System Specifications

3

Communication

3

Application Layer

7

Summary

This document specifies the Application Layer of the KNX System.

Version 01.06.02 is a KNX Approved Standard.

This document is part of the KNX Specifications v2.1.

Document updates

Version	Date	Modifications
1.0	2001.12.19	Preparation of the Approved Standard.
1.0	2005.08.22	AN045 → 3.3.2 “A_DeviceDescriptor_InfoReport-service”
1.0	2007.01.08	S15 “Easy Common Parts” – A_Link_Read and A_Link_Write integrated.
1.0	2007.01.12	• AN038 “Function Services” integrated.
1.0	2007.01.24	• AN028 “A_NetworkParameter_Write” integrated.
1.0	2007.08.21	• AN022 “A_PropertyDescription_Read-service” integrated.
1.0	2007.08.21	• AN026 “DoA on RF” integrated. • Introducing of grouping of AL-services in “System Broadcast”
1.0	2007.09.19	• AN026 “Connectionless A_Restart” integrated.
1.0	2007.09.20	• AN065 “Number field in memory services” integrated.
1.0	2008.01.17	• AN044 “RF specification complements” integrated
1.0	2008.05.05	• AN044 “APCI_ServiceInformation_Indication_Write” integrated: service A_ServiceInformation_Indication_Write-Service removed.
1.0	2008.05.07	• AN072 “A_Authorize correction and configuration” integrated.
1.0	2008.05.14	• AN046 “Device Descriptor error handling” integrated.
1.0	2008.05.23	• Update of A_Link_Write handling according [KSG00105-14]
1.0	2008.08.12	• AN101 “File Transfer Protocol” integrated.
1.1	2008.12.24	Finalisation of the Approved Standard v1.1.
	2009.02.24	Indication about variable size of A_Link_Response-PDU ([Ref20090224ab])
1.2.00	2009.11.05	• AN116 “connectionless use of manufacturer specific user messages” integrated.
1.2.01	2010.08.02	• AN124 “Interface Object Index Discovery” integrated.
1.2.02	2010.09.27	Corrected reference to definition of TMedia from [06] to [05] in 3.2.6. Checked all instances.
1.3.00	2010.10.22	• AN127 “Master Reset” integrated.
01.04.00	2012.09.10	• Revision of the integration of AN127 “Master Reset” according the updated AN127 “Master Reset” v05. • Editorial update
01.04.01	2013.07.11	• A_Authorize-service: corrected error handling in case of missing authorisation. • A_UserManufacturerInfo_Read not recommended for new stack implementations.
01.05.00	2013.07.22	• AN132 “A_NetworkParameter_InfoReport” integrated.
01.05.01	2013.09.10	• Revision of the integration of AN132.
01.05.02	2013.09.27	• Editorial update to include APCI_FileStream_InfoReport in Table 1.
01.06.00	2013.10.23	• AN162 “System aspects of RF S-Mode” integrated.
01.06.01	2013.10.28	Editorial updates for the publication of KNX Specifications 2.1.
01.06.02	2013.12.10	Vertical alignment of cover page corrected. Final editorial review in view of publication of the KNX Specifications v2.1.

References

[01] Chapter 3/3/3 “Network Layer”
[02] Chapter 3/3/4 “Transport Layer”
[03] Chapter 3/4/1 “Application Interface Layer”
[04] Chapter 3/5/1 “Resources”
[05] Chapter 3/5/2 “Management Procedures”
[06] Volume 6 “Profiles”

Filename: 03_03_07 Application Layer v01.06.02 AS.docx
Version: 01.06.02
Status: Approved Standard
Savedate: 2013.12.11
Number of pages: 93

Contents

1 Introduction

1.1 Application Layer services and Transport Layer communication modes

The Application Layer provides a large variety of application services to the application process. Application processes in different devices interoperate by using services of Application Layer over communication modes. According to Transport Layer, different types of communication modes exist:

- point-to-multi-point, connection-less (multicast)
- point-to-all-points, connection-less (broadcast)
- point-to-point connection-less
- point-to-point connection-oriented.

Depending on the communication mode, different Application Layer services are offered.

Some services can be used on the point-to-point connection-oriented, as well as the point-to-point connection-less communication mode, although Application Layer services are always mapped to Transport Layer services depending on the type of the communication mode.

1.2 Service Primitives of the Application Layer

Each specified Application Layer service shall be invoked by the Transport Layer primitives Request(req), Indication(ind) and confirm(con).For a remote confirmed Service the remote device shall use the same Transport Layer primitives to respond to the service.

The Transport Layer confirm primitive shall only be a confirmation from the Transport Layer instance and shall include all data from the request plus the state that shall indicate whether the service is sent successfully or not. The Application Layer shall map the Transport Layer confirm to a Local confirm (Lcon).

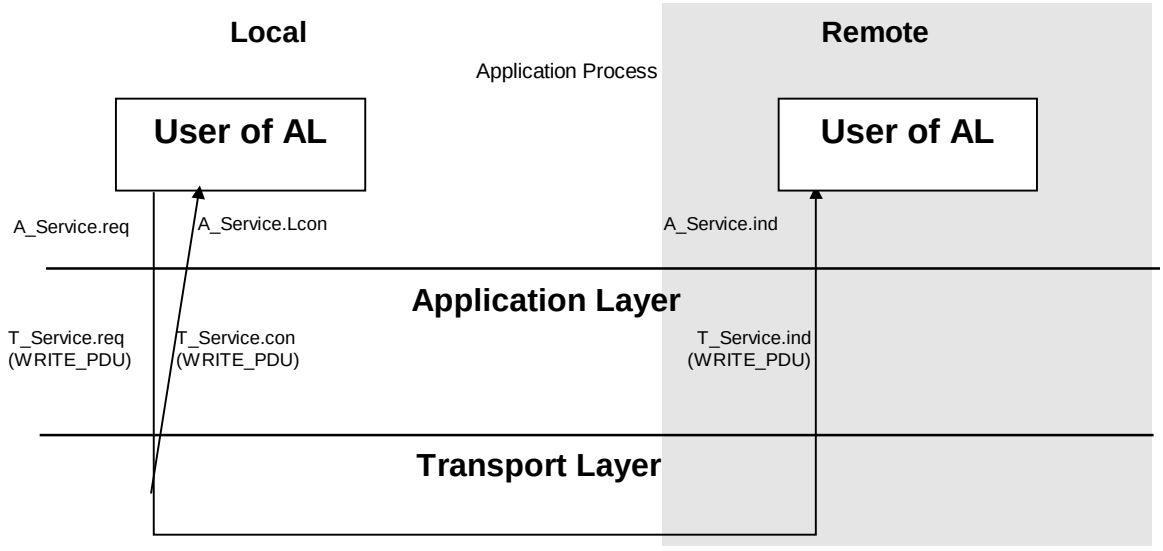


Figure 1 - Interactivity of the Application Layer for services that are not remote confirmed

