7 Network layer interface

7.1 General information

This part of ISO 15765 makes use of the network layer services defined in ISO 15765-2 for the transmission and reception of diagnostic messages. This section defines the mapping of the Application layer protocol data units (A_PDU) onto the Network layer protocol data units (N_PDU).

NOTE The network layer services are used to perform the application layer and diagnostic session management timing (see 6.3).

7.2 FlowControl N_PCI parameter definition

The client shall not use the values of F1 hex – F9 hex for the Stmin parameter. These Stmin parameter values shall be supported by the server(s) if requested by the vehicle manufacturer.

7.3 Mapping of A_PDU onto N_PDU for message transmission

The parameters of the application layer protocol data unit defined to request the transmission of a diagnostic service request/response are mapped in accordance with Table 9 onto the parameters of the network layer protocol data unit for the transmission of a message in the client/server.

The network layer confirmation of the successful transmission of the message (N_USData.con) is forwarded to the application, because it is needed in the application for starting those actions, which shall be executed immediately after the transmission of the request/response message (ECUReset, BaudrateChange, etc.).

Table 9 — Mapping of ServiceName.request/ServiceName.response A_PDU onto N_USData.request N_PDU

A_PDU parameter (Application Protocol Data Unit)	Description	N_PDU parameter (Network Protocol Data Unit)	Description
A_SA	Application Source Address	N_SA	Network Source Address
A_TA	Application Target Address	N_TA	Network Target Address
A_Tatype	Application Target Address type	N_Tatype	Network Target Address type
A_RA	Application Remote Address	N_AE	Network Address Extension
A_PCI.SI	Application Protocol Control Information Service Identifier	N_Data[0]	Network Data
A_Data[0] – A_Data[n]	Application Data	N_Data[1] N_Data[n+1]	Network Data

7.4 Mapping of N_PDU onto A_PDU for message reception

The parameters of the network layer protocol data unit defined for the reception of a message are mapped in accordance with Table 10 onto the parameters of the application layer protocol data unit for the confirmation/indication of the reception of a diagnostic response/request.

The network layer indication for the reception of a FirstFrame N_PDU (N_USDataFirstFrame.ind) is not forwarded to the application, because it is only used within the application layer to perform the application layer timing (see 6.3). Therefore, no mapping of the N_USDataFirstFrame.ind N_PDU onto an A_PDU is defined.

Table 10 — Mapping of N_USData.ind N_PDU onto ServiceName.conf/ServiceName.ind A_PDU

N_PDU parameter (Network Protocol Data Unit)	Description	A_PDU parameter (Application Protocol Data Unit)	Description
N_SA	Network Source Address	A_SA	Application Source Address
N_TA	Network Target Address	A_TA	Application Target Address
N_TAtype	Network Target Address type	A_TAtype	Application Target Address type
N_AE	Network Address Extension	A_RA	Application Remote Address
N_Data[0]	Network Data	A_PCI.SI	Application Protocol Control Information Service Identifier
N_Data[1] N_Data[n+1]	Network Data	A_Data[0] - A_Data[n]	Application Data

8 Standardized diagnostic CAN identifiers

8.1 Legislated 11 bit OBD CAN identifiers

The 11 bit CAN identifiers for legislated OBD can also be used for enhanced diagnostics (e.g. the functional request CAN Id can be used for the functionally addressed TesterPresent (3E hex) request message to keep a non-defaultSession active).

If the 11 bit CAN Identifiers as specified in ISO 15765-4 are re-used for enhanced diagnostics, then the following requirements apply:

a) network layer timing parameters according ISO 15765-4 shall also apply for enhanced diagnostics;

b) the DLC (CAN data length code) shall be set to eight (8) and the CAN frame shall include eight (8) bytes (unused bytes shall be padded).

NOTE ISO 15765-4 allows for max. 8 OBD related servers (ECUs); therefore, 11 bit CAN identifiers for max. 8 servers are defined.

