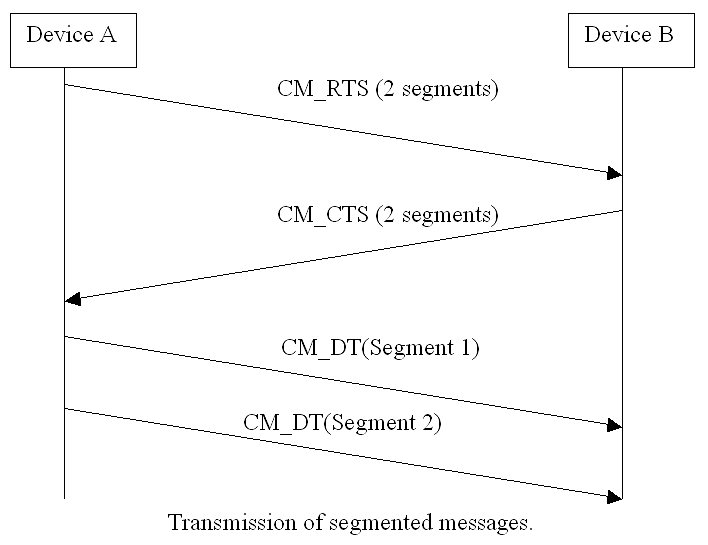
**J1939**

J1939 messages are organized in to **Protocol Data Unit**s (PDU) which consists of an identifier and 8 data bytes. Numerical data that is larger than a single byte is sent with least significant byte (LSB) first. J1939 uses CAN 2.0B with the extended (29 bit) identifier. The CAN identifier consists of a priority(3 bits), a reserved (1 bit), a data page (1 bit), PDU format (one byte), PDU specific (one byte) and source address (one byte).

There are two different PDU formats. PDU1 format is used for sending messages with a specific destination address. PDU2 format can only be sent broadcast. The PDU format byte in the identifier determines the message format. If the PDU format byte is less than 240 (0xF0) then the format is PDU1 and if it is greater than 239 it is PDU2.

The transport protocol provides functions for transmitting messages longer than 8 bytes since they will not fit in a single CAN frame. In J1939, these messages are segmented into 7 byte packets with a sequence number for each packet. There are special Connection Management (CM) messages for handling the communication of segmented messages. Examples of these messages are: Request to Send (RTS), Clear to Send (CTS) and Broadcast Announce Message (BAM). The segmented messages can be sent to a specific device or as broadcast. CM messages provide a virtual connection and a handshake procedure between the sender and receiver. The sender requests to send a segmented message and the receiver answers with how many segments it can receive for the moment (figure 4). Segmented messages can be sent as broadcast as well with no handshaking between sender and receiver(s



**----->** When a message must be directed to a particular device, a specific destination address can be included within the message identifier. For example, a request for a specific torque value from the engine instead of a specific torque value from the brake controller.

J1939 uses the 29-bit identifier defined within the CAN 2.0B protocol shown in Figure 1. The identifier is used slightly different in a message with a destination address (”PDU 1”) compared to a message intended for broadcast (”PDU 2”).

**----->** PDU stands for Protocol Data Unit (i.e. Message Format).

The SOF, SRR, and IDE bits are defined by the CAN standard and will be ignored here. The RTR bit (remote request bit) is always set to zero in J1939.

The 29-bit identifier used in J1939 is structured in the following way.

| Priority | Reserved | Data page | PDU format | PDU specific | Source Address |
| --- | --- | --- | --- | --- | --- |
| 3 bits | 1 bit | 1 bit | 8 bits | 8 bits | 8 bits |

Table 1: Structure of a 29-bit identifier.

The first three bits of the identifier are used for controlling a message’s priority during the arbitration process. A value of 0 has the highest priority. Higher priority values are typically given to high-speed control messages, for example, the torque control message from the transmission to the engine. Messages containing data that is not time critical, like the vehicle road speed, are given lower priority values.

The next bit of the identifier is reserved for future use and should be set to 0 for transmitted messages.

The next bit in the identifier is the data page selector. This bit expands the number of possible Parameter Groups that can be represented by the identifier.

The PDU format (PF) determines whether the message can be transmitted with a destination address or if the message is always transmitted as a broadcast message.

The interpretation of the PDU specific (PS) field changes based on the PF value:

* If the PF is between 0 and 239, the message is addressable (PDU1) and the PS field contains the destination address.
* If the PF is between 240 and 255, the message can only be broadcast (PDU2) and the PS field contains a Group Extension.

The Group extension expands the number of possible broadcast Parameter Groups that can be represented by the identifier.

The term Parameter Group Number (PGN) is used to refer to the value of the Reserve bit, DP, PF, and PS fields combined into a single 18 bit value.

Example: The ID 0xCF004EE can be divided into the following fields in table 2.

| 0x0C | | | | | 0xF0 | | 0x04 | | 0xEE |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 000 | 011 | 0 | 0 | 11110000 | | 00000100 | | 11101110 | |
| — | Prio | R | DP | PF | | PS | | SA | |

Table 2. PGN example.

* PGN = the R, DP, PF and PS fields – in this case 0x0F004.
* PF = 0xF0 = 240, i.e. this is a PDU2 (broadcast) message
* PS = 0x04, i.e. the Group Extension = 4

The last 8 bits of the identifier contain the address of the device transmitting the message. The address is the label or “handle” which is assigned to provide a way to uniquely access a given device on the network. For a given network, every address must be unique (254 available). This means that two different devices (ECUs) cannot use the same address.

* The PGN uniquely identifies the Parameter Group (PG) that is being transmitted in the message.
* Each PG (a grouping of specific parameters) has a definition that includes the assignment of each parameter within the 8-byte data field (size in bytes, location of LSB), and the transmission rate and priority of the message.
* The structure of a PGN permits a total of up to 8672 different parameter groups to be defined per page.
* When an ECU receives a message, it uses the PGN in the identifier to recognize the type of data that was sent in the message

**How are diagnostics supported?**

A Diagnostic Message (DM) may be sent and/or received by an ECU. If the ECU supports diagnostic messaging, each type of potential fault in the module will have associated with it a Diagnostic Trouble Code (DTC).

A DTC is a combination of four independent fields: the Suspect Parameter Number (SPN) of the channel or feature that can have faults; a Failure Mode Identifier (FMI) of the specific fault; the occurrence count (OC) of the SPN/FMI combination; and the SPN conversion method (CM) which tells the receiving mode

how to interpret the SPN. Together, the SPN, FMI, OC and CM form a number that a diagnostic tool can use to understand the failure that is being reported. When an ECU detects a fault, it will send an Active Diagnostic Trouble Code, DM1, message. The DM1 message send by the ECU will also contain the status lamps of the module. While the fault is still present, the ECU will continue to broadcast the DM1 message to the network every second. When the fault clears, the ECU will send a final DM1 message showing that there are no further faults present. If multiple faults are present simultaneously, the ECU will send all the active faults in a single DM1 by using the Broadcast Announce Message (BAM) in a Transport Protocol session. Refer to a product's datasheet to see if it will send a DM1, and what type of DTCs it supports. An ECU on the bus may respond to the data in the DM1, or a diagnostic tool may be connected to the network to show an operator all the active faults on the network.

-------->When an ECU receives a message, it uses the PGN in the identifier to recognize the type of data that was sent in the message. <-----