Support n/w management, manufacturer specific PGS & diagnostics.

Parameter Group: –

A PG is a set of parameters belonging to same topic and sharing the same transmission rate. The length of PG is not limited to length of can frame. PG min length 8 bytes and it can up to 1785 bytes. PG more than 8 bytes require TP for transmission of data.

Interpretation of CAN Identifier: –

The can identifier of j1939 message contains priority, DPB (data page bit), EDPB (ext. Data page bit), PDUF (pdu format), PDUS (pdu specific), TA (target add (if p-p comm)), SA (source address).

PRIORITY	**DPB**	**EDPB**	**PDUF**	**PDUS/TA**	**SA**
3 BIT	1 BIT	1 BIT	8 BIT	8 BIT	8 BIT
	PGN				
ID					

If PDUF< 240 è peer to peer comm, PDUs contains ta. If PDUs>= 240 è broadcast comm. PDUF together with GE in PDUs fields forms the PGN of the transmitted pg.

Parameter group number: –

For a PGN a 24 bit value is used that is composed of 6 bit set to 0 + PDUF + PDUs + DPB + EDPB. Two type of PGN 1. Global PGN 2. Specific PGN

Global: Global PGNs identify PGs that are sent to all. Here the PDUF, PDUs, DPB, EDPB are used for identification of corresponding pg. In this PDUF>= 240 &PDUs is GE.

Specific: Specific PGNs are for PGs that are sent to particular device. Here the PDUF, DPB, EDPB are used for identification of corresponding pg. In this PDUF>= 240 &PDUs is ta.

Suspect Parameter Number: –

A SPN is assigned to each parameter of a pg. SPN is a 19 bit number & has a range from 0 to 524287. It is used for diagnostic purpose to report and identify abnormal operation of a controller application (CA).

Special Parameter Group: –

SAE J1939-21 defines some parameter groups on the data link layer:

Request PG: The request PG (RQST, PGN 00ea00) can be sent to all or a specific CA to request a specified pg. The RQST contains the PGN of the request pg. If the receiver of a specific request cannot respond, it must send a negative acknowledgment. The RQST has a data length code of 3 bytes and is the only PG with a data length code less than 8 bytes.

Acknowledgement PG: The acknowledgement PG (ACKM, PGN 00e800) can be used to send a negative or positive acknowledgment, i.e. in response to a request.

Add Claiming PG: The address claiming PG (ACL, PGN 00ee00) is used for network management.

Command Add PG: The commanded address PG (CA, PGN 00fed8) can be used to change the address of a CA.

Transport Protocol PG: The Transport Protocol PGS (TPCM, PGN 00EC00 and TPDT, PGN 00EB00) are used to transfer PGS with more than 8 bytes.

Network Management: –

The s/w of an ECU is ca. An ECU may contain one or more CAS. Each CA has unique address & associated device name. Each message that is sent by a CA contains this source address. There are 255 possible addresses:

0 TO 253 : VALID SA FOR CAs, **254->**NULL, **255->**GLOBAL.

0 TO 127 & 248 TO 253: USED FOR CAs WITH PREFERRED ADDRESSES & DEFINED FUNCTIONS.

128 TO 247 -> Available for all CAs.

Most CAs like engine, gearbox, etc. Have a preferred address. Before a CA may use an address, it must register itself on the bus. This procedure is called "address claiming." thereby the device sends an "address claim" parameter group (ACL, PGN 00ee00) with the desired source address. This PG contains a 64-bit device name. If an address is already used by another CA, then the CA whose **device name** has the higher priority has claimed the address.

Device Name: The device name contains some information about the ca and describes its main function. A manufacturer code must be requested by the SAE. The values of the fields are defined in SAE j1939.