8.2 Legislated 29 bit OBD CAN identifiers

The 29 bit CAN identifiers for legislated OBD comply with the Normal fixed addressing format specified in ISO 15765-2 and can also be used for enhanced diagnostics.

If the 29 bit CAN Identifiers as specified in ISO 15765-4 are re-used for enhanced diagnostics, then the following requirements apply:

a) network layer timing parameters as specified in ISO 15765-4 shall also apply for enhanced diagnostics;

b) the DLC shall be set to eight (8) and the CAN frame shall include eight (8) bytes (unused bytes shall be padded).

NOTE The CAN identifier values given in the tables use the default value for the priority information in accordance with ISO 15765-2.

8.3 Enhanced diagnostics 29 bit CAN identifiers

8.3.1 General information

This section specifies a standardized addressing and routing concept for CAN using 29 bit identifiers. The concept makes use of the well-known and approved mechanisms of the internet protocol (IP). By this means, standardized algorithms for addressing and routing can be used for all nodes in the whole network independent of their positioning in subnetworks.

This addressing and routing concept provides the following features:

— maximum flexibility during the design process of network structures,

— full customization of network and node address,

— the possibility of CAN controller hardware filter feature optimization by the assignment of the appropriate network and node address,

— gateways need to know only network addresses of the connected sub-networks instead of all addresses of their sub-network members.

The following specifies the technical details of the CAN identifier structure, the structure of addresses, and subnet masks. A detailed description of the algorithms used for routing and broadcasting is also included.

8.3.2 Structure of 29 bit CAN identifier

The 29 bit CAN identifier structure specified in this document is compatible in regard to coexistence with the definitions in ISO 15765-2, ISO 15765-3 and ISO 15765-4 and with SAE J1939-21. Therefore, the encoding of bit 25 (Reserved/Extended Data Page) and bit 24 (Data Page) in the 29 bit CAN identifier structure defined in SAE J1939-21 shall be used to determine whether a CAN identifier and frame is of SAE J1939 or ISO 15765 format. This enables the vehicle network designer to define non-diagnostic messages and associated CAN identifiers customized according to his needs or to utilize and benefit from the definitions in SAE J1939 in combination with a diagnostic services implementation as defined in ISO 15765-2, ISO 15765-3 and ISO 15765-4.

8.3.2.1 Structure of SAE J1939 29 bit CAN identifier

For information about the structure of the SAE J1939 29 bit CAN identifier format, see Table 11.

Table 11 — SAE J1939 structure of 29 bit CAN identifiers

29 bit CAN identifier													
28	27	26	25	24	23		16	15		8	7		0
Priority			Reserved/ Extended data page	Data page	PDU Format			PDU-specific (destination or PDU format extension)			Source address (unique source address)		

8.3.2.2 Structure of ISO 15765 29 bit CAN identifier

Table 12 shows the structure of ISO 15765 CAN identifier that can be distinguished from the SAE J1939 format through the "SAE J1939 Reserved/Extended Data Page and ISO 15765 Extended Data Page" bit 25 and the "SAE J1939 Data Page ISO 15765 Data Page" bit 24. Thus, ISO 15765-formatted and SAE J1939-formatted 29 bit CAN identifiers can coexist on the same physical CAN bus system without interference.

Table 12 — ISO 15765 structure of 29 bit CAN identifiers

29 bit CAN identifier										
28	27	26	25	24	23	22	21	11	10	0
Priority			Extended data page	Data page	Type of service (TOS)		Source address		Destination address	
			Encoding see 8.3.2.4		Encoding see 8.3.2.5		Unique source address, see 8.3.3		Unique destination address, see 8.3.3	

8.3.2.3 Priority field

The priority field is defined as specified in SAE J1939, to make use of the arbitration mechanism of CAN. Because the CAN identifier can no longer be assigned freely (source and target address are included in CAN identifier), the priority of a CAN message would be assigned by the sender (source address) and the receiver (target address) of that message indirectly. Eight (8) different priority levels are possible.

