Universitatea Politehnica Timişoara

Lecture 6. CAN Higher Layer Protocols

SAE J1939, Time-Triggered CAN and CAN Calibration Protocol

Why the need for CAN higher layer protocols?

- CAN follows the Open systems Interconnect (OSI) reference model and includes specification for two layers: physical and data link
- Standard CAN doesn’t specify behaviour for other OSI layers

- Some applications also include functionality for these other OSI layers

Application Layer (7)
Presentation Layer (6)
Session Layer (5)
Transportation Layer (4)
Network Layer (3)
Data Link Later (2)
Physical Layer (1)

CAN-based higher layer protocols

- **DeviceNet** – IEC 62026-3
- **SAE J1939**
- **CANopen** – EN 50325-4
- **CAN Calibration Protocol (CCP)** – AE MCD 1
- **Time-Triggered CAN (TTCAN)** – ISO 11898-4
- **NMEA 2000** – IEC 61162-3
- **OBD II / ISO-TP** – ISO 15765

SAE J1939

- The SAE J1939 protocol is defined by the Society of Automotive Engineers (SAE)
- Higher layer protocol for commercial vehicles used for standardized communication between ECUs from different manufacturers
- Covers five of the OSI layers: physical, data link, network transportation and application

SAE J1939 – Main features

- Implemented at the application layer
- Uses extended CAN frames (29-bit identifiers)
- Standardized bit rates: 250 and 500 kbit/s
- Supports point-to-point and global addressing
- Support for sending multi-packet messages
- Standardized messages covering general communication for main ECUs
- Manufacturer-specific definition of messages possible
- Diagnostic functionality

J1939 Device NAME

- All network nodes involved in J1939-compliant communication requires an unique identifier
- This identifier is defined as a 64 bit value called the NAME
- The NAME contains various useful information about the node like: manufacturer, functionality, etc.
- The NAME is used in the allocation of dynamic addresses

1 bit	3 bit	4 bit	7 bit	1 bit	8 bit	5 bit	3 bit	11 bit	21 bit
AAC	IG	VSI	System	r	Function	Function Instance	ECUI	Manufacturer Code	Identity Number

AAC - Arbitrary Address Capable
IG - Industry Group
VSI - Vehicle System Instance
r - reserved
ECUI - ECU Instance

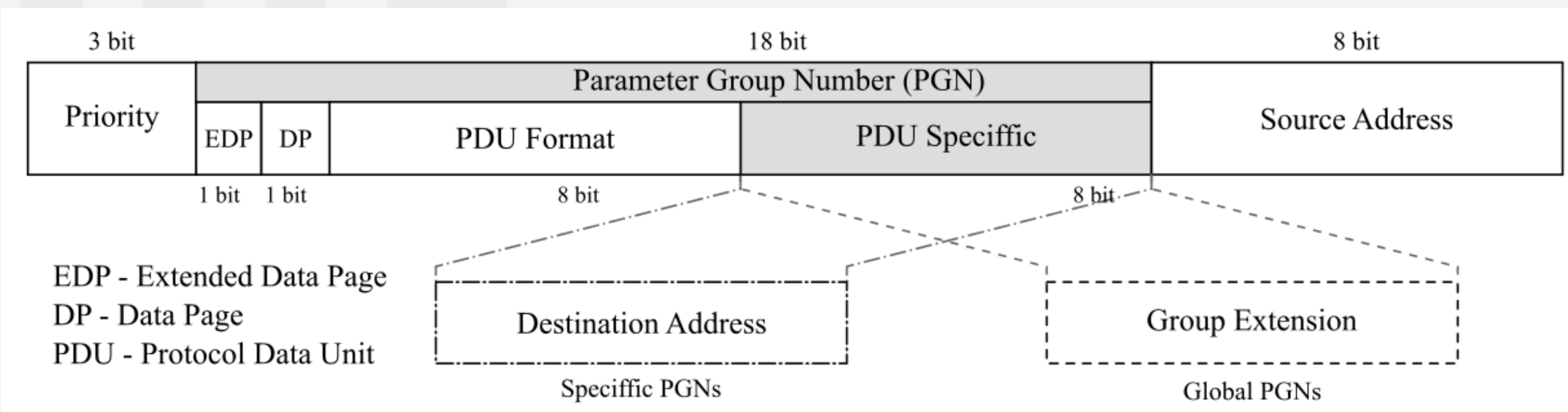
J1939 Device address

- Each J1939 node should have a valid 8 bit address allocated
- Addresses can be allocated statically (hard coded on each node) or dynamically (will be allocated before communication can start)
- In particular, for nodes that implement multiple functionalities (controller applications) an address is required for each of these controller applications