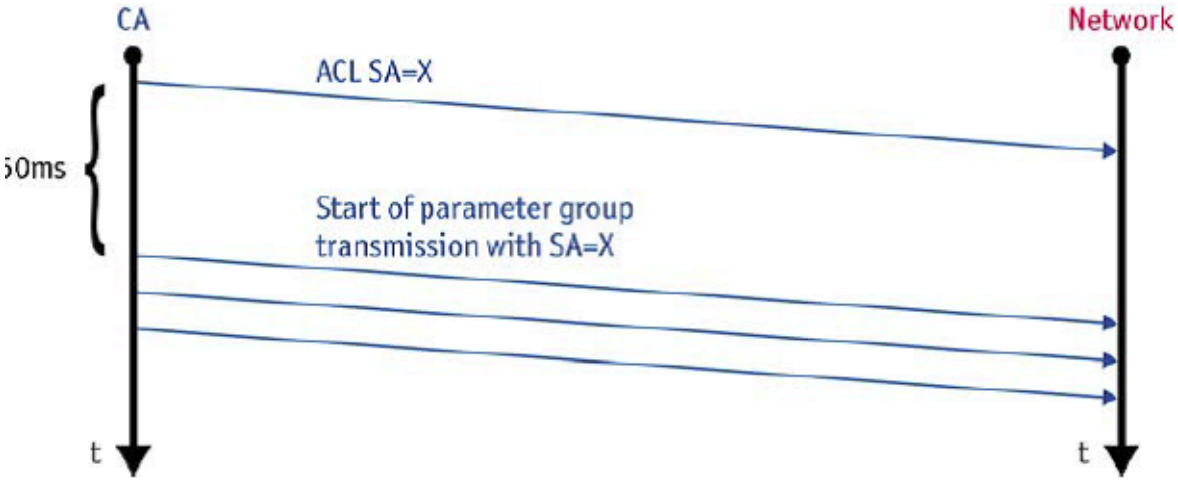
IDENTI	***MANU***	***ECU***	***FUNCTI***	***FUN***	***RES***	***VEC***	***VECHI***	***INDUT***	***A***

FY NUMBER	*FACT. CODE*	*INSTANCE*	*ON INSTANCE*	*CTION*	*VERVE*	*HILE SYSTEM*	*LE INSTANCE*	*RY GROUP*	*AC*
21 BIT	*11 BIT*	*3 BIT*	*5 BIT*	*8 BIT*	*1B*	*7 BIT*	*4 BIT*	*4 BIT*	*1*

Addressing Claiming Procedure: –

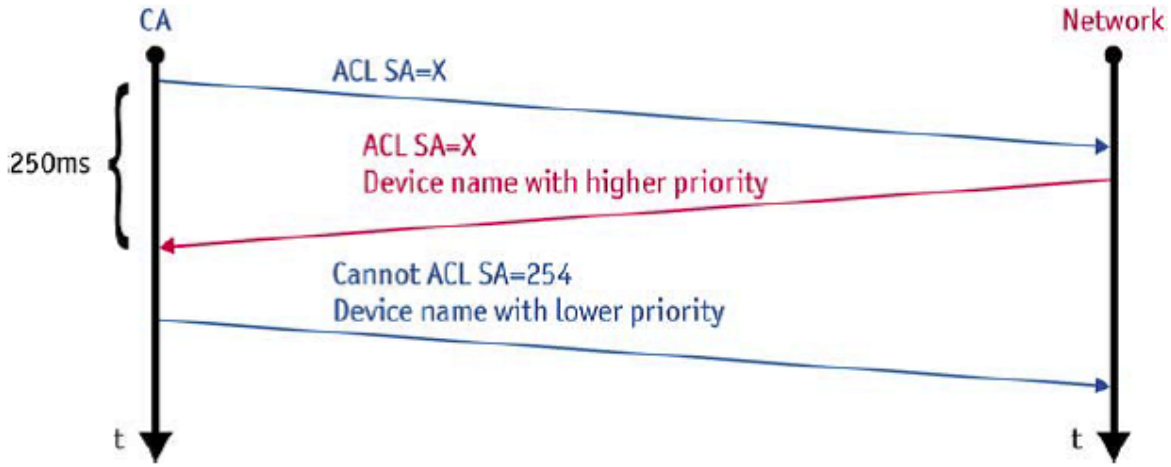
In a common situation the CA sends an address claim PG at start up and waits a defined amount of time. If it does not detect an address conflict it can start with its normal communication.

Figure: Address Claim Procedure



In a situation where another CA already uses the address an address conflict occurs. The CA with the higher priority of the device name will obtain the address. The other CA must send a "cannot claim address" parameter group, with source address null (254).