In case of a remote confirmed service the remote device shall initiate the Response(res) primitive and the Application layer shall map this service to a Transport Layer request primitive. The local Application Layer shall receive the TL indication primitive and shall map it to an Application Layer Confirm(Acon). The Transport Layer confirmation in the remote device shall be mapped by the Application Layer to a Response Confirm (Rcon). In the following service descriptions the Local Confirm(Lcon) and the Response Confirm (Rcon) are not described.

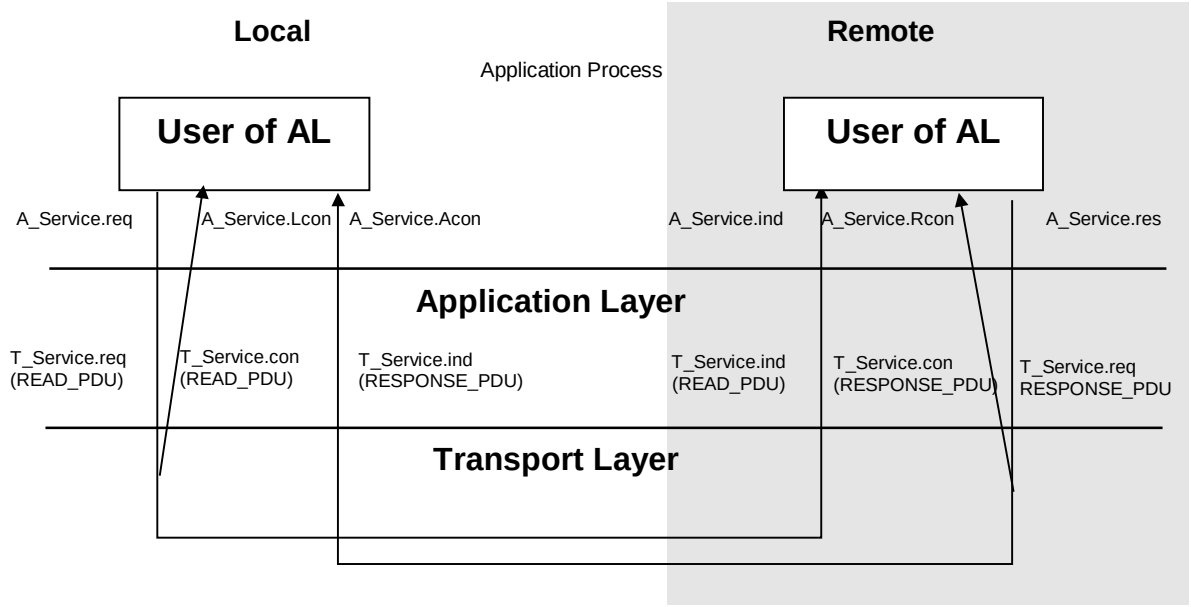


Figure 2 - Interactivity of the Application Layer for services that are remote confirmed

2 APDU

The APDU is shown in Figure 3 below.

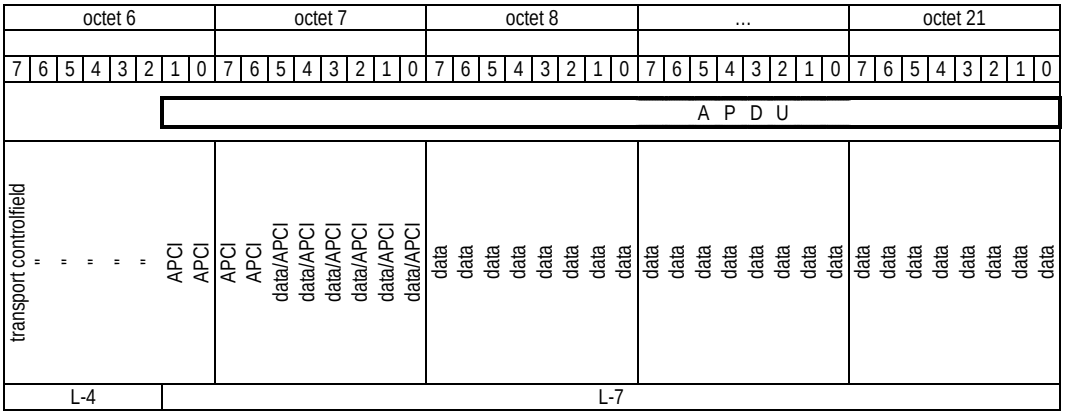


Figure 3 - Format of the APDU (example)

The APDU shall correspond to the TPDU, but shall be reduced by the Transport Control field. The application control field shall be encoded and decoded by Application Layer and shall contain the Application Layer service codes. The Application Control field shall have a length of 4 or 10 bits, depending on the Application Layer service. The codes for the application control field are shown in Table 1.

The complete PDU for each service primitive is shown in the description of every service. Not defined and not supported Application Layer services shall be ignored by the Application Layer.

The symbols in Table 1 shall be interpreted as follows. Wether or not an Application Layer services is mandatory, is specified in [06] either explicitly or through the requirements of other features than Application Layer services. If for a certain Profile is is required that an Application Layer service be implemented, then additionally Table 1 applies, with the following definition.

- If the service primitive is marked with "M" then this service primitive shall be implemented for this communication mode.
- If the service primitive is marked with "O" then this service primitive may be implemented for this communication mode.
- If the service primitive is marked with "X" then this service primitive shall not be implemented for this communication mode.

