INTERNATIONAL STANDARD

ISO 22902-4

First edition
2006-11-01

Road vehicles — Automotive multimedia interface —

Part 4: **Network protocol requirements for vehicle interface access**

Véhicules routiers — Interface multimédia pour l'automobile —

Partie 4: Exigences du protocole de réseau pour accès à l'interface du véhicule

Reference number
ISO 22902-4:2006(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 22902-4 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

ISO 22902 consists of the following parts, under the general title *Road vehicles — Automotive multimedia interface*:

— *Part 1: General technical overview*

— *Part 2: Use cases*

— *Part 3: System requirements*

— *Part 4: Network protocol requirements for vehicle interface access*

— *Part 5: Common message set*

— *Part 6: Vehicle interface requirements*

— *Part 7: Physical specification*

Introduction

This part of the standard network architecture provides a set of common interfaces for accessing network-connected devices and vehicle functions through the vehicle interface. This network architecture has two elements:

— network protocol requirements for vehicle interface access;

— Common Message Set (CMS).

The CMS is the companion to the network protocol requirements: in general, the CMS specifies semantic requirements; the network protocol specifies syntactical requirements. It should be recognized that the network protocol requirements are focused on supporting access to vehicle services; the CMS contains – in addition to vehicle services – semantic requirements for audio visual (AV) messages, phone messages, Human Machine Interface (HMI) messages, etc.

As shown in Figure 1, vehicle interface is a component that is a proxy of the vehicle functions: interfacing objects from functional modules may interact with the device it represents. The object abstracts and exposes the services of devices in the vehicle. Objects called doors, windows, lights and vehicle speed, correspond to the devices connected to an automaker's proprietary network.

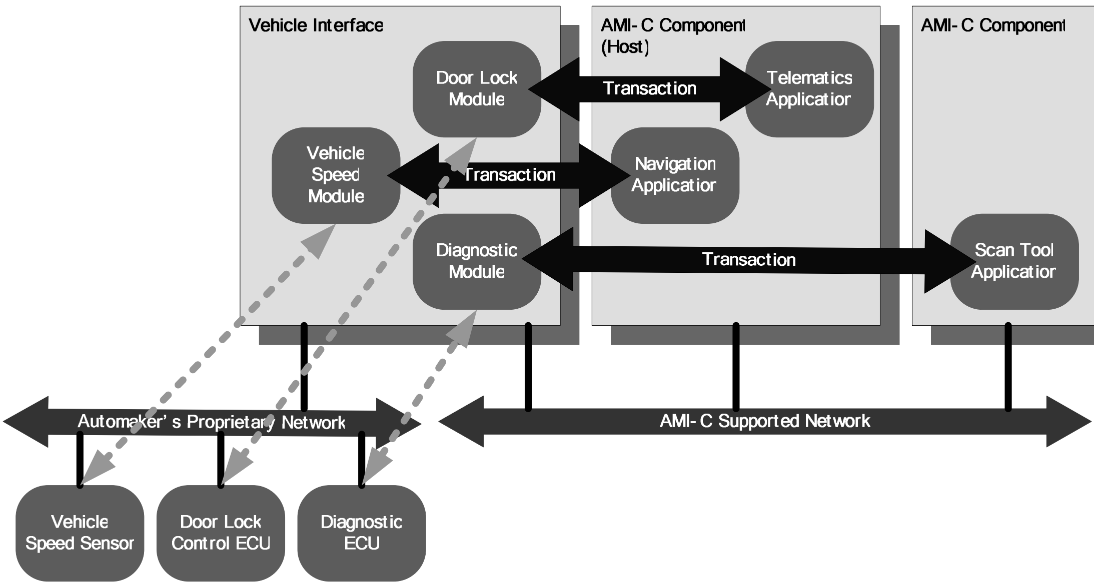

Figure 1 — Vehicle interface (example)

This network protocol requirement for vehicle interface access is a network framework and high-level (meta) protocol that enables the abstraction of devices and thereby facilitates their communication with applications and networks. The AMI-C network protocol requires that each device object consist of a network interface (called its network adaptation layer) to ensure compatibility with different networks and also several functional modules. This requirement makes it possible for network applications to access a vehicle interface independently from specific network technology. Additionally, the network technology must be able to support multiple application level protocols on the network devices.

AMI-C network protocol requirements for vehicle interface access define a Vehicle Interface Protocol (VIP) that can be instantiated on a network technology. A VIP is an application protocol to access a vehicle's devices (functions) via the network transport protocol of AMI-C networks. This, in turn, allows devices on such an AMI-C network to communicate with an automaker's proprietary vehicle network through that vehicle interface.

AMI-C network protocol requirements for vehicle interface access apply to automotive multimedia networks that do NOT have existing protocols for transporting vehicle function messages to and from a vehicle interface. For example, network transport layers are supposed to be general: TCP (UDP)/IP, FCP for 1394 Automotive, or L2CAP for Bluetooth. However, MOST Coorporation has already defined how to transport vehicle functions over the MOST network. Hence, the AMI-C network protocol requirements for vehicle interface access do not apply to the MOST network.

For specific network technologies, the CMS is instantiated into technology-specified message frames. A specific network technology may already include the functionality of some portion of the CMS. In those cases the CMS is not instantiated. Rather, the CMS permits a semantic mapping from AMI-C specifications to the messages of that particular network technology.

Road vehicles — Automotive multimedia interface —

Part 4:
Network protocol requirements for vehicle interface access

1 Scope

This document provides a communication model that contains the requirements for a Vehicle Interface Protocol (VIP) to access a vehicle interface over a network transport protocol for AMI-C networks. It does not apply to networks that have pre-existing protocols and messages for transporting vehicle functions.

A VIP defines how an application communicates over a simple network transport mechanism. These requirements can be applied to network technologies that use UDP/IP as the transport method. However, pre-existing systems may have unique protocols and messages for transporting messages about vehicle functions; therefore, this document does not cover such pre-existing technology.

Messages transported are network-specific instantiations of the CMS.

This document addresses the following aspects related to the AMI-C's approach to network communication:

— message frame format;

— application transaction;

— system transaction;

— initialization process;

— address resolution; and

— functional module discovery and removal process.

2 Network architecture

2.1 Outline

The Network protocol requirements for vehicle interface access, which consists of component, functional modules, API, network application layer, and network transport layer. The vehicle interface is considered one of its components.

2.2 Component

A component is a physical entity attached to an AMI-C network (1394 Automotive, Bluetooth, and so on). A component contains one or more functional modules, the network application layer and the network transport layer. A component is identified within a network transport layer with a physical ID, such as node ID, AM address, or IP address.

2.3 Vehicle interface

For those services resident on the automaker's network, the AMI-C vehicle interface (VI) is a component that provides access to vehicle services via an AMI-C-compliant network. It may act as a gateway to a network defined by a vehicle manufacturer or it may implement some or all of the vehicle services directly. Vehicle information and services such as vehicle speed, door lock, windows, and diagnostic services are members of the automaker's proprietary network and a functional module on the VI represents each device. There may be more than one vehicle interface component on a network.