Priority level 6 (110b) shall be assigned to diagnostic request messages/frames.

8.3.2.4 Extended Data Page and Data Page field

The Extended Data Page and Data Page bits determine which format of the 29 bit CAN identifier shall be used. Table 13 specifies the encoding.

Table 13 — Definition of Extended Data Page and Data Page field

Extended data page bit 25	Data page bit 24	Description
0	0	SAE J1939-defined or manufacturer-defined "Normal Communication Message" strategy if SAE J1939 is not implemented
0	1	SAE J1939-defined or manufacturer-defined "Normal Communication Message" strategy if SAE J1939 is not implemented
1	0	SAE J1939-reserved or manufacturer-defined "Normal Communication Message" strategy if SAE J1939 is not implemented
1	1	ISO 15765-3-defined

8.3.2.5 Type of service (TOS) field

The type of service field is used to be able to address different services of a node without having to assign different addresses to it. Thus, eight (8) different service types of a node can be addressed concurrently using a single destination address. The different types of services and their usage are defined in Table 14.

Table 14 — Definition of Types Of Service (TOS)

Bit 23	Bit 22	Type Of Service (TOS)	Description
0	0	ISO reserved	This bit combination is reserved for future use by ISO.
0	1	OEM-defined messages	This bit combination indicates that the messages are OEM-specific. A combination of ISO 15765-3 and legacy protocol messages can be used to support a mixture of servers on the same network with different protocol messages.
1	0	Network control message protocol / network management	This bit combination indicates that the frame(s) contain data sent and received by gateways to supply information about the current state of subnets (e.g. network unreachable, network overload) and nodes (e.g. host unreachable).
1	1	ISO 15765-3-defined messages	This bit combination indicates an ISO 15765-3-defined diagnostic service addressed to a node. The user data bytes of the CAN frame contain diagnostic requests (ISO 15765-3) using the network layer services and transport layer defined in ISO 15765-2.

8.3.2.6 Source address

The source address contains the address of the sending entity. This information ensures the correct arbitration and can be used by the receiver of a message to address its replies. The structure of the source address is described in 8.3.3.

8.3.2.7 Destination address

The destination address contains the address of the receiving entity. This can be a single node, the broadcast address of a network or a generic broadcast. The destination address is used by gateways to determine whether the CAN frame shall be routed to another CAN bus or not. The structure of the target address is described in 8.3.3.

8.3.3 Structure of address

8.3.3.1 General information

The source and destination addresses are encoded in the 29 bit CAN identifier with a length of 11 bits each. In the following subclauses, the letters "X" and "Y" are used to represent a variable parameter.

8.3.3.2 Definition of address

An address consists of two parts.

a) Network address

The network address part consists of the first "X" sequential bits of the address and determines a node's network. The same network address shall be assigned to the nodes on one physical bus. The network address part shall not have all bits set to one (1). Thus, the minimum length for the network address part is two (2) bits. The maximum length is nine (9) bits because at least two (2) bits are needed to provide valid node address parts. The maximum number of possible subnets can be calculated as follows:

$2^X - 1$ (where X is the number of bits used for the network address part)

b) Node address

The node address part consists of the remaining "Y" (Y = 11 − X) sequential bits of the address and determines the node within a subnet. It shall be unique within the subnet. All bits set to zero (0) and all bits set to one (1) are not allowed. Thus, the minimum length of the node address part is two (2) bits. The maximum length is nine (9) bits because at least two (2) bits are needed for the network address part. The maximum number of nodes per sub-network can be calculated as follows:

$2^Y - 2$ (where Y is the number of bits used for the node address part)

A node is assigned a unique address that shall be stored in the node's internal memory. A node shall receive messages with the node's assigned address in the destination address field.

Table 15 presents an example for source and destination addresses. The sending and the receiving nodes are on different sub-networks.