Table 1 – Application Layer control field

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		Communication modes: multicast	broadcast	system broadcast	point-to-point connectionless	point-to-point connection-orioented
						APCI	APCI	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat						
						0	0	0	0	0	0	0	0	0	0	A_GroupValue_Read-PDU	M	x	x	x	x
						0	0	0	1							A_GroupValue_Response-PDU	M	x	x	x	x
						0	0	1	0							A_GroupValue_Write-PDU	M	x	x	x	x
						0	0	1	1	0	0	0	0	0	0	A_IndividualAddress_Write-PDU	x	M	x	x	x
						0	1	0	0	0	0	0	0	0	0	A_IndividualAddress_Read-PDU	x	M	x	x	x
						0	1	0	1	0	0	0	0	0	0	A_IndividualAddress_Response-PDU	x	M	x	x	x
						0	1	1	0							A_ADC_Read-PDU	x	x	x	x	M
						0	1	1	1							A_ADC_Response-PDU	x	x	x	x	M
						0	1	1	1	0	0	1	0	0	0	A_SystemNetworkParameter_Read-PDU	x	x	M	x	x
						0	1	1	1	0	0	1	0	0	1	A_SystemNetworkParameter_Response-PDU	x	x	M	x	x
						0	1	1	1	0	0	1	0	1	0	A_SystemNetworkParameter_Write-PDU	x	x	M	x	x
						0	1	1	1	0	0	1	0	1	1	planned for future system broadcast service	x	x	M	x	x
						1	0	0	0	0	0					A_Memory_Read-PDU	x	x	x	O	M
						1	0	0	1	0	0					A_Memory_Response-PDU	x	x	x	O	M
						1	0	1	0	0	0					A_Memory_Write-PDU	x	x	x	O	M
						1	0	1	1	0	0	0	0	0	0	A_UserMemory_Read-PDU	x	x	x	x	M
						1	0	1	1	0	0	0	0	0	1	A_UserMemory_Response-PDU	x	x	x	x	M
						1	0	1	1	0	0	0	0	1	0	A_UserMemory_Write-PDU	x	x	x	x	M
						1	0	1	1	0	0	0	1	0	0	A_UserMemoryBit_Write-PDU (not for future use)	x	x	x	x	M
						1	0	1	1	0	0	0	1	0	1	A_UserManufacturerInfo_Read-PDU	x	x	x	x	M
						1	0	1	1	0	0	0	1	1	0	A_UserManufacturerInfo_Response-PDU	x	x	x	x	M
						1	0	1	1	0	0	0	1	1	1	A_FunctionPropertyCommand-PDU	x	x	x	M	M
						1	0	1	1	0	0	1	0	0	0	A_FunctionPropertyState_Read-PDU	x	x	x	M	M
						1	0	1	1	0	0	1	0	0	1	A_FunctionPropertyState_Response-PDU	x	x	x	M	M
						1	0	1	1	0	0	1	0	1	0		x	x	x	x	M
						...	...									... Reserved USERMSG					
						1	0	1	1	1	1	0	1	1	1		x	x	x	x	M
						1	0	1	1	1	1	1	0	0	0		x	x	x	O	M
						...	...									... manufacturer specific area for USERMSG					
						1	0	1	1	1	1	1	1	1	0		x	x	x	O	M
						1	1	0	0	0	0	0	0	0	0	A_DeviceDescriptor_Read-PDU	x	x	x	O	M
						1	1	0	1	0	0	0	0	0	0	A_DeviceDescriptor_Response-PDU	x	x	x	O	M
						1	1	1	0	0	0	0	0	0	0	A_Restart-PDU	x	x	x	O	M
Coupler specific services																					
						1	1	1	1	0	0	0	0	0	0	A_Open_Routing_Table_Req-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	0	0	0	1	A_Read_Routing_Table_Req-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	0	0	1	0	A_Read_Routing_Table_Res-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	0	0	1	1	A_Write_Routing_Table_Req-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	1	0	0	0	A_Read_Router_Memory_Req-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	1	0	0	1	A_Read_Router_Memory_Res-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	1	0	1	0	A_Write_Router_Memory_Req-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	1	1	0	1	A_Read_Router_Status_Req-PDU (not for future use)	x	x	x	x	M
						1	1	1	1	0	0	1	1	1	0	A_Read_Router_Status_Res-PDU (not for future use)	x	x	x	x	M

Table 1 – Application Layer control field

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			**Communication modes**				
						APCI	APCI	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat	APCI/dat			multicast	broadcast	system broadcast	point-to-point connectionless	point-to-point connection-orioented
						1	1	1	1	0	0	1	1	1	1	A_Write_Router_Status_Req-PDU	(not for future use)	x	x	x	x	M
						1	1	1	1	0	1	0	0	0	0	A_MemoryBit_Write-PDU	(not for future use)	x	x	x	x	M
						1	1	1	1	0	1	0	0	0	1	A_Authorize_Request-PDU		x	x	x	x	M
						1	1	1	1	0	1	0	0	1	0	A_Authorize_Response-PDU		x	x	x	x	M
						1	1	1	1	0	1	0	0	1	1	A_Key_Write-PDU		x	x	x	x	M
						1	1	1	1	0	1	0	1	0	0	A_Key_Response-PDU		x	x	x	x	M
						1	1	1	1	0	1	0	1	0	1	A_PropertyValue_Read-PDU		x	x	x	M	M
						1	1	1	1	0	1	0	1	1	0	A_PropertyValue_Response-PDU		x	x	x	M	M
						1	1	1	1	0	1	0	1	1	1	A_PropertyValue_Write-PDU		x	x	x	M	M
						1	1	1	1	0	1	1	0	0	0	A_PropertyDescription_Read-PDU		x	x	x	M	M
						1	1	1	1	0	1	1	0	0	1	A_PropertyDescription_Response-PDU		x	x	x	M	M
						1	1	1	1	0	1	1	0	1	0	A_NetworkParameter_Read-PDU		x	M	x	x	x
						1	1	1	1	0	1	1	0	1	1	A_NetworkParameter_Response-PDU		x	M	x	M	x
						1	1	1	1	0	1	1	1	0	0	A_IndividualAddressSerialNumber_Read-PDU		x	M	x	x	x
						1	1	1	1	0	1	1	1	0	1	A_IndividualAddressSerialNumber_Response-PDU		x	M	x	x	x
						1	1	1	1	0	1	1	1	1	0	A_IndividualAddressSerialNumber_Write-PDU		x	M	x	x	x
						1	1	1	1	0	1	1	1	1	1	reserved [1]	(not for future use)	x	M	x	x	x
Open media specific services																						
						1	1	1	1	1	0	0	0	0	0	A_DomainAddress_Write-PDU		x	M	M	x	x
						1	1	1	1	1	0	0	0	0	1	A_DomainAddress_Read-PDU		x	M	M	x	x
						1	1	1	1	1	0	0	0	1	0	A_DomainAddress_Response-PDU		x	M	M	x	x
						1	1	1	1	1	0	0	0	1	1	A_DomainAddressSelective_Read-PDU		x	M	M	x	x
						1	1	1	1	1	0	0	1	0	0	A_NetworkParameter_Write-PDU		x	M	x	M	x
						1	1	1	1	1	0	0	1	0	1	A_Link_Read-PDU		x	x	x	M	M
						1	1	1	1	1	0	0	1	1	0	A_Link_Response-PDU		x	x	x	M	M
						1	1	1	1	1	0	0	1	1	1	A_Link_Write-PDU		x	x	x	M	M
						1	1	1	1	1	0	1	0	0	0	A_GroupPropValue_Read-PDU		M	x	x	x	x
						1	1	1	1	1	0	1	0	0	1	A_GroupPropValue_Response-PDU		M	x	x	x	x
						1	1	1	1	1	0	1	0	1	0	A_GroupPropValue_Write-PDU		M	x	x	x	x
						1	1	1	1	1	0	1	0	1	1	A_GroupPropValue_InfoReport-PDU		M	x	x	x	x
						1	1	1	1	1	0	1	1	0	0	A_DomainAddressSerialNumber_Read-PDU		x	M	M	x	x
						1	1	1	1	1	0	1	1	0	1	A_DomainAddressSerialNumber_Response-PDU		x	M	M	x	x
						1	1	1	1	1	0	1	1	1	0	A_DomainAddressSerialNumber_Write-PDU		x	M	M	x	x
						1	1	1	1	1	1	0	0	0	0	A_FileStream_InforReport-PDU		x	x	x	M	M