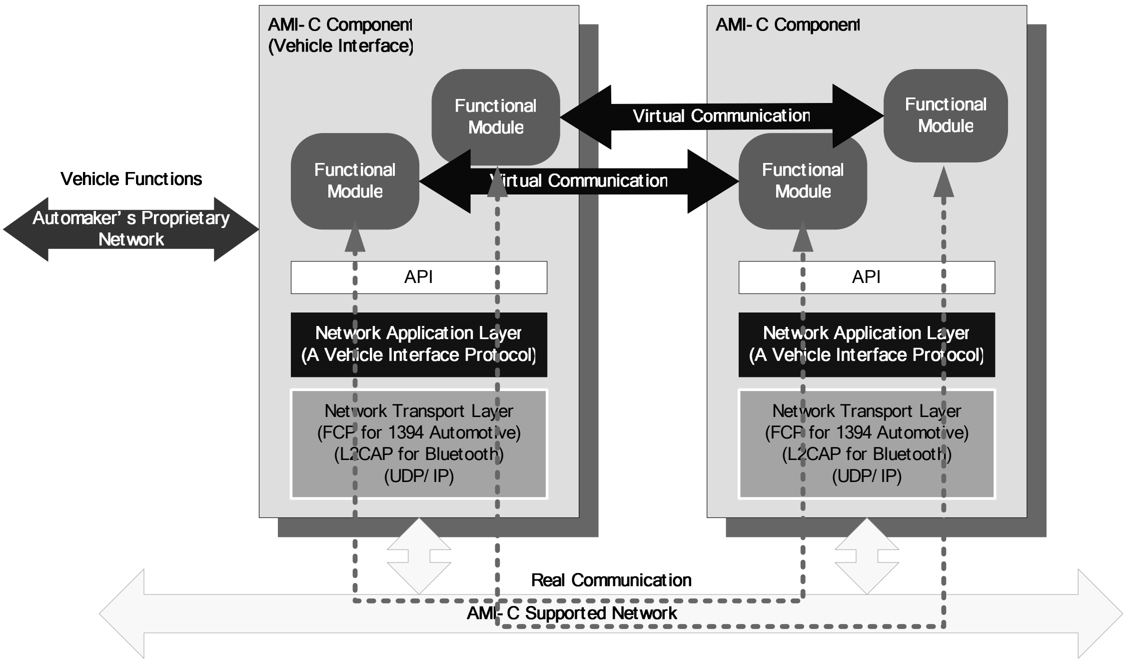

Figure 2 — AMI-C meta protocol conceptual architecture

2.4 Network application layer

A network application layer provides the interface through an API for functional modules to access transport layers (and also, as necessary, lower layers) of network protocol. This document establishes the requirements for the application layer protocol. The instantiation of the application layer requirements into a specific network technology is the vehicle interface protocol for that network.

A VIP allows devices on an AMI-C network to communicate with an automaker's proprietary vehicle network through a vehicle interface.

2.5 Network transport layer

Common network transport layers include TCP(UDP)/IP, FCP for 1394 Automotive, or L2CAP for Bluetooth. MOST has defined a layer for transporting vehicle functions.

2.6 Functional module

Within the AMI-C network architecture, a functional module (FM) is an abstraction of a device. An AMI-C component can have one or more FMs. An FM defines the properties of a device (e.g., the up/down position of a car window). It also defines commands (e.g., set position) that control the properties of a device. An FM is addressed through a logical address, composed of function type (F-Type) and instance number (I-Num).

There can be more than one FM with the same F-Type (4 windows, for example). To distinguish between these identical F-Types, I-Nums are statically or dynamically assigned.

2.7 Component control module (CCM)

Each AMI-C component (including the VI) must have a component control module (CCM). The CCM is the FM that performs initialization, network management, address resolution, service discovery, as well as registry maintenance and update. The I-Num of a CCM serves as the Component ID.

A CCM in each component communicates with other FMs to exchange attributes and status regarding FM installed in its components during initialization or when a component joins the network at the first time or leaves the network. In addition, the CCM is able to reply to service discovery queries from remote functional modules about all FMs inside its own component and component hardware status. The CCM keeps track of the status information in a registry, such as the power status of a remote module, or visibility of a remote module in the network.

2.8 Registry

A Registry is an address-mapping table between the Logical Address of an FM (that is, F-Type and I-Num) and the network-specific ID of the AMI-C component containing this FM (e.g., node number in the case of 1394 Automotive).

The registry contains this matching information about remote and local FMs throughout the AMI-C Network. It is beyond the scope of this specification to define the format of the registry, and how often it should be updated.

3 Addressing

There are three types of addressing: unicast, broadcast and multicast.

3.1 Unicast

Unicast addressing defines a point-to-point transaction.

Unicast is characterized by F-Type values from ’00’H to ‘EF’H, and I-Num values from ‘00’H to ‘FE’H.

Unicast is the most common type of transaction. For example, an application (hosted by an FM) needs information that can be supplied by a remote FM. It sends an INQUIRE message to that remote FM and waits for the corresponding REPORT message from that remote FM. The logical address (F-Type and I-Num) of the destination FM is specified in the header. The Logical Address of the source FM is also specified in the header so that the destination FM can respond to the source component (See Figure 3), illustrates a unicast transaction.

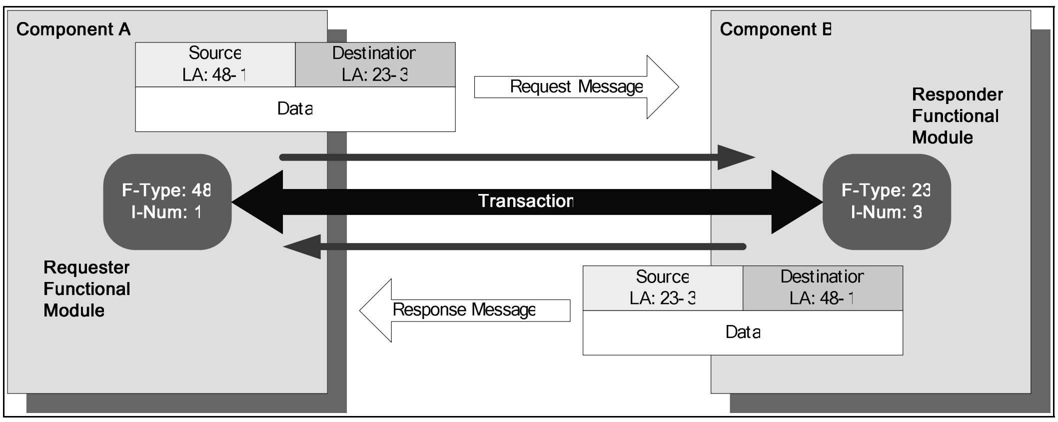

Figure 3 — Unicast transaction mechanism

3.2 Broadcast

The network protocol requirements for vehicle interface access define 3 types of broadcast transactions:

- a broadcast message sent from an FM to all the instances of the same F-Type, e.g. lock all windows. In order to address all the instances of a particular F-Type, the I-Num field is given the maximum value: ‘FF’H. F-Type field specifies the Function Type.