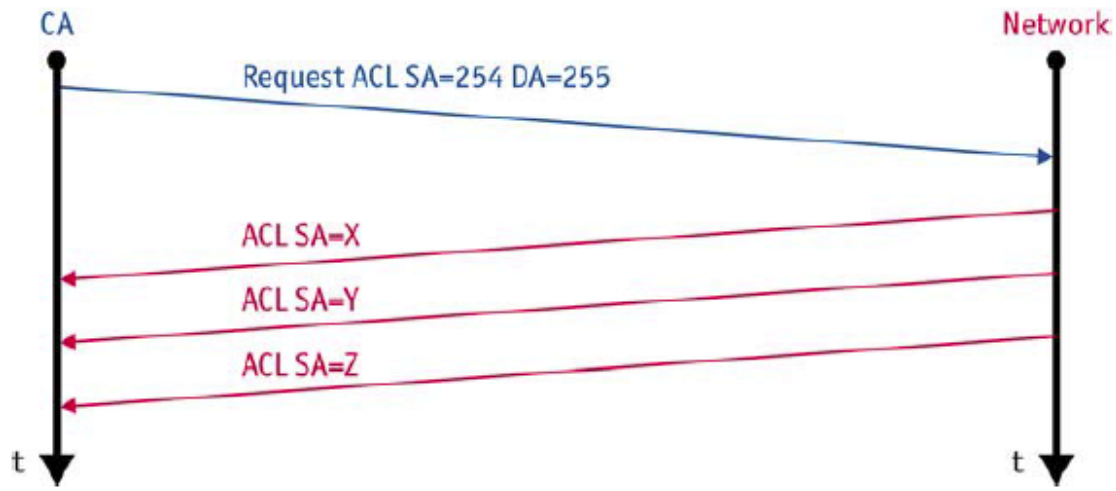
Figure: Address Claiming Procedure With Address Conflict:



Request for Address Claim: –

A CA can detect other CAs in the network by requesting the ACL pg. It is allowed to use the null-address (254) as source address before a CA has performed the address claiming procedure. If the request is addressed to the global address (255) all CAs in the network must respond with the ACL parameter group (including own CA if an address has been claimed already).

Figure: Request Address Claim



Address Capability: –

Arbitrary address capable CA : The CA selects it source address by internal algorithm. It is able to select a new address on an address conflict. The arbitrary address capable field in the device name indicates this capability.

Single address capable CA : A CA of this type can use only one address. On address conflicts it is not able to select another address.