[1] This APCI-value 3DFh has been used in the past for the service A_ServiceInformation_Indication. This APCI shall not be used for new implementations.

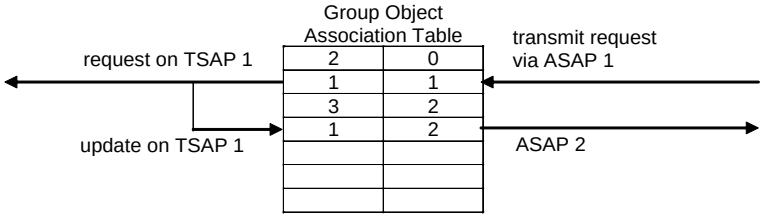
3 Application Layer services

3.1 Application Layer services on Multicast Communication Mode

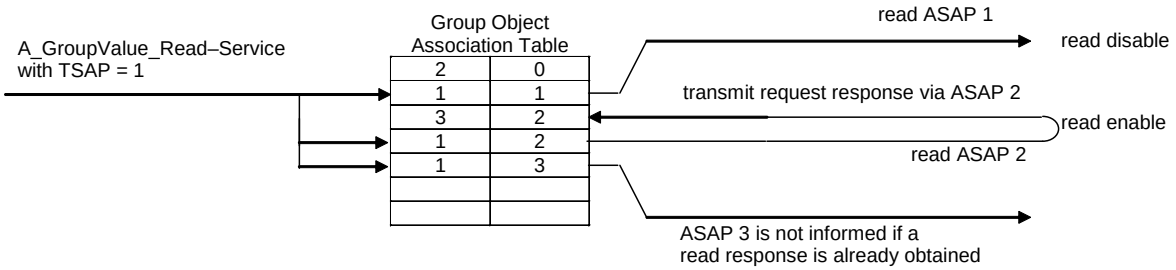
3.1.1 The relation between TSAPs and ASAPs

A multicast communication mode shall connect Transport Layer Service Access Points (TSAP) to Application Layer Service Access Points (ASAP). If one device sends an A_GroupValue_Service each device that is member of this group shall receive the A_GroupValue_Service.

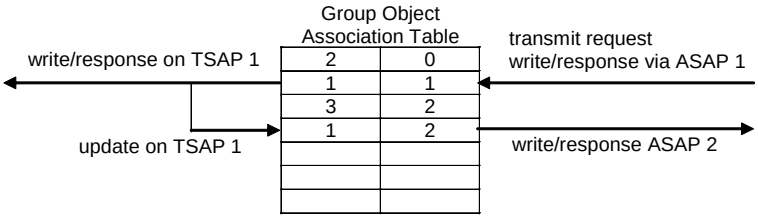
If the Application Layer of a device receives an A_GroupValue_Write-Service, it shall search the TSAP in all entries of the Group Object Association Table and informs all the associated ASAP.



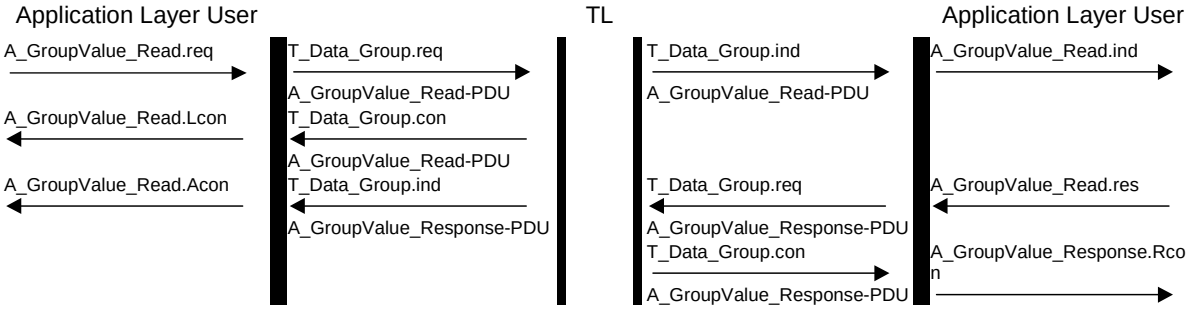
If the Application Layer of a device receives an A_GroupValue_Read-service, it shall search the TSAP in all entries of the Group Object Association Table and shall inform all the associated ASAPs. Only one read response shall be generated by the user.



If a transmission is requested (read response or write) via an ASAP, the Application Layer shall take the TSAP from the Group Object Association Table, shall update all the ASAPs with the same TSAP and shall generate an A_GroupValue-Service-Request.