- a broadcast message sent from a FM to all the FMs of the AMI-C Network. In order to address all the FMs of the network, the F-Type field is given the maximum value: ‘FF’H and I-Num field is ‘FF’H.

- a broadcast message for an address request or response (e.g. request to know which component contains a specific FM). When the system field (Sys, see section 4.2) equals 1B, this indicates that the message is a system transaction; therefore, it is always a broadcast message. The subsequent response will also have its Sys field set.

3.3 Multicast

Multicast is used for subscribed information. A component will request information from another component to be sent periodically. The responder component shall then allocate a multicast group ID and start to send the packets with the requested information.

A multicast packet is identified by its F-Type destination field (equal to ‘FE’H) and its I-Num field is equal to the multicast group ID (‘00’H to ‘FE’H).

The message class is used by all the receiving components to indicate the nature of the information being sent (for example the dashboard sends a message containing vehicle speed information at a regular interval. The F-Type indicates that the message comes from the dashboard. The object property indicates that the nature of the information: Vehicle Speed).

The following Table 1 summarizes the roles of the F-Type and I-Num fields.

NOTE Refer to Table 3 in clause 5.1 for the format of the full message.

Table 1 — Logical addressing

	F-Type	I-Num	Logical Addressing for Destination
Unicast and Broadcast to Specific Functional Module(s)	00	00	Unicast to/from the F-Type ‘00’H, I-Num ‘00’H module
	00	01	Unicast to/from the F-Type ‘00’H, I-Num ‘01’H module
	...	...	...
	00	FF	Unicast to/from the F-Type ‘00’H, I-Num ‘EF’H module
	01	00	Unassigned I-Num for F-Type ‘01’H
	01	01	Unicast to/from the F-Type ‘01’H, I-Num ‘01’H module
	...	...	...
	01	FE	Unicast to/from the F-Type ‘01’H, I-Num ‘FE’H module
	...	...	...
	...	...	...
	EF	00	Unassigned I-Num for F-Type ‘EF’H, I-Num ‘00’H module
	EF	01	Unicast to/from the F-Type ‘EF’H, I-Num ‘01’H module
	...		...
	EF	FE	Unicast to/from the F-Type ‘EF’H, I-Num ‘FE’H module
Dynamic Addressing	F0	00	Dynamic addressing mechanism (reserved for AMI-C Host)
	...	...	
	FD	FF	Dynamic addressing mechanism (reserved for AMI-C Host)
Multicast	FE	00	Multicast to Functional Modules with Multicast Group ‘00’H
	FE	01	Multicast to Functional Modules with Multicast Group ‘01’H
	...	...	
	FE	FE	Multicast to Functional Modules with Multicast Group ‘FE’H
Extended Addressing	FF	00	Extended addressing mechanism (reserved)
	...	...	
	FF	FE	Extended addressing mechanism (reserved)
Broadcast	FF	FF	Broadcast to all available functional modules in all components

4 Message frame format

The AMI-C message frame format has the structure shown below in Table 2.

Table 2 — AMI-C message frame format

0 … 9	10	11 12	13 14	15	16 … 23	24 … 31
Message Length	Sys	Rsv	Pri	More	Transaction ID	Multi-frame Sequence ID
AMI-C Message						

4.1 Message length

Messages, within the frame, have a length of 10 bits. The length is expressed in bytes and does not include the first 8 bytes (1st and 2nd quadlets).

4.2 System bit (Sys)

This bit indicates if the message is part of an application transaction (Sys='0'B) or of a system transaction (Sys='1'B).

4.3 Reserved (Rsv)

This field is reserved and not used.

4.4 Priority (Pri)

The Priority field indicates the priority with which a message is sent. A higher priority message should be processed faster than lower priority messages. It is a 2-bit field, which provides 4 levels. Priority "0" is the lowest and the default; Priority and "3" is the highest.

4.5 More bit field (More)

The More bit, when set to '0'B, indicates that no other packet accompanies the current packet. When set to '1'B it indicates that the next packet (with the same Transaction-ID) is part of a multi-packet message. It is used most often for large messages where the multi-frame ID indicates the order of the packets in the message being sent. When a transaction takes only one message, the More bit should be set to '0'B and the Multi-frame sequence ID (see 4.7) should be '0'B.

4.6 Transaction identifier (T-ID)

The Transaction ID identifies messages belonging to the same transaction. For example, the same T-ID value shall be used for a request and its subsequent response. The value of the T-ID is a random value limited by the size of its range (8 bits; therefore, the value is between 1 and 255).

4.7 Multi-frame sequence Id (M-ID)

The Multi-frame sequence ID identifies packets belonging to the same message. If a message is too large to be contained in one packet, this message shall be split in a sequence of smaller packets which order is indicated by the M-ID field. The first packet shall have an M-ID equal to 0. The following packets shall have their M-ID incremented sequentially, the maximum being fixed by the size of the M-ID bit field, i.e. 8 bits = 255. The last packet shall be identified by its More bit set to '0'B.

4.8 AMI-C message

The AMI-C Message area contains messages defined for application or system transactions. The application transaction is to send or receive vehicle interface messages, and the system transaction is for resource manager inquiry, I-Num allocation, or address resolution.

5 Application transaction

5.1 Message format

The system transaction message format is described in Clause 6. The fields are detailed thereafter. The system bit (Sys) shall be set to ‘0’B for application transaction.

Table 3 — Application transaction frame format

0 3 4 5 7 8 10 11 12 13 14 15 16 23 24 31

Message Length				Sys	Rsv	Pri	More	Transaction ID	Multi-frame Sequence ID
Dest F-Type									**Dest** I-Num
Source F-Type									**Source** I-Num
Rsv	Cf	Msg Type	Message Class					Object Property	Operand 0
Operand 1			Operand 2					Operand 3	…
								Zero Pad Bytes (if necessary)	

5.1.1 Source and destination function type (F-Type)

The function type shows the type of a functional module that is the source or receiver; for example, window, mirror, and so on. It is an 24-bit field, defining 15,728,640 (2^{24}–2^{20}) function types. If there is more than one functional module of the same type, I-Num i.

5.1.2 Source and destination instance number (I-Num)

The instance number identifies one of the functional modules to which a source functional module sends messages. It is an 24 to 31 bit field. I-Num ‘FF’h is a broadcast to all functional modules with the same function type. Each I-Num is assigned during the first configuration stage. I-Num is used to uniquely identify one phone in a car when, for example three occupants each have their own Bluetooth phone. Some instance numbers are predefined for certain functional modules. Using windows as an example, driver-side window is ‘01’H, passenger-side window is ‘02’H, and so on.