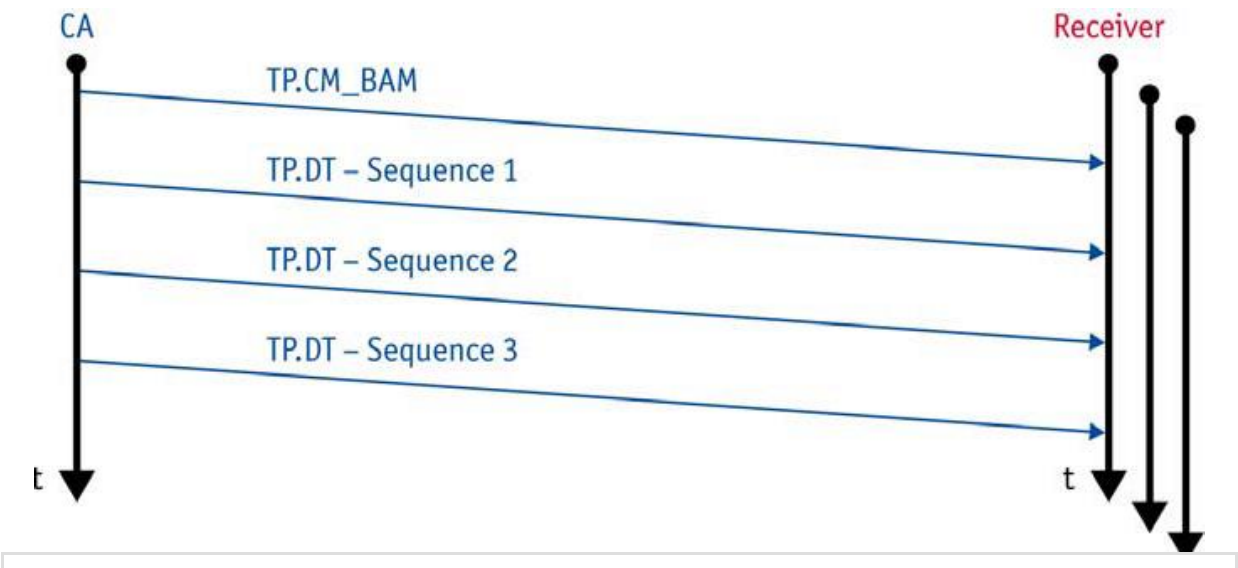
Transport Protocol: –

Parameter groups that contain more than 8 data bytes are transmitted by means of a transport protocol.

For peer-to-peer and broadcast transmission, there are two different protocols. The transport protocols utilize two special parameter groups which are used for the connection management (tp.cm) and the transmission of the data (tp.dt).

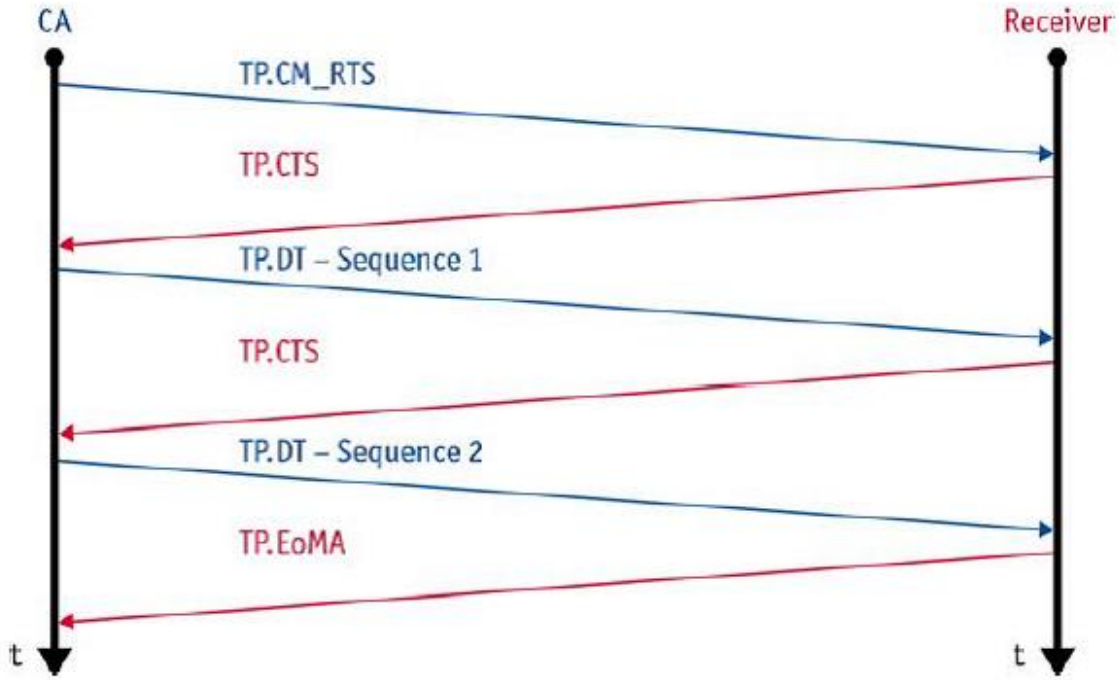
For broadcast transmission, the bam (broadcast announce message) protocol is used. Here, after a bam-PG, the transmitter sends all data PGs at a minimum interval of 50ms.

Figure: BAM Transmission



With the peer-to-peer transmission, the transmitter initiates the connection with a "REQUEST TO SEND" message. The receiver then controls the transport protocol with "CLEAR TO SEND" and finally acknowledge it with "END OF MESSAGE ACKNOWLEDGE."

Figure: CTS/RTS Transmission



Share this:

Twitter Facebook

Like

Be the first to like this.

Automotive & Embedded Info / Powered by WordPress.com.