3.1.2 A_GroupValue_Read-service



The A_GroupValue_Read.req primitive shall be applied by the user of Application Layer, to receive an update of the value of its ASAP by making a communication partner respond with an A_GroupValue_-Read.res, i.e. the service shall be confirmed by the remote application process. The ASAP shall be associated to the TSAP via the Group Object Association Table, i.e. with a Group Address (see Transport Layer). All other group members shall receive the A_GroupValue_Response-PDU as well.

The local Application Layer shall accept the service request, map the ASAP to the TSAP and pass it with a T_Data_Group.req to the local Transport Layer. The user shall decide during configuration about this mapping. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Group.req primitive; the TSDU shall be an A_GroupValue_Read-PDU.

The remote Application Layer shall map a T_Data_Group.ind primitive with TSDU = A_GroupValue_-Read-PDU to an A_GroupValue_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_GroupValue_Read.ind primitive. One A_GroupValue_Read.ind primitive shall be generated per ASAP that is assigned to the corresponding TSAP (i.e. Group Address).

The application process may respond to the A_GroupValue_Read.ind primitive with an A_Group-Value_Read.res primitive containing the value of the ASAP. The user can decide during configuration, whether or not the A_GroupValue_Read.res primitive shall be generated, although it makes sense that exactly one ASAP generates the A_GroupValue_Read.res primitive.

NOTE 1 It is left to the user application programmer to decide whether an A_GroupValue_Read.Acon time-out supervision is necessary.

Two different formats of the A_GroupValue_Response-PDU are used depending on the length of the value. The maximum length of the value is 14 octets. Unused data bits shall be set to zero.

octet 6								octet 7							
						APCI									
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI						
						0	0	0	0						

Figure 4 - A_GroupValue_Read-PDU (example)

octet 6								octet 7								octet 8 to octet 21							
						APCI										data (up to 14 octets)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI														
						0	0	0	1	0	0	0	0	0	0	D	D	D	D	D	D	D	D

Figure 5 - A_GroupValue_Response-PDU, length of ASAP data is more than 6 bit (example)

Values that only consist of 6 bits or less shall have the following optimized A_GroupValue_Response-PDU format.

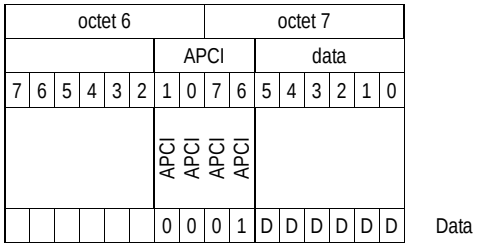


Figure 6 - A_GroupValue_Response-PDU, length of ASAP data is 6 bit or less (example)

The remote Application Layer shall accept the service response, map the ASAP to the TSAP and pass it with a T_Data_Group.req to the local Transport Layer. The parameters ack_request, TSAP, hop_count_type and priority shall be mapped to the corresponding parameters of the T_Data_Group.req primitive; the TSDU shall be an A_GroupValue_Response-PDU.

The local Application Layer shall map a T_Data_Group.ind primitive with TSDU = A_GroupValue_-Response-PDU to an A_GroupValue_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_GroupValue_Read. Acon primitive. More than one A_GroupValue_Read.Acon primitive may occur depending on the number of group members that have been configured to respond.

A_GroupValue_Read.req(ack_request, ASAP, priority, hop_count_type)

ack_request:	Data Link Layer Acknowledge requested or don't care
ASAP:	local reference of the Service Access Point
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority

A_GroupValue_Read.Lcon(ack_request ,ASAP, priority, hop_count_type, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
ASAP:		local reference of the Service Access Point
hop_count_type:		hop count 7 or standard
priority:		system, urgent, normal or low priority
a_status:	ok:	A_GroupValue_Read.req sent successfully with T_Data_Group service
	not_ok:	transmission of the associated T_Data_Group request frame did not succeed

A_GroupValue_Read.ind(ASAP, priority, hop_count_type)

ASAP:	local reference of the Service Access Point
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority

A_GroupValue_Read.res(ack_request, ASAP, priority, hop_count_type, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
ASAP:	local reference of the Service Access Point
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority
data:	the value of the associated Service Access Point

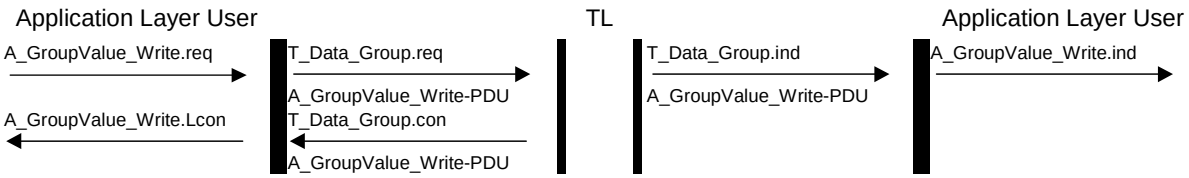
A_GroupValue_Read.Rcon(ack_request, ASAP, priority, hop_count_type, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
ASAP:		local reference of the Service Access Point
hop_count_type:		hop count 7 or standard
priority:		system, urgent, normal or low priority
data:		the value of the associated Service Access Point
a_status:	ok:	A_GroupValue_Read.res sent successfully with T_Data_Group service
	not_ok:	transmission of the associated T_Data_Group request frame did not succeed

A_GroupValue_Read.Acon(ASAP, priority, hop_count_type, data)

ASAP:	local reference of the Service Access Point
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority
data:	the value of the associated Service Access Point

3.1.3 A_GroupValue_Write-service



The A_GroupValue_Write.req primitive shall be applied by the user of Application Layer, to send an update of its ASAP to all connected ASAPs. The service shall not be confirmed by the remote application process, the confirmation shall be caused by the local T_Data_Group.con. The ASAP shall be associated to the TSAP via the Group Object Association Table, i.e. with a Group Address (see [02]). All group members shall receive the A_GroupValue_Write-PDU.

The local Application Layer shall accept the service request, map the ASAP to the TSAP and pass it with a T_Data_Group.req to the local Transport Layer. The user decides during configuration about this mapping. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Group.req primitive, the TSDU shall be an A_GroupValue_Write-PDU.