5.1.3 Destination logical address

The destination logical address consists of the destination F-Type and destination I-Num, and identifies the functional module to which a message is sent.

When the F-Type field is equal to ‘FF’H, it indicates that the packet is broadcast (see 3.2) or Multicast (see 3.3).

5.1.4 Source logical address

The source logical address consists of the source F-Type and source I-Num, and identifies the functional module who originated the current message.

5.1.5 Reserved (Rsv)

These four bits are reserved and shall be set to 0.

5.1.6 Confirmation flag (Cf)

The confirmation flag is set by the application when it needs to know whether the current message is received. Upon reception of a message with the Cf flag set to ‘1’B, the destination logical address will send a CONFIRM message to the source logical address.

5.1.7 Message type (Msg type)

Message Type can have 6 values: INQUIRE, REPORT, SET, CONFIRM, COMMAND, and WARNING.

- **INQUIRE** - A request to inquire the current value of a property at one instant of time from an object residing in an FM.
- **REPORT** - A reply for an ’Inquire” message containing current value of one property in one instant of time.
- **SET** - A request to change the current value of one property in one instant of time.
- **COMMAND** - A message to perform one of the following five kinds of actions: start the execution of software in a remote component, actuate device in a remote component, request access to a resource, start a subscription for a property or end subscription for a property.
- **CONFIRM** - A message to indicate the success or error status of the requested operation when a confirmation message is requested with the Cf bit set to ‘1’B.
- **WARNING** - Notification of status to other nodes (with no requests).

Figure 4 illustrates the relationship between the INQUIRE and REPORT message types.

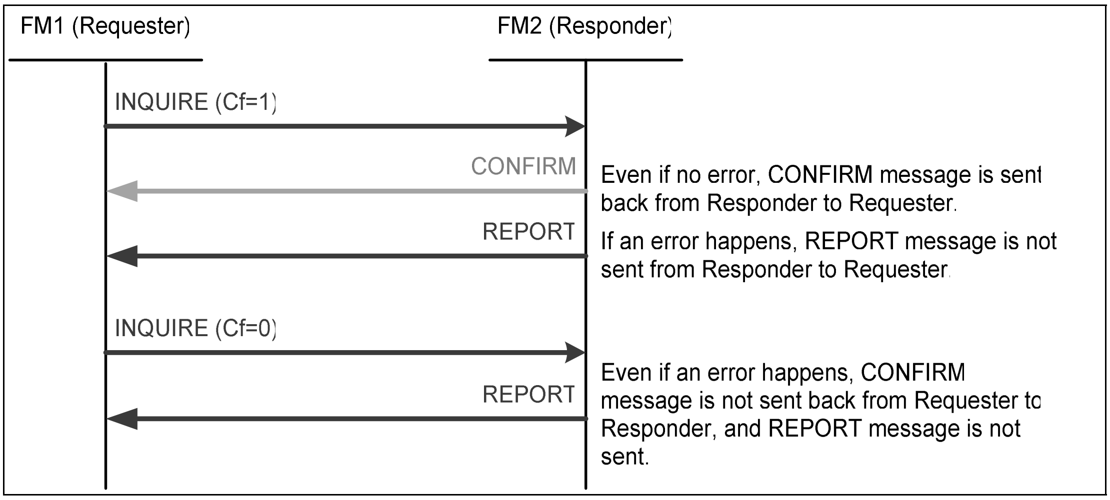

Figure 4 — INQUIRE and REPORT message sequence

Figure 5 illustrates the relationship between the SET and CONFIRM message types.

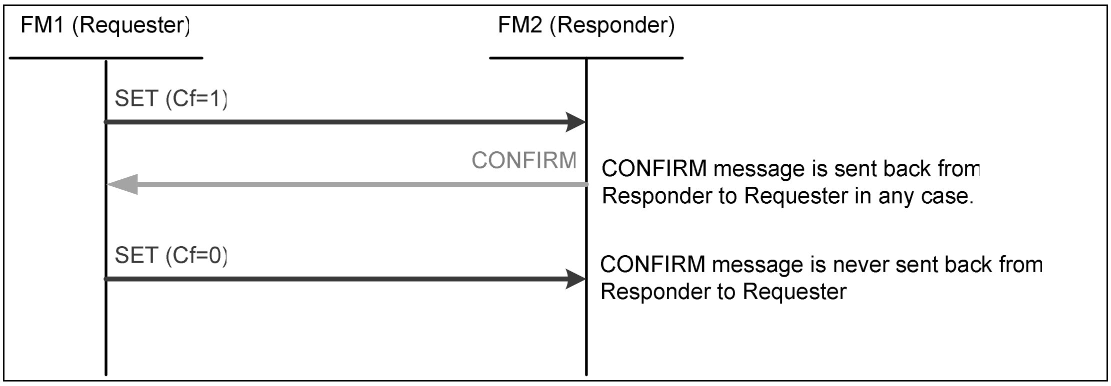

Figure 5 — SET and CONFIRM message sequence

5.1.8 Message class

Message classes are logical groups defined as follows:

- **Management** - network device management, audio/video stream management, and service discovery;
- **Core** - information originally inherent in a vehicle (VIN, static configuration information, etc.);
- **Body module** - control and status related to body module (window, seat, mirror, light, trip meter, vehicle speed, etc.);
- **Power train** - status related to power train (oil temp, coolant temp, gear, etc.);
- **Vehicle diagnostics** - message for vehicle diagnostics (ISO 15031-5 emission related, ISO 14229-1 non-emission related);
- **Text display** - command to display simple texts and input texts.

5.1.9 Object property

This field defines the specific property of the FM. For example, the Mirror FM can have two object properties: vertical position and horizontal position. Refer to AMI-C 2003 AMI-C vehicle interface requirements.

5.1.10 Extension possibility

In order to allow the addition of further object property codes (beyond 254) we define the following expansion method. The restricted value 255 (‘FF’H) indicates that Operand 0 contains the extended object property code.

5.1.11 Operands

These fields depend on the object property and the message type. The value and the format of an operand are defined in AMI-C 2002, AMI-C Common Message Set.

Big Endian format (the most significant byte takes the least significant address) is required when expressing values larger than 8 bits.

There are two types:

1) simple text display and input,

2) XML documents—for more complex transactions (see AMI-C 4002 AMI-C requirements and specifications for Human Machine Interfaces).

6 System transaction

A System Transaction consists of two kinds of messages: one for an initialization mechanism and the other for Address Resolution Protocol (ARP).

6.1 Message format

The system transaction message format of the management packets that are used during the initialization process is shown in Table 4.

Table 4 — System transaction frame format

0 7 8 9 10 11 12 13 14 15 16 23 24 31

Message Length		Sys	Rsv	Pri	More	Transaction ID	Multi-frame Sequence ID
Mgt Command Type	Command Type Specific Information						

System bit (Sys) shall be set to ‘1’B, and Priority (Pri) set to the highest (‘11’B) for the System transaction message frame.