Table 15 — Example for source and destination address

29 bit CAN identifier																												
28	27	26	25	24	23	22	21										11	10										0
Priority 0x6			ISO 15765 format		Type of service ISO 15765-3 messages		Source address 0x2ED											Destination address 0x32F										
1	1	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	0	0	1	0	1	1	1	1

8.3.3.3 Subnet mask

The subnet mask assigns the number of bits used for the network address part and for the node address part.

The length of the subnet mask is 11 bits (same as the length of the address). The value of a subnet mask is assigned by setting the first "X" sequential bits set to one (1). The number of sequential bits set to one (1) selects the network address part from the whole address. The remaining sequential bits set to zero (0) select the node address part from the whole address (see Table 16 and Table 17 for examples of subnet masks for sender and receiver).

Due to the fixed length of a subnet mask and the first "X" sequential bits set to one (1), only the number of bits set to one (1) needs to be stored instead of the whole bit mask. Thus, a short notation is used to define a subnet mask.

Table 16 — Example for sender's subnet mask

Subnet mask										
10	9	8	7	6	5	4	3	2	1	0
0x7C0 (short notation: /5)										
Network address part					Node address part					
1	1	1	1	1	0	0	0	0	0	0

Table 17 — Example for receiver's subnet mask

Subnet mask										
10	9	8	7	6	5	4	3	2	1	0
0x7E0 (short notation: /6)										
Network address part						Node address part				
1	1	1	1	1	1	0	0	0	0	0

Each node is assigned a subnet mask that shall be stored in its internal memory. Nodes of the same subnet are assigned the same subnet mask.

8.3.3.4 Network address

The network address of a node can now be calculated using its assigned address and subnet mask. Therefore, a simple bit by bit AND operation of address and subnet mask shall be performed. See Tables 18 and 19 for examples of determining the network address of sender and receiver.

Table 18 — Example for sender's network address

Source address											
Bit	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Address: 0x2ED	0	1	0	1	1	1	0	1	1	0	1
Subnet mask: /5	1	1	1	1	1	0	0	0	0	0	0
Network address: 0x2C0	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**

Table 19 — Example for receiver's network address

Destination address											
Bit	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Address 0x32F	0	1	1	0	0	1	0	1	1	1	1
Subnet mask: /6	1	1	1	1	1	1	0	0	0	0	0
Network address: 0x320	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**

To describe a subnet, its network address and subnet mask are noted in the following form:

<network address> / <short subnet mask notation>

For the given examples this results in

sender's subnet: 0x2C0 / 5

receiver's subnet: 0x320 / 6

This information is used by gateways for routing.

8.3.3.5 Broadcast address

8.3.3.5.1 Generic broadcast (0x7FF)

The generic broadcast allows for broadcasting messages to all nodes of a network. To send a broadcast to the whole network, the target address 0x7FF [all bits set to one (1)] shall be used. A message with that target address will be routed by all gateways. All nodes on the network shall receive and process messages with destination address 0x7FF.

8.3.3.5.2 Subnet broadcast

The subnet broadcast is intended to be used for broadcasting messages to the nodes of a specific sub-network. To send a broadcast to a specific subnet, the broadcast address of that subnet shall be calculated. This is done by taking the destination's subnet information (network address and subnet mask) and setting all node address part bits [marked with zero (0) in subnet mask] to one (1). See Table 20 for a subnet broadcast example for the receiver's subnet.

Table 20 — Example for subnet broadcast to receiver's network

Destination address											
Bit	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Network address: 0x320	0	1	1	0	0	1	0	0	0	0	0
Subnet mask: /6	1	1	1	1	1	1	0	0	0	0	0
Broadcast address: 0x33F	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**

Subnet broadcast messages are normally routed by gateways.

All nodes have to receive messages with the network address part equal to their own network address part and all bits set to "1 in the node address part of the destination address field.