The remote Application Layer shall map a T_Data_Group.ind primitive with TSDU = A_Group-Value_Write-PDU to an A_GroupValue_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_GroupValue_Write.ind primitive. One A_GroupValue_Write.ind primitive shall be generated per ASAP that is assigned to the corresponding TSAP (i.e. Group Address).

Two different formats of the A_GroupValue_Write-PDU shall be used depending on the length of the value. The maximum length of the value shall be 14 octets. Unused data bits shall be set to zero.

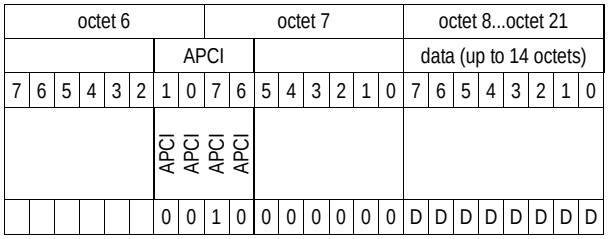


Figure 7 - A_GroupValue_Write-PDU,
length of ASAP data is more than 6 bit (example)

Values that only consist of 6 bits or less have the following optimized A_GroupValue_Write-PDU format:

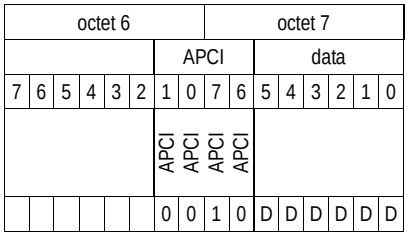


Figure 8 - A_GroupValue_Write-PDU,
length of ASAP data is 6 bit or less (example)

Prior to passing an A_GroupValue_Write.Lcon primitive to the local application process, the local Application Layer shall need a T_Data_Group.con from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_GroupValue_Write.Lcon (a_status = ok) to the local application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_GroupValue_Write.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated T_Data_Group.req did not succeed.

A_GroupValue_Write.req(ack_request, ASAP, priority, hop_count_type, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
ASAP:	local reference of the Service Access Point
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
data:	data of the associated application layer Service Access Point

A_GroupValue_Write.Lcon(ack_request, ASAP, priority, hop_count_type, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
ASAP:		local reference of the Service Access Point
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
data:		data of the associated application layer Service Access Point
a_status:	ok:	A_GroupValue_Write sent successfully with T_Data_Group service
	not_ok:	transmission of the associated T_Data_Group request frame did not succeed

A_GroupValue_Write.ind(ASAP, priority, hop_count_type, data)

ASAP:	local reference of the Service Access Point
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
data:	data of the associated application layer Service Access Point

3.2 Application Layer services on Broadcast Communication Mode

3.2.1 Goal

A broadcast communication mode shall be connectionless and shall connect one device with all others.

3.2.2 A_IndividualAddress_Write-service

The A_IndividualAddress_Write.req primitive shall be applied by the user of Application Layer to modify the Individual Address in a communication partner. The communication partner shall not be identified in the service, i.e. the destination must be defined by selecting a destination manually. This can be done by pressing a button on exactly one device that brings this device into a ´programming mode´, i.e. only the device where the button is pressed shall accept the A_IndividualAddress_Write.ind, others shall ignore it. The way that a product is set to ´programming mode´ may be manufacturer specific.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter of the T_Data_Broadcast.req primitive, the TSDU shall be an A_IndividualAddress_Write-PDU.

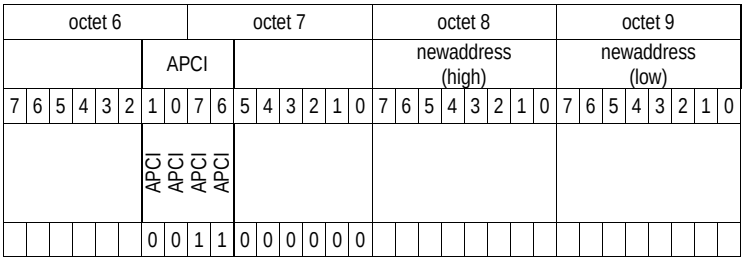


Figure 9 - A_IndividualAddress_Write-PDU (example)

The remote Application Layer shall map a T_Data_Broadcast.ind primitive with TSDU = A_Individual-Address_Write-PDU to an A_IndividualAddress_Write.ind primitive. The argument priority, implicitly with value ‘system’, shall be mapped to the corresponding argument priority of the A_Individual-Address_Write.ind primitive.

The application process shall ignore the A_IndividualAddress_Write.ind primitive if the device is not in ´programming mode´. Otherwise the local Individual Address shall be set to the new address.

Prior to passing an A_IndividualAddress_Write.Lcon primitive to the local application process, the local Application Layer shall need a T_Data_Broadcast.con from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_IndividualAddress_-Write.Lcon(a_status = ok) to the local application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_IndividualAddress_Write.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated T_Data_Broadcast.req did not succeed.

A_IndividualAddress_Write.req(ack_request, priority, hop_count_type, newaddress)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
newaddress:	the new value of the Individual Address

A_IndividualAddress_Write.Lcon(ack_request, priority, hop_count_type, newaddress, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
newaddress:		the new value of the Individual Address
a_status:	ok:	A_IndividualAddress_Write sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_IndividualAddress_Write.ind(priority, hop_count_type, newaddress)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
newaddress:	the new value of the Individual Address

3.2.3 A_IndividualAddress_Read-service

The A_IndividualAddress_Read.req primitive shall be applied by the user of Application Layer to read the Individual Address in a communication partner. The communication partner shall not be identified in the service, i.e. the destination must be defined by selecting a destination manually. This can be done by pressing a button on one or more devices that brings these devices into a ´programming mode´, i.e. only a device where the button is pressed shall accept the A_IndividualAddress_Read.ind, others shall ignore it. The way that a product is set to ´programming mode´ may be manufacturer specific.

The local Application Layer shall accept the service request and shall pass it with a T_Data_Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value 'system', shall be mapped to the corresponding parameter of the T_Data_Broadcast.req primitive; the TSDU shall be an A_IndividualAddress_Read-PDU.

The remote Application Layer shall map a T_Data_Broadcast.ind primitive with TSDU = A_IndividualAddress_Read-PDU to an A_IndividualAddress_Read.ind primitive. The argument priority, implicitly with value 'system', shall be mapped to the corresponding argument priority of the A_IndividualAddress_Read.ind primitive.