Table 5 — Management command type values

Management command type	Value
Resource Manager Inquiry	‘00’H
Resource Manager Response	‘01’H
Instance Number Allocation Request	‘02’H
Instance Number Allocation Response	‘03’H
Instance Number Deallocation Request	‘04’H
Instance Number Deallocation Response	‘05’H
ARP Request	‘06’H
ARP Response	‘07’H
Suspected RM Failure Warning	‘08’H
Reset I-Num Request	‘09’H

6.1.1 Resource Manager-capable (RMc) inquiry

The following packet is used to inquire whether another RMc FM has already assigned itself as RM. This is a broadcast message.

Table 6 — RMc FM inquiry packet format

0 7	8 15	16 23	24 31
RMC Inquiry	Weight	Rsv	
Network Physical ID			

b

Field descriptions:

- **RMc inquiry** Refer to Table 6 for the Management Command Type Values.
- **Weight** is ‘FF’H (highest priority) for the Main RMc FM of the network and is comprised between 0 (lowest priority) and ‘FE’H for the backup RMc FMs.
- **Network physical ID** This is a unique 32-bit number, which depends on the physical interface with which the functional module communicates. Each network specifies this for itself. The length of this field may differ from one specific network to another; it is not restricted to 32 bits.

6.1.2 RMc response

The following is the response packet of an RMc FM to the RMc FM inquiry. This is a unicast message.

Table 7 — RMc FM response packet format

0 7	8 15	16 23	24 31
RMC Response	Weight	I-Num of Responser	Rsv
Network Physical ID			

Field descriptions:

- **RMc response** Refer to Table 7, for the management command type values.
- **Weight** the weight of the responding FM (‘FF’H for the Main RMc FM of the network and its value is between ‘00’H and ‘FE’H for the Backup RMc FMs).
- **I-Num of responder** the assigned I-Num value of the responder or 0 if unassigned.
- **Network physical ID** a unique 32-bit number, which depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one specific network to another and is not restricted to 32 bits.

6.1.3 I-Num allocation request

This request packet of an FM has an I-Num that is currently unassigned. An FM with a pre-defined I-Num must use this request to register its FM in the RM. This message is broadcast.

Table 8 — Instance number request packet format

0 7	8 15	16 23	24 31
I-Num Alloc Req	F-Type	Request I-Num	I-Num = 00_{16}
Network Physical ID			

Field descriptions:

- **I-Num Request** Refer to Table 8, for the management command type values.
- **F-Type** is the function type value of the requesting FM.
- **Request I-Num** is a randomly assigned value used to differentiate requests coming from 2 FMs that have the same F-Type and belong to the same Component.
- **I-Num equals** '00'H indicates that the I-Num is unassigned.
- **Network physical ID** a unique 32-bit number that depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one specific network to another and is not restricted to 32 bits.

6.1.4 I-Num allocation response

The following is the response packet of the RM to the FM requesting the allocation of an I-Num. This is a unicast message.

Table 9 — Instance number response packet format

0 7	8 15	16 23	24 31
I-Num Alloc Resp	F-Type	Request I-Num	Allocated I-Num
Network Physical ID			

Field descriptions:

- **I-Num allocation** Refer to Table 9, for the management command type values.
- **F-Type** is the function type value of the requesting the FM.
- **Request I-Num** has the same value as in the corresponding request.
- **Allocated I-Num** equals the value assigned by the RM.
- **Network physical ID** a unique 32-bit number that depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one network to another and is not restricted to 32 bits.

6.1.5 I-Num deallocation request

An FM that is going to be disconnected, or an FM that has detected the “dirty” disconnection of another FM shall send the following packet. This message is broadcast message.

Table 10 — Instance number deallocation request packet format

0 7	8 15	16 23	24 31
I-Num Unalloc Req	F-Type	Allocated I-Num	00_{16}

Field descriptions:

- **I-Num deallocation request** Refer to Table 10, for the management command type values.
- **F-Type** is the function type value of the FM that is or will be disconnected.
- **Deallocated I-Num** equals the value requested for deallocation.

6.1.6 I-Num deallocation response

The following packet is the response packet of the RM to the FM requesting the deallocation of an I-Num. This is a unicast message.

Table 11 — Instance number deallocation response packet format

0 7	8 15	16 23	24 31
I-Num Unalloc Resp	F-Type	Allocated I-Num	Unallocated I-Num

Field descriptions:

- **I-Num Deallocation Response** Refer to Table 11, for the management command type value.
- **F-Type** is the function type value of the FM that is or will be disconnected.
- **Deallocated I-Num** equals the value requested for deallocation.

6.1.7 ARP request

The following packet is sent by a component requesting the logical address of a functional module. This is a broadcast message.

Table 12 — ARP request packet format

0 … 7	8 … 15	16 … 23	24 …	… 31
ARP request	Src physical ID length	Dest physical ID length	Src network ID	Dest network ID
Source F-Type	Source I-Num	Dest F-Type	Dest I-Num	
Source network physical ID				
Destination network physical ID				

Field descriptions:

- **ARP request** Refer to Table 12 or the management command type values.
- **Src physical** ID length is the total length in bytes of the source network physical ID fields. In case of 1394 Automotive, the field is set to 4.
- **Dest physical Id length** is the total length in bytes of the destination network physical ID field since the latter information is unknown. This field shall be set to 4 in this case by default.
- **Src network ID** identifies the physical network where the source component is connected. Refer to Table 13 for the network ID values.
- The following table lists the specific networks endorsed by AMI-C and their corresponding codes used for the Network Physical Id fields.

Table 13 — Networks ID values

Network	Value
Bluetooth	0
1394 Automotive	1
Reserved	2
MOST	3
UDP/IP	4
Reserved	...
Unknown	15

- **Dest network ID** identifies the physical network where the destination component is connected. Refer to Table 13 for the network ID values.
- **Source F-Type & I-Num** is the Logical Address of the requesting FM.
- **Destination F-Type & I-Num** together forms the logical address of the found FM.
- **Source network physical ID** depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one network to another and is not restricted to 32 bits. This field shall be quadlet aligned, and pad bytes (value = 0) shall be used when needed.
- **Destination network physical ID** is left empty: 1 quadlet equal to 0.

6.1.8 ARP response

The following packet shall be sent by the component containing the functional module for which an ARP request has been issued. This is a unicast message.