Address	Details
0-253	Standard communication addresses. The first 127 addresses are reserved for particular device functions
254	NULL address – defined for ECUs with no valid addresses
255	Global address – used for global addressing

J1939 CAN identifier

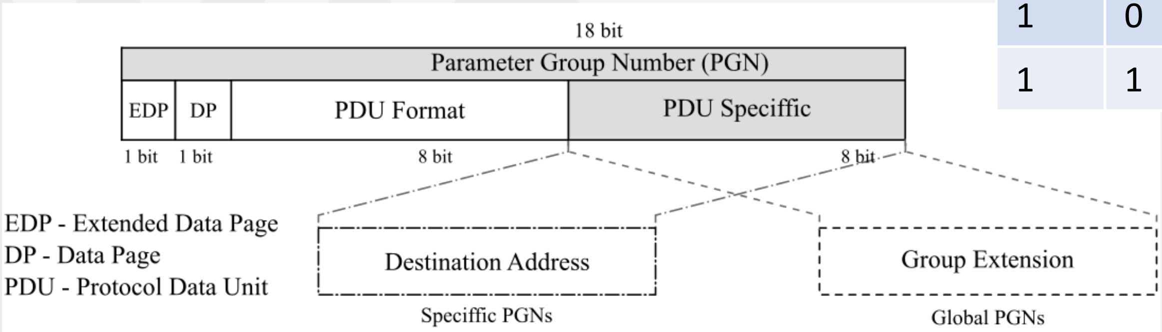
- The 29-bit identifier has 3 main parameters:
  - Priority – 3 bits for establishing priority (0 highest 7 smallest)
  - Parameter Group Number (PGN) – defines the communication context
  - Source Address – address of the sender node



Parameter Group Number

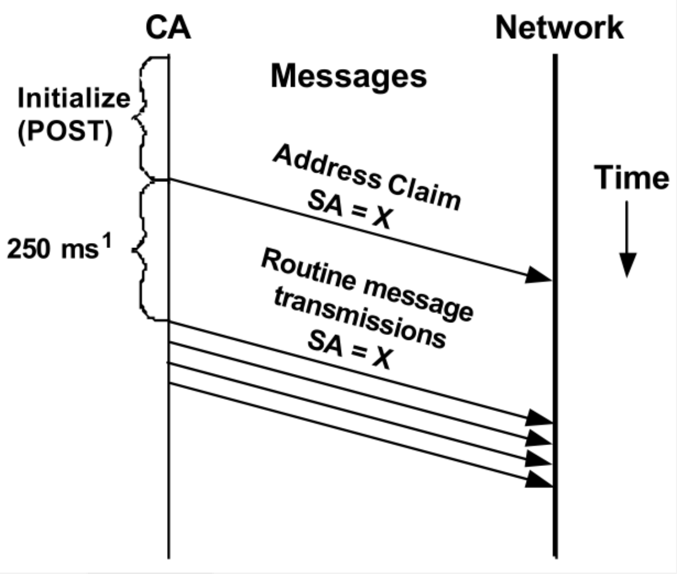
- Extended Data Page (EDP) and Data Page (DP) bits define 4 pages dedicated to specific uses (see table below)
- PDU Format
  - if < 240 -> PDU Specific should contain destination address
  - If ≥ 240 the message is a global message and PDU Specific should be interpreted as the group expansion

EDP	DP	Description
0	0	SAE J1939 parameter groups
0	1	Defined by NMEA 2000
1	0	Reserved for SAE J1939
1	1	Defined by ISO 15765-3