8.3.4 Message retrieval

Each node on a subnet compares the destination address of a CAN frame with its own address. If those match, the information contained is transferred to the next higher layer in the OSI model for further processing.

8.3.5 Routing

8.3.5.1 General information

Routing applies whenever nodes from physically disconnected subnets communicate with each other and their CAN frames have to be transferred from one subnet to another subnet. This is performed by additional nodes, which are physically connected to the network where the CAN frame is received and the network where the CAN frame shall be transmitted to, to reach its destination. Thus, a CAN frame may pass several gateways from its source subnet to its destination.

8.3.5.2 Network and subnet structure

Generally, networks can be designed as needed when the following restrictions are respected:

— addresses shall be unique;

— all nodes in a subnet have the same subnet mask;

— all nodes in a subnet have the same network address;

— whenever a network address is assigned to a subnet, no further network addresses in that address scope may be assigned to other networks, as this would result in a routing problem.

Figure 13 shows a configuration with four (4) subnets connected to a gateway. Three (3) subnets are connected through one gateway and the 4th subnet is connected through an additional gateway.

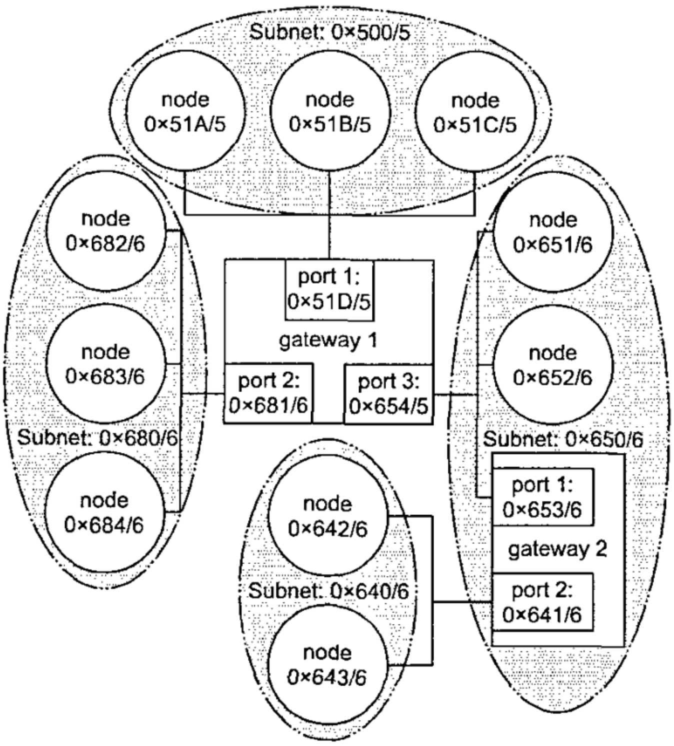


Figure 13 — Network configuration example

8.3.5.3 Gateways and routing

8.3.5.3.1 Description

Gateways are nodes connected to more than one subnet and therefore able to transfer CAN frames from one subnet to another.

8.3.5.3.2 Ports

A port is the connection of a gateway to one physical subnet. A gateway shall have at least two (2) ports. Each port is assigned a network address and subnet mask of the subnet it is connected to.

In Figure 13, the configuration includes two (2) gateways, where gateway 1 has three ports and gateway 2 has two ports.

8.3.5.3.3 Routing table

To determine whether a CAN frame needs to be routed, a routing table shall be generated and stored in the gateway's memory. A routing entry contains the network address, subnet mask and the port on which the subnet can be reached. Such an entry shall exist for each subnet that is connected (directly or indirectly) through this gateway.

See Table 21 for the network shown in Figure 13. Through hierarchical design of the networks 640/6 and 650/6, the routing table entries can be reduced to one entry 640/5.