Table 14 — ARP response packet format

0 — 7	8 — 15	16 — 23	24 — 31	
ARP response	Src physical ID length	Dest physical ID length	Src network ID	Dest network ID
Source F-Type	Source I-Num	Dest F-Type	Dest I-Num	
Source network physical ID				
Destination network physical ID				

Field descriptions:

- **ARP response** Refer to Table 14, for the management command type values.
- **Src physical ID** length is the total length in bytes of the source network physical Id fields. In case of 1394 Automotive, the field is set to 4.
- **Dest physical ID** length is the total length in bytes of the destination network physical ID fields. In case of 1394 Automotive, the field is set to 4.
- **Src network ID** identifies the physical network where the source component is connected. Refer to the Table 3 for the network ID values.
- **Dest network ID** identifies the physical network where the destination component is connected. Refer to the Table 13 for the network ID values.
- **Source F-Type & I-Num** together form the logical address of the responding FM.
- **Destination F-Type & I-Num** together form the logical address of the requesting FM.
- **Source network physical** ID depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one network to another and is not restricted to 32 bits. This field shall be quadlet aligned and pad bytes (value = 0) shall be used when needed. This value is the same as in the corresponding ARP request.
- **Destination network physical ID** depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one network to another and is not restricted to 32 bits. This field shall be quadlet aligned and pad bytes (value = 0) shall be used when needed.

6.1.9 Suspected RMc failure warning

If an FM has not been able to receive the I-Num it has requested, it sends the following packet. This packet is broadcast and the Backup RMc FMs that receive it shall check if the RMc works properly before deciding whether a Reset I-Num request must be sent or not. This is a broadcast message.

Table 15 — Suspected RM failure warning packet format

0 … 7	8 … 15	16 … 23	24 … 31
Suspected RM Failure Warning	Reserved		

Field description:

— Suspected RM failure warning Refer to Table 15, for the management command type values.

6.1.10 Reset I-Num request packet

The backup RM sends the following packet when it detects that the actual RM fails to respond to requests. This message is broadcast.

Table 16 — Reset instance numbers request packet format

0 … 7	8 … 15	16 … 23	24 … 31
Reset I-Num Request	Reserved		
Network Physical ID			

Field descriptions:

— **Reset I-Num request** Refer to Table 16, for the management command type values.

— **Network physical ID** This is a unique 32-bit number that depends on the physical interface with which the functional module communicates. Each network specification defines this section accordingly. The length of this field may differ from one network to another and is not restricted to 32 bits.

7 Initialization process

During an initialization process, an I-Num is assigned to each FM.

A functional module (FM) is identified through its logical address (F-Type and I-Num). The F-Type is a fixed value that characterizes the FM (window, door, seat, and so on) and the I-Num is used to differentiate FMs that have the same F-Type on the network. The I-Num is either static, as in the case of mandatory devices in a car (left front door, for example), or dynamic, as in the case of removable devices.

This section presents the process for allocating dynamic I-Nums, and which FM allocates them.

7.1 Resource manager FM

The network shall have a particular FM called resource manager, referred to as RM in the rest of this document. There shall be a minimum of one RM per AMI-C network. A contention mechanism prevents the problem of having two active RMs at the same time, ensuring the consistency of allocated I-Nums through an AMI-C network.

The roles of the resource manager FM are the following:

— Provide dynamic I-Nums to the FMs that request them when a new component joins the network.

— Deallocate the I-Nums of the FMs that belong to a component that unplugs from the network.

— Allocate the multicast group ID (see 3.3) when requested.

The function type of a resource manager capable functional module is defined in AMI-C 2003 AMI-C vehicle interface requirements.

7.1.1 Network configuration process

An FM that contains the ability to perform the operations listed above is called resource manager-capable (RMc)

Whenever an RMc FM wishes to become the active RM of an AMI-C Network it shall proceed as described in Figure 6 and the list that follows it.

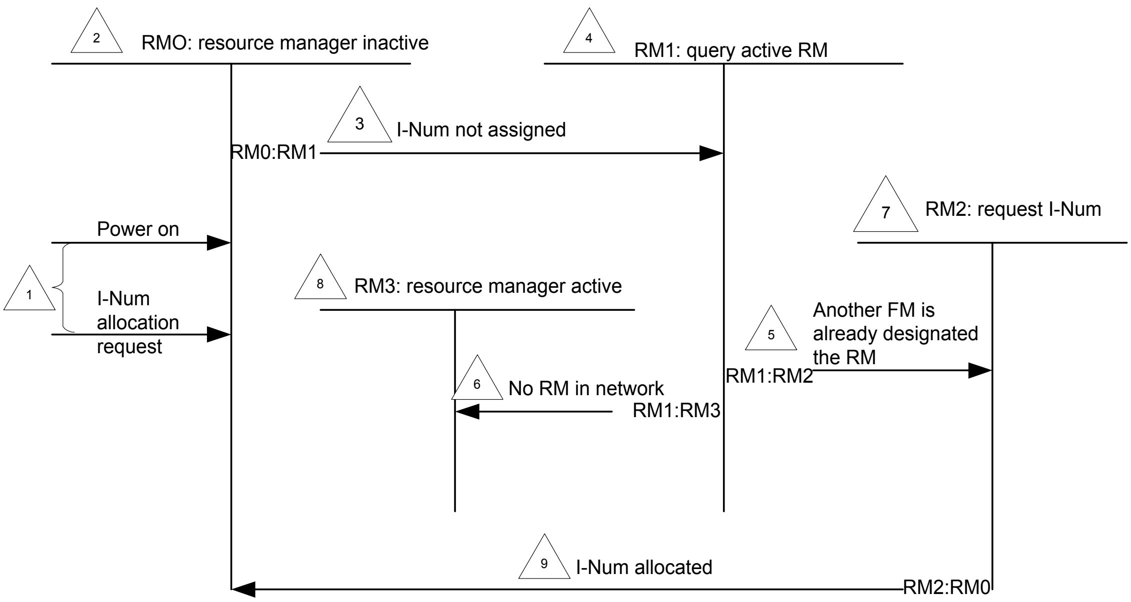

Figure 6 — Resource manager initialization state machine

1) **Transition All: RM0.** Whenever an I-Num allocation request packet is received the candidate RMc FM shall clear its I-Num and restart this state machine in state RM0.

2) **State RM0: Resource manager inactive**. The candidate RMc FM is inactive either as a result of power-on reset or an I-Num allocation reset request received from another RMc FM.

3) **Transition RM0: RM1.** This assignment can be made if the FM has not been assigned any Instance Number.

4) **State RM1: Query active RM.** The candidate RMc FM shall send a broadcast request in order to know if another RMc FM on the network has already been assigned as Resource Manager (see packet format).

5) **Transition RM1: RM2.** Another RMc FM has replied to the previous request (in State RM1) with a higher weight than the candidate RMc FM.

6) **Transition RM1: RM3.** No other RMc FM has replied to the previous request (in #4, State RM1) after the time-out. No other RMc FM has replied with a higher weight to the previous request.

7) **State RM2: Request I-Num.** The RMc FM requests an Instance Number and has lost the Resource Management contention.

8) **State RM3: Resource manager active.** The candidate RMc FM has won the RM contention. Its Instance Number becomes ‘01’H. It shall be responsible for allocating the I-Nums to the other FMs and the Multicast resources when requested.

9) **Transition RM2:RM0.** When the RMc FM has received its instance number from the resource manager, it shall act as any other FM on the network.

Table 17 lists the different values used for the time-out requests to the main resource manager and the backup RMc’s.