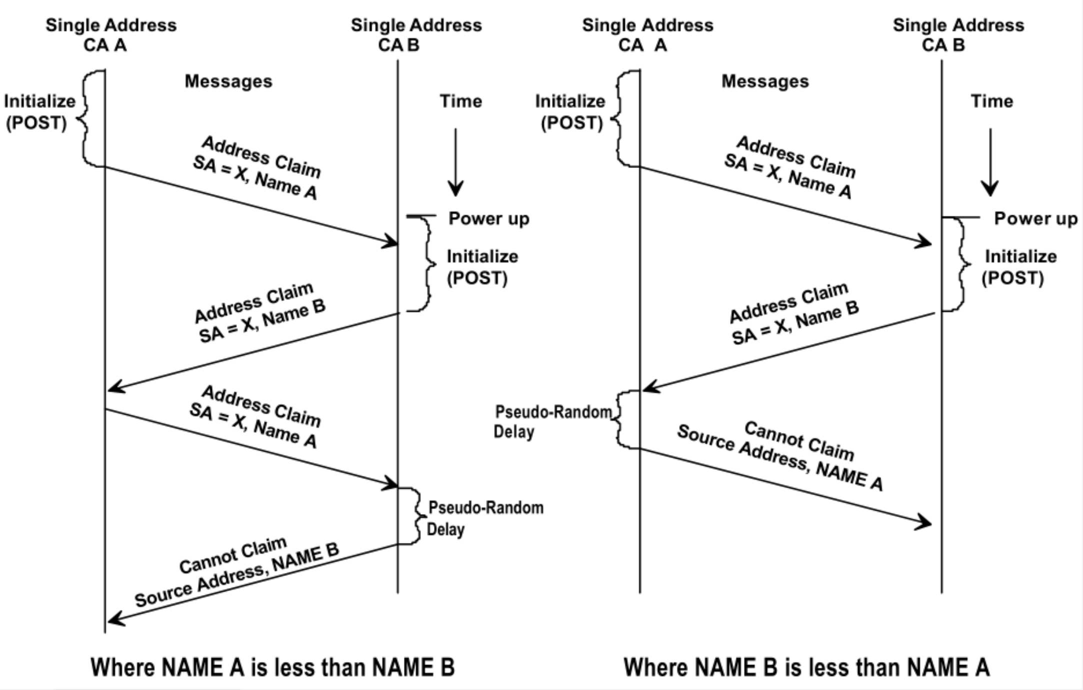
J1939 Address Claiming

- For dynamic address allocation a node sends an Address Claim message to all nodes on the network.
- The address claim is considered successful if there is no answer from another node requesting the same address