The application process shall respond to the A_IndividualAddress_Read.ind primitive with an A_IndividualAddress_Read.res primitive only if the device is in ´programming mode'.

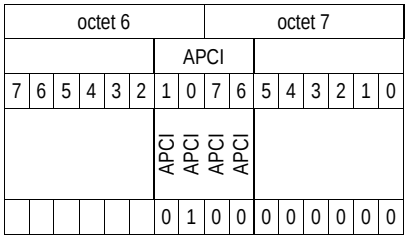


Figure 10 - A_IndividualAddress_Read-PDU (example)

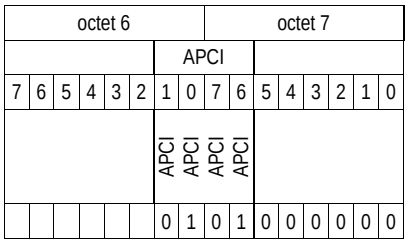


Figure 11: A_IndividualAddress_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Broadcast.req to the Transport Layer; the TSDU shall be an A_IndividualAddress_Response-PDU. The local Application Layer shall map a T_Data_Broadcast.ind primitive with TSDU = A_IndividualAddress_-Response-PDU to an A_IndividualAddress_Read.Acon primitive. The argument priority, implicitly with value ‘system’, shall be mapped to the corresponding argument priority of the A_IndividualAddress_-Read.Acon primitive. The argument source_address shall be mapped to the corresponding argument individual_addres of the A_IndividualAddress_Read.Acon primitive.

A_IndividualAddress_Read.req(ack_request, priority, hop_count_type)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard

A_IndividualAddress_Read.Lcon(ack_request, priority, hop_count_type, a_status)

ack_request:		Data Link Layer Acknowledge requested or don’t care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
a_status:	ok:	A_IndividualAddress_Read sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_IndividualAddress_Read.ind(priority, hop_count_type)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard

A_IndividualAddress_Read.res(ack_request, priority, hop_count_type, Individual Address)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
individual_address:	the value of the Individual Address

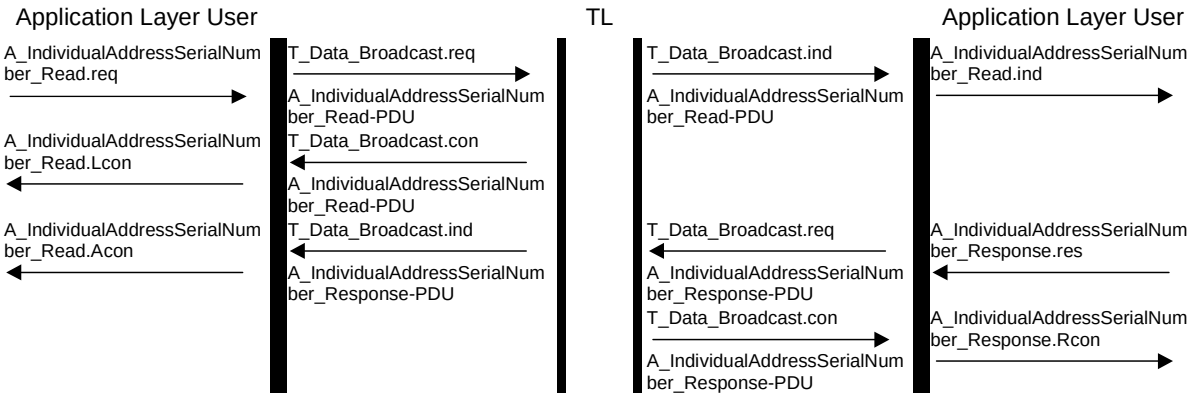
A_IndividualAddress_Read.Rcon(ack_request, priority, hop_count_type, individual_address, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
individual_address:		the value of the Individual Address
a_status:	ok:	A_IndividualAddress_Read sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_IndividualAddress_Read.Acon(priority, hop_count_type, Individual Address)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
individual_address:	the value of the Individual Address

3.2.4 A_IndividualAddressSerialNumber_Read-service



The A_IndividualAddressSerialNumber_Read.req primitive shall be applied by the user of Application Layer to read the Individual Address in a communication partner. The communication partner shall be identified using the unique KNX Serial Number (6 octets) of the device. KNX Serial numbers are administered by the KNX Association.

The local Application Layer shall accept the service request and pass it with a T_Data_Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value 'system', shall be mapped to the corresponding parameter of the T_Data_Broadcast.req primitive; the TSDU shall be an A_Individual-AddressSerialNumber_Read-PDU.

Prior to passing an A_IndividualAddressSerialNumber_Read.Lcon to the local user, the local Application Layer shall need a T_Data_Broadcast.con from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_IndividualAddressSerial-Number_Read.Lcon (a_status = ok) to the local user. If the confirmation is negative (a_status = not_ok), the local Application Layer shall pass an A_IndividualAddressSerialNumber_Read.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated A_IndividualAddressSerial-Number_Read.req did not succeed.

The remote Application Layer shall map a T_Data_Broadcast.ind primitive with TSDU = A_Individual-AddressSerialNumber_Read-PDU to an A_IndividualAddressSerialNumber_Read.ind primitive. The argument priority, implicitly with value 'system', shall be mapped to the corresponding argument priority of the A_IndividualAddressSerialNumber_Read.ind primitive.

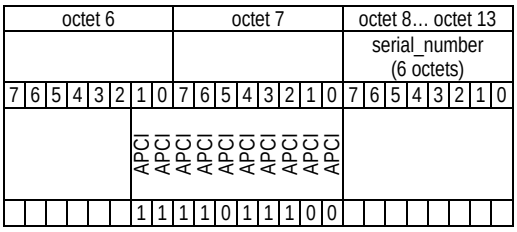


Figure 12 - A_IndividualAddressSerialNumber_Read-PDU (example)

The application process shall respond to the A_IndividualAddressSerialNumber_Read.ind primitive with an A_IndividualAddressSerialNumber_Read.res primitive, if the KNX Serial Number received is equal to the KNX Serial Number of the device.

The remote Application Layer shall accept the service response and pass it with a T_Data_Broadcast.req to the remote Transport Layer; the TSDU shall be an A_IndividualAddressSerialNumber_Response-PDU.