Table 17 — Time-outs for candidate RMc inquiry

Attempt	Main RMc time-outs (ms)	Backup RMc time-outs (ms)
1	100	5000

The format of the resource manager capable inquiry request and response packets are defined in 6.1.1 and 6.1.2.

7.1.2 Contention process

When an RM powers up it shall send a broadcast packet to inquire whether another RMc FM already is performing the role of RM. This packet contains a priority that prevents RM contention problems. This priority may be predefined but the Main RM always has the highest priority (‘FF’H).

Each RMc FM that receives such an inquiry shall respond with its own priority, Instance Number and Physical Identification.

When an RMc FM sees only responses with a lower priority within the period before time-out, it shall assume the role of resource manager and assign itself I-Num ‘01’H.

If two RMc FMs send this inquiry at the same time, the RMc FM with the higher priority prevails. This method prevents several RMc FMs from assuming the role of Resource Manager at the same time. If the self-assigned RM receives an inquiry from another RMc FM, it replies with its I-Num equal to ‘01’H.

7.2 Instance numbers allocation process

A functional module that requires a dynamic instance number shall request it from the resource manager FM as described in Figure 7.

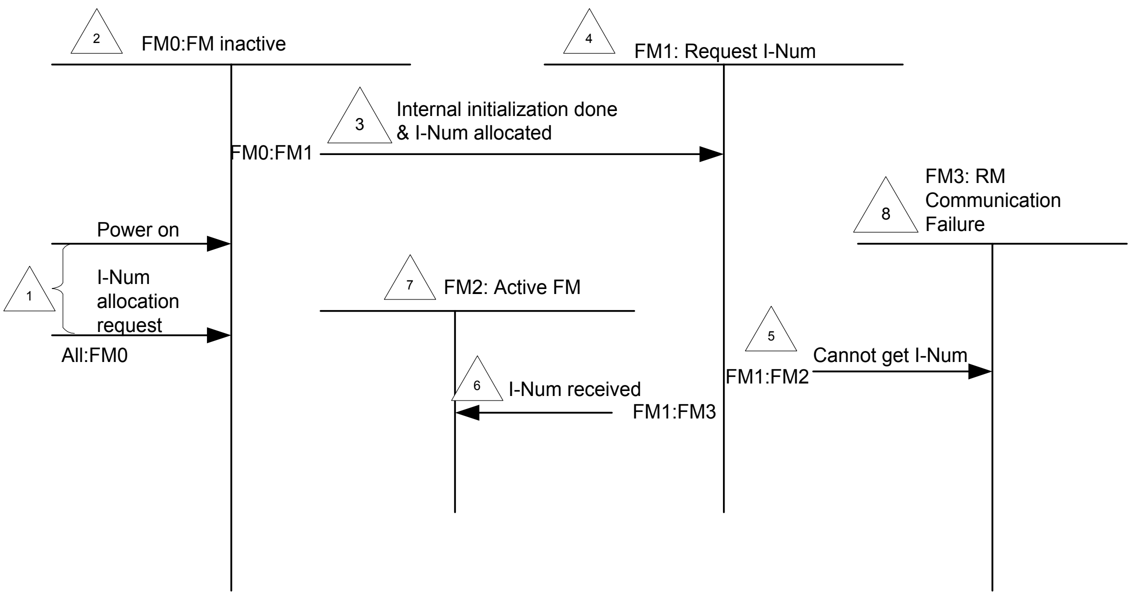

Figure 7 — Dynamic I-Num allocation

1) **Transition All: FM0.** Whenever an I-Num Allocation Request packet is received the FM shall clear its I-Num and restart this state machine in state FM0.

2) **State FM0:** FM Inactive. The Functional Module performs its internal initializations (system level, physical bus level).

3) **Transition FM0:FM1.** The FM has finished its internal initialization and goes to state FM1 if it does not have an Instance Number (static I-Num).

4) **State FM1: Request I-Num.** The FM sends an I-Num request to the Resource Manager.

5) **Transition FM1: FM2.** The FM has received a response from the Resource Manager and has been allocated an I-Num.

6) **Transition FM1: FM3**. The FM made up to five request attempts and the Resource Manager failed to respond within the timeout period.

7) **State FM2: Active FM.** The FM has been allocated an I-Num and may begin to communicate with the rest of the network FMs.

8) **State FM3: RM communication failure.** If the FM can to communicate on the network, it shall send a broadcast message to notify the backup RMs when the current RM is not responding to I-Num allocation requests, (see 7.3). The application shall be notified of the I-Num allocation failure and follow its own timeout rules.

Table 18 — Incremental time-outs for initialization requests attempts

Attempt	Time-out (ms)
1	100
2	200
3	400
4	800
5	1600

The format of the FM I-Num allocation request and response packets is defined in 6.1.

7.3 Recovery process

If the resource manager fails to respond to requests from an FM (after the incremental time-outs), the requesting FM shall send a broadcast message to notify the backup RMs that the current RM is not responding.

The backup RMs shall attempt to communicate with the RM by sending a test I-Num allocation request. If the RM fails to respond correctly, then the backup RMs shall restart the I-Num allocation process by sending a Reset I-Num Request. Upon reception of this packet, the FMs shall pause for a reset delay as defined in Table 19 and then restart the I-Num Allocation process before resuming any other communication.

The failed RM will then be replaced by the backup RM with the next-highest priority.

Table 19 — General configuration timing values

Variable	Value
reset_delay	1 s

8 Address resolution process

When a requesting FM sends a message, it needs to know the physical address of the responder component, including the responder FM. A component may have a cache memory (ARP cache) that is a matching table between a logical address and its physical address. When the requester component does not receive the physical address of the responder component, the physical address shall be determined by an address resolution mechanism (ARP mechanism).

When a request is issued from a FM, the destination FM logical address is indicated in the message header. The ARP mechanism checks its ARP cache to confirm that a physical address matches that logical address.

— If a physical address matches the logical address, the ARP mechanism encapsulates the message for the component; then uses the physical address to define the destination component; and, last, sends the packet.

— If no specific physical address matches the logical address, the ARP mechanism shall update the ARP cache. It shall send a system request to find the physical address of the component that contains the destination FM. An ARP mechanism in the queried component shall respond with the requested information.

8.1 ARP mechanism

An ARP mechanism may be composed of several transactions if the ARP cache does not contain the necessary information shows an address resolution sequence.

1) A Requester FM attempts to send a request to a Responder FM.

2) The ARP cache of the RM FM does not contain the physical address of the component that contains the responder FM.

3) The requester component sends a broadcast request concerning the responder FM, providing information concerning the requester FM at the same time.

4) The ARP mechanism that holds the responder FM may update its own ARP cache concerning the Requester FM. Other components on the network may also do it since they received the information.

5) The responder component's ARP mechanism sends a response to the requester component, providing the physical address of the responder component.

6) The requester component registers the information concerning the responder FM.

7) The requester component sends the original message to the responder FM, specifying the physical address of the responder component in the physical header.