Table 21 — Routing table example

subnet (network address/subnet mask)	port
Gateway 1	
500/5	1
680/6	2
640/5	3
Gateway 2	
500/5	1
680/6	1
650/6	1
640/6	2

8.3.5.3.4 Routing algorithm

A gateway receives all messages from the ports that are connected to the different subnets. If the gateway is an addressable node, then only one address out of the address scopes of the subnets connected directly to the ports of that gateway shall be assigned. An additional message retrieval check is performed before the proper routing algorithm. If the destination address is 0x7FF, the message is copied to all ports except the one on which the message was received. The normal routing algorithm is skipped.

After having received Message A, the routing steps shall be as shown in Figure 14.

8.3.5.3.5 Routing example

In Figure 15, a routing example is shown for a CAN message from the client with the address 0x51A to the server with the address 0x642 using the routing information from Table 21.

The following steps are performed by the gateways on reception of that message.

a) **Gateway 1**

1) Analysis of CAN-ID: DA = 0x642. See Tables 22 and 23.

Table 22 — Gateway 1 routing decision

Routing decision	Network	Port
(0x642 AND 0x7C0) = 0x640 != 500 → no local message → routing	500/5	1

Table 23 — Gateway 1 routing analysis

Routing analysis	Network	Port
0x642 AND 0x7E0 = 0x640 != 680 → next entry	680/6	2
0x642 AND 0x7C0 = 0x640 = 640 → correct path	640/5	3

2) Check of whether the message is addressed to gateway: 0x642 != 0x654.

3) Forwarding of message to port 3.

b) **Gateway 2**

1) Analysis of CAN-ID: DA = 0x642. See Tables 24 and 25.

Table 24 — Gateway 2 routing decision

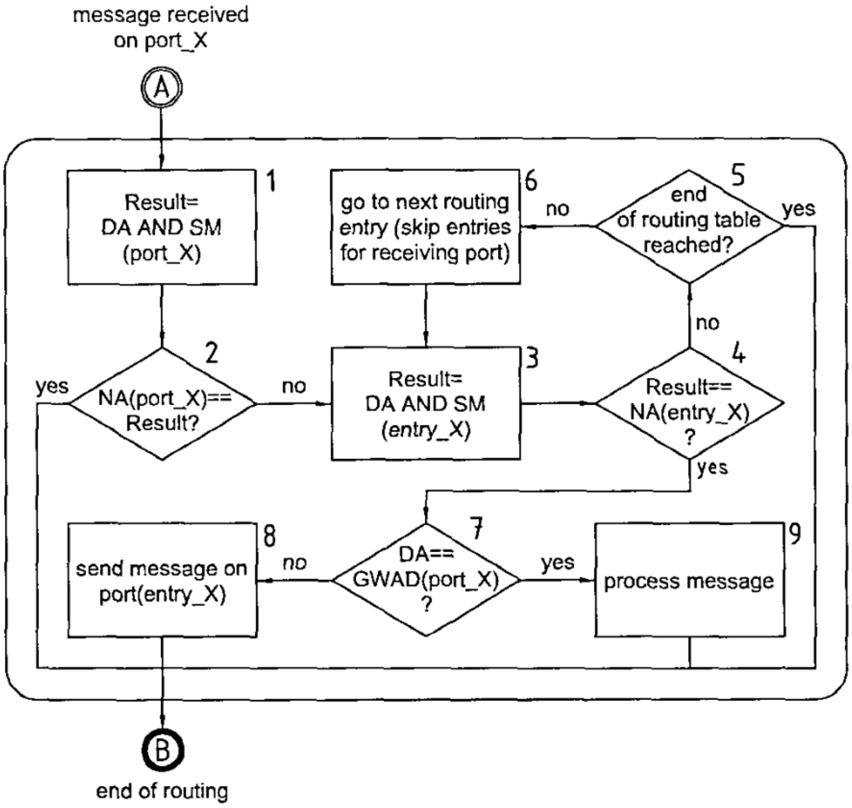
Routing decision	Network	Port
0x642 AND 0x7C0 = 0x640 != 650 → no local message → routing	650/6	1

Table 25 — Gateway 2 routing analysis

Routing analysis	Network	Port
0x642 AND 0x7E0 = 0x640 = 640 → correct path	640/6	2

2) check whether the message is addressed to gateway: 0x642 != 0x641.