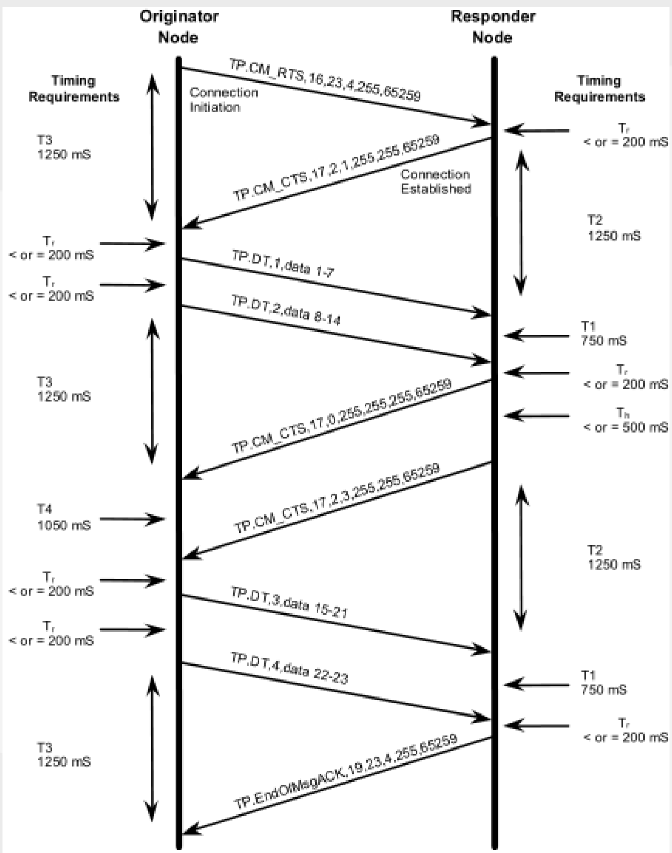
J1939 Address Claim conflict

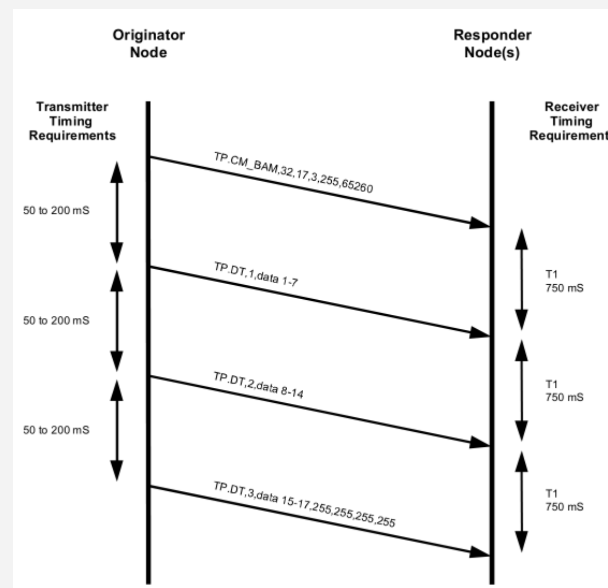
- When another node claims the same address it will answer with an Address Claim message
- The node with the highest priority NAME field wins the address claim
- The loosing node sends a Cannot Claim message and will try claiming another address



J1939 Transport Protocol

- J1939 specifies a transport protocol for sending multi-frame messages (remember the 8 byte payload limit in CAN)
- Transport protocol available for point-to-point and global messages





Time-Triggered CAN

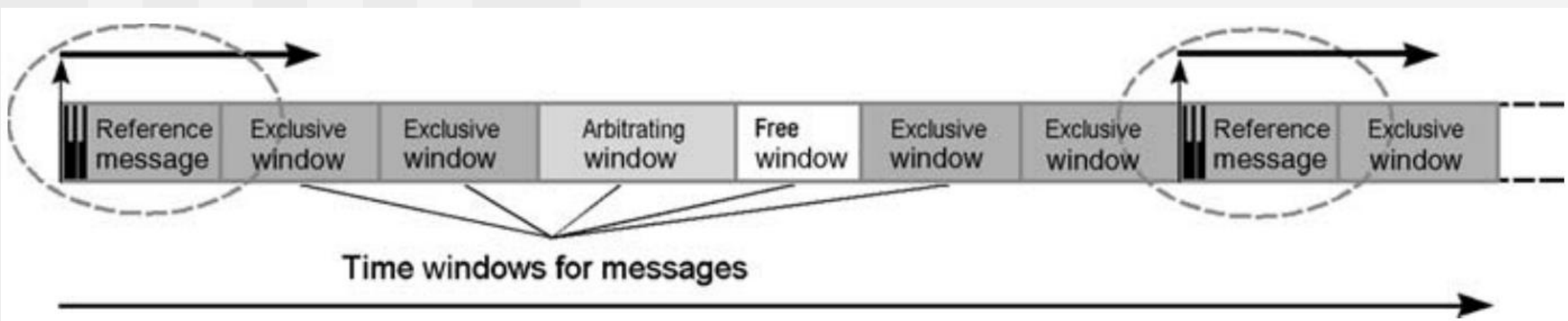
- It was developed due to the demand for time-triggered communication in real-time applications
- Proposed by the CAN in Automation (CiA) group and Bosch and currently specified in the ISO 11898-4 standard
- TTCAN is located mainly in the Session layer of the OSI stack

TTCAN session

- The TTCAN implementation of the session layer provides services needs to support a session-bases communication between two entities
- Functions are provided for basic actions such as: initialization, synchronization, dialogue termination and recovery services