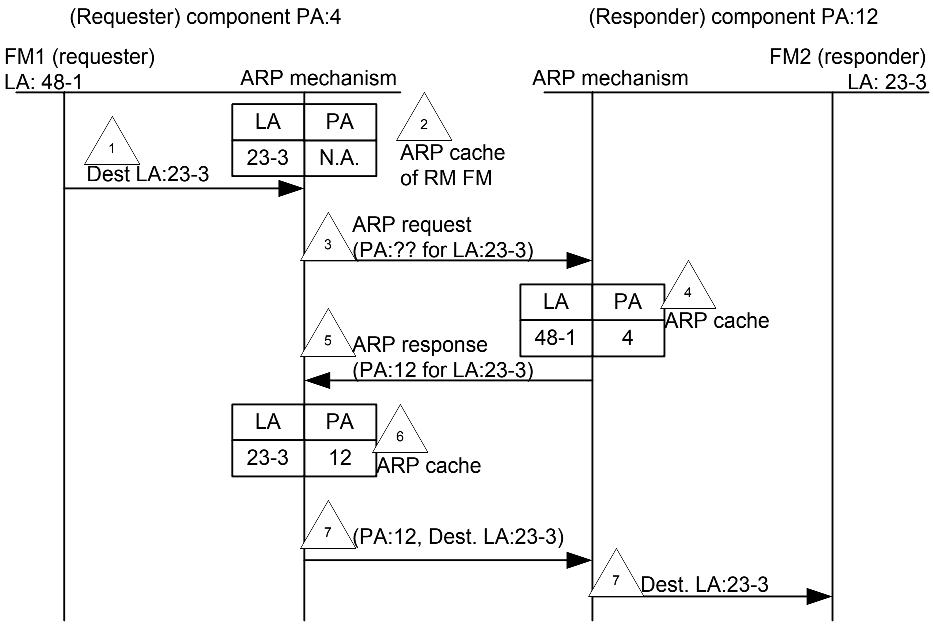

Figure 8 — Address resolution sequence

8.2 No responder error handling

In the case that the responder component does not respond, the process will fail at step 4. The ARP mechanism of the requester component shall continue to attempt the transmission to the destination FM for the time-out periods defined in Table 20.

After five attempts, the ARP mechanism shall respond to the application that the transaction FM was not completed.

Table 20 — Incremental time-outs for FM localization attempts

Attempt	Time-out (ms)
1	100
2	200
3	400
4	800
5	1600

9 FM discovery / removal process

A network may gain and lose devices during its lifetime.

The following section presents the methods for discovering and removing devices.

9.1 Dynamic I-Num allocation when new component plugs in

When a new component joins the network, its FMs shall send their I-Num allocation requests the same way they would do during the initialization process.

9.2 Dynamic I-Num deallocation when component un-plugs

There are two cases to consider because a component may not have the time to indicate that it is being disconnected from the network.

— "**Clean" disconnection** A component receives a notice from the application that it is going to be disconnected. It then sends information in a System Message to indicate that its resources (the I-Nums of its FMs, the multicast group ID it had been allocated) are to be freed. This method allows other components to monitor this event. The component responds with a unicast message to the disconnecting node. The disconnecting component may wait for the response.

— **"Dirty" disconnection** The RM FM detects that another component has disconnected. The component shall send a system message in order to inform the master about the disconnection without specifying which component has disconnected. The method the RM FM uses to resolve this situation is beyond the scope of this specification.

9.3 Multicast resource allocation

When a component receives a subscription request from a requester component, it shall allocate a multicast group ID the following way:

— Broadcast a system message asking the aster (that is, the RM FM) to allocate a group ID.

— The master then provides a group ID (the range of the value is limited by the field of the I-Num" 8 bits, therefore 255 values, see Logical addressing) via a unicast message.

— The component shall then send a subscription response to the requester component indicating the value of the multicast group ID.

10 Service discovery

10.1 Protocol identification

Each network technology has its own mechanism for identifying which upper layer services are supported over a network's transport layer, such as configuration ROM for 1394 network, SDP for Bluetooth, and port number for TCP/IP. Therefore, a specific identifier needs to be defined for each network.

10.2 Service identification

Service identification and its discovery is realized by message transportation over VIP. This mechanism is independent of protocol identification.

10.3 Service discovery messages

The service discovery mechanism identifies and distinguishes general service information and specific service information.

General service information is identified F-Type, which stands for function type. The master registry table (the RM FM) keeps general service information for all functional modules in the network.

Each functional module keeps its own specific service information. When an application tries to find other services, first the application sends to the RM FM a request for the module with the general service. The RM FM sends a response to the requesting application. After this, the application sends a request to the module to be notified by the RM FM to request its specific service information.

The master registry table may or may not keep specific service information for all functional modules in the network when memory size is limited.

Annex A
(informative)

Registry table

Each component has a registry table: a database for relating AMI-C network addresses to addresses recognized by the network connectivity technology used (here, “specific network”)—for example, 1394 Automotive, Bluetooth, an automaker’s proprietary network, etc.

The registry table consists of the AMI-C network definition (F-Type, I-Num, and component ID), the specific network definition (network ID, physical device ID, and logical device ID), and service information (module status and so on).

The network ID portion of the “specific network” definition is necessary for a gateway to identify where the component, including a certain module, is connected. A network ID is assigned to a component by the “specific network” and is not necessarily a unique number with respect to the AMI-C network. When a component works on two network connectivity technologies—say, both the 1394 Automotive and an automaker’s proprietary network—the component needs to serve as a gateway, identifying its devices to both technologies. An application does not need to know which connectivity technology a functional module belongs to. The network ID is not necessary if a component is a gateway for more than two connectivity technologies.

An example of the contents of a registry table is shown in Table A.1.

Table A.1 — Registry table example

AMI-C Network Definition			Specific Network Definition			Service Information
F-Type	I-Num	Component ID	Network ID (Gateway)	Physical Device ID	Logical Device ID	Module Status
F-Type	I-Num	Component ID	Network ID (Gateway)	Physical Device ID	Logical Device ID	Module Status
F-Type	I-Num	Component ID	Network ID (Gateway)	Physical Device ID	Logical Device ID	Module Status

Suppose one of the vehicle interfaces with a component ID of 0 has the master registry table. The RM FM (component management functional module) in the vehicle interface assigns the component ID and I-Num for other modules and updates the registry table. The master registry table contains information about all functional modules the network.

Other components may have all information about other modules in their registry tables, but this is not a requirement. A component’s table may update its information whenever it receives update information broadcasted whenever the master registry table is updated. Another option is for a component to load the information it requires from the master registry table and cache in its own table the access of this component with other functional modules.

When a component leaves the network or powers down—or a functional module is shut down—the master registry table is updated and broadcasts a notification message to each component (functional module). Individual tables shall be updated when they contain an entry for the now-unavailable functional module.

ICS 43.040.15

Price based on 24 pages