3) forward message to port 2.



Steps

A Message received on port "X".

1 Bit by bit logical AND operation is performed with destination address of the received message and subnet mask of the port on which the message was received.

2 The result is compared with the network address of the port on which the message was received. The network address of the port is either stored in the node's memory or can be calculated using the address and subnet mask of that port. If the result and the network address are equal, the received message is a local message of the port's subnet and no routing will apply (B). If the result and the port's network address are different, a routing analysis shall be performed, continuing with step 3.

3 A bit by bit logical AND operation is performed with destination address of the received message and the subnet mask of the current routing table entry.

4 The result of the operation and the network address of the current routing table entry are compared. If those matches the algorithm will continue with step 8, otherwise the algorithm will continue with step 5.

5 If there are additional routing table entries, the algorithm will continue with step 6. Otherwise no routing will apply (B).

6 The next routing table entry is selected and the algorithm jumps back to step 3.

7 The destination address of the message is compared with the gateway's address on the current port. This step is only needed if the gateway is an addressable node, otherwise the algorithm jumps directly to step 8. If the destination address is the address of the gateway for the current port, the algorithm continues with step 9. If destination address and address of the gateway are not equal, the algorithm is continued at step 8.

8 The message is sent on the port of the routing table entry that matched the network address of the destination address.

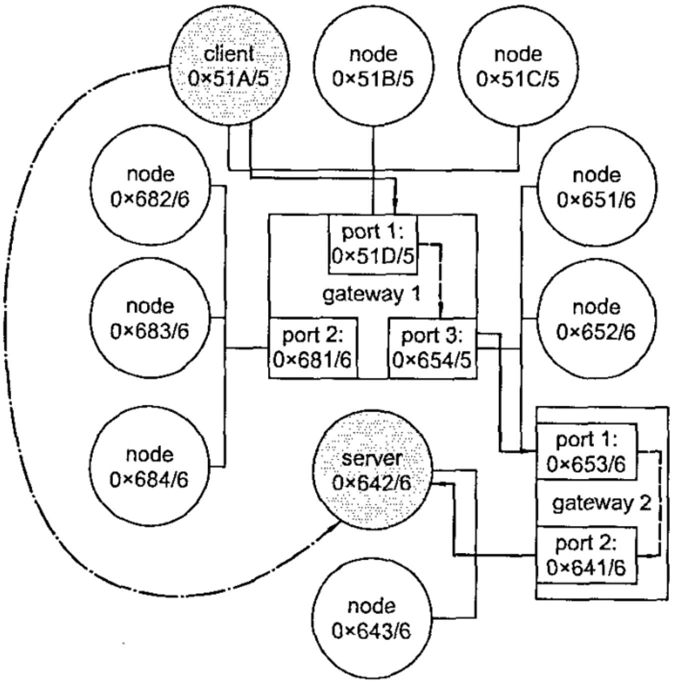
9 The message was addressed to gateway node and thus, it is processed by application.

B End of routing algorithm.

Key

DA destination address
GWAD gateway's address on port_X
NA network address
SM subnet mask
entry_X entry #X in the gateway's routing table
port_X port #X of the gateway

Figure 14 — Routing algorithm sequence chart



Key

—·— virtual connection
——— message path on CAN bus
– – – message path in gateway

Figure 15 — Routing example from client 0x51A to server 0x642

9 Diagnostic services implementation

9.1 Unified diagnostic services overview

This clause defines how the diagnostic services as defined in ISO 14229-1 apply to CAN. For each applicable service, the applicable subfunction and data parameters are defined.

NOTE The subfunction parameter definitions take into account that the most significant bit is used for the suppressPosRspMsgIndicationBit parameter as defined in ISO 14229-1.