TTCAN operating principle

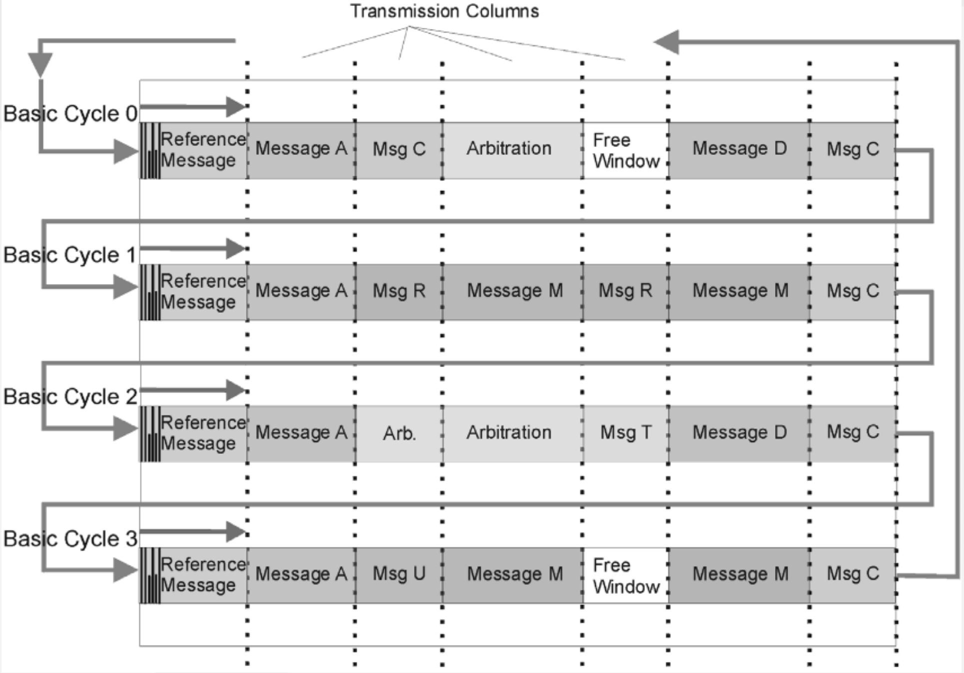
- The main operating principle of TTCAN is defined based on time windows and operation cycles
- One network node is responsible for organizing time division and time window allocation
- A TTCAN basic cycle contains three types of time windows:
  - Exclusive window – should be used for periodic messages
  - Arbitrating window – should be used for occasional messages
  - Free window – free scape for any kind of traffic



Dominique Paret, Multiplexed Networks for Embedded Systems, Wiley 2007.

TTCAN Schedule matrix

- The TTCAN schedule is represented as a matrix in which each row represents a basic cycle.
- Cell represents messages that should be sent in the specified slot
- Each basic cycle starts with the transmission of a Reference Message – sent by the Time Master node



Lars-Berno Fredriksson, TTCAN explained, KVASER AB

CAN Calibration Protocol (CCP/XCP)

- The CCP protocol is defined by the Association for Standardization of Automation and Measuring Systems (ASAM)
- It’s intended for enabling the calibration of ECUs providing read and write access to network nodes at runtime
- XCP (Universal Measurement and Calibration Protocol) was developed to extend CCPs usage on other bus systems: CAN, CAN-FD, SPI, SCI, Ethernet, USB, FlexRay

CCP concept

- CCP functions as a single master/multi slave system
- The node performing the measurement and calibration operations assumes the role of the master
- Target ECUs represent the slaves

- Each node must have an unique station address
- A connection has to be established between the master and the slave

CCP messages

- CCP only uses 2 type of messages:
  - Command Receive Object (CRO) – master to slave

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CMD	CTR	Data	Data	Data	Data	Data	Data

  - Data Transmission Object (DTO) slave to master

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
PID	ERR	CTR	Data	Data	Data	Data	Data

CMD – Command code
CTR – Command counter
PID – Packet Identifier
ERR – Error code
Data – Additional parameter or don’t care

CCP commands

Command	Code	TimeOut to ACK [ms]	Remark
CONNECT	0x01	25	
GET_CCP_VERSION	0x1B	25	
EXCHANGE_ID	0x17	25	
GET_SEED	0x12	25	optional command
UNLOCK	0x13	25	optional command
SET_MTA	0x02	25	
DNLOAD	0x03	25	
DNLOAD_6	0x23	25	optional command
UPLOAD	0x04	25	
SHORT_UP	0x0F	25	optional command
SELECT_CAL_PAGE	0x11	25	optional command
GET_DAQ_SIZE	0x14	25	
SET_DAQ_PTR	0x15	25	
WRITE_DAQ	0x16	25	
START_STOP	0x06	25	
DISCONNECT	0x07	25	
SET_S_STATUS	0x0C	25	optional command
GET_S_STATUS	0x0D	25	optional command
BUILD_CHKSUM	0x0E	30 000	optional command
CLEAR_MEMORY	0x10	30 000	optional command
PROGRAM	0x18	100	optional command
PROGRAM_6	0x22	100	optional command
MOVE	0x19	30 000	optional command
TEST	0x05	25	optional command
GET_ACTIVE_CAL_PAGE	0x09	25	optional command
START_STOP_ALL	0x08	25	optional command
DIAG_SERVICE	0x20	500	optional command
ACTION_SERVICE	0x21	5 000	optional command