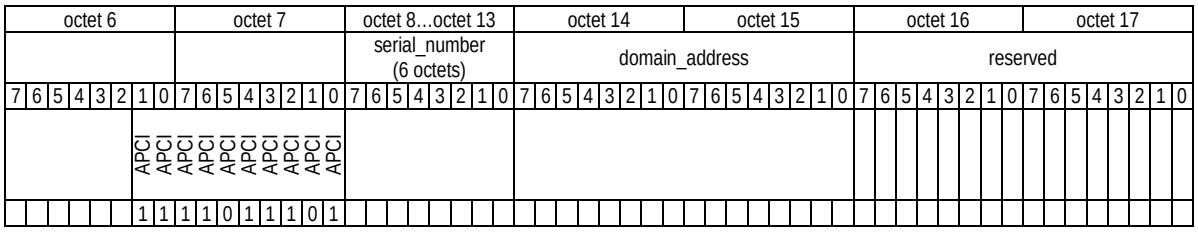


Figure 13 - A_IndividualAddressSerialNumber_Response-PDU (example)

Prior to passing an A_IndividualAddressSerialNumber_Read.Rcon to the remote user, the remote Application Layer shall need a T_Data_Broadcast.con from the remote Transport Layer. If the confirmation is positive (t_status = ok), the remote Application Layer shall pass a positive A_Individual-AddressSerialNumber_Read.Rcon (a_status = ok) to the remote user. If the confirmation is negative (a_status = not_ok), the remote Application Layer shall pass an A_IndividualAddress-SerialNumber_Read.Rcon (a_status = not_ok) to the remote user indicating that the transmission of the associated A_IndividualAddressSerialNumber_Read.res did not succeed.

The local Application Layer shall map a T_Data_Broadcast.ind primitive with TSDU = A_Individual-AddressSerialNumber_Response-PDU to an A_IndividualAddressSerialNumber_Read.Acon primitive. The argument priority, implicitly with value ‘system’, shall be mapped to the corresponding argument priority of the A_IndividualAddressSerialNumber_Read.Acon primitive. The argument source_address shall be mapped to the argument individual_address.

A_IndividualAddressSerialNumber_Read.req (ack_request, priority, hop_count_type, serial_number)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
serial_number:	the KNX Serial Number

A_IndividualAddressSerialNumber_Read.Lcon (ack_request, priority, hop_count_type, serial_number, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
serial_number:		the KNX Serial Number
a_status	ok:	A_IndividualAddressSerialNumber_Read sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_IndividualAddressSerialNumber_Read.ind (priority, hop_count_type, serial_number)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
serial_number:	the KNX Serial Number

A_IndividualAddressSerialNumber_Read.res (ack_request, priority, hop_count_type, serial_number, domain_address)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
serial_number:	the KNX Serial Number
domain_address:	the Domain Address of the remote device

A_IndividualAddressSerialNumber_Read.Rcon (ack_request, priority, hop_count_type, serial_number, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
serial_number:		the KNX Serial Number
a_status	ok:	A_IndividualAddressSerialNumber_Response sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_IndividualAddressSerialNumber_Read.Acon (priority, hop_count_type, serial_number, individual_address, domain_address)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
serial_number:	the KNX Serial Number
individual_address:	the value of the Individual Address
domain_address:	the Domain Address of the remote device

3.2.5 A_IndividualAddressSerialNumber_Write-service

The A_IndividualAddressSerialNumber_Write.req primitive shall be applied by the user of Application Layer, to modify the Individual Address in a communication partner. The communication partner shall be identified using the unique KNX Serial Number (6 octets) of the device.

The local Application Layer shall accept the service request and pass it with a T_Data_Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value 'system', shall be mapped to the corresponding parameter of the T_Data_Broadcast.req primitive; the TSDU shall be an A_Individual-AddressSerialNumber_Write-PDU.

octet 6						octet 7								octet 8 ... octet 13								octet 14								octet 15								octet 16 ... octet 19							
						APCI								serial_number (6 octets)								newaddress (high)								newaddress (low)								reserved (4 octets)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																																
						1	1	1	1	0	1	1	1	1	0																														

Figure 14 - A_IndividualAddressSerialNumber_Write-PDU (example)

The remote Application Layer shall map a T_Data_Broadcast.ind primitive with TSDU = A_Individual-AddressSerialNumber_Write-PDU to an A_IndividualAddressSerialNumber_Write.ind primitive. The argument priority shall be mapped to the corresponding argument priority of the A_IndividualAddress-SerialNumber_Write.ind primitive.

Prior to passing an A_IndividualAddressSerialNumber_Write.Lcon primitive to the local application process, the local Application Layer shall need a T_Data_Broadcast.con from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_IndividualAddressSerialNumber_Write.Lcon(a_status = ok) to the local application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_IndividualAddressSerialNumber_Write.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated T_Data_Broadcast.req did not succeed.

A_IndividualAddressSerialNumber_Write.req(ack_request, priority, hop_count_type, serial_number, newaddress)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
serial_number:	the serial number
newaddress:	the new value of the Individual Address

A_IndividualAddressSerialNumber_Write.Lcon(ack_request, priority, hop_count_type, newaddress, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
newaddress:		the new value of the Individual Address
a_status:	ok:	A_IndividualAddressSerialNumber_Write sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_IndividualAddressSerialNumber_Write.ind(priority, hop_count_type, serial_number, newaddress)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
serial_number:	the serial number
newaddress:	the new value of the Individual Address

3.2.6 A_NetworkParameter_Read-service

The A_NetworkParameter_Read.req primitive shall be applied by the user of Application Layer of the management client to check about the configuration of a network parameter. The service request shall be communicated on either point-to-point connectionless or on point-to-all-points (broadcast) communication mode. This shall be specified in the parameter comm_mode_req of the A_NetworkParameter_-Read.req primitive. A device shall respond to the service if it complies with the conditions specified in the service parameters.

The local Application Layer shall accept the service request. If the parameter comm_mode_req equals “point-to-point, connectionless”, it shall pass the service with a T_Data_Individual.req to the local Transport Layer; the parameter ASAP shall be mapped to the corresponding parameter TSAP of the T_Data_Individual.req primitive. If the parameter comm_mode_req equals “point-to-all-points connectionless”, it shall pass the service with a T_Data_Broadcast.req to the local Transport Layer. The TSDU shall in both cases always be an A_NetworkParameter_Read-PDU. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter of the T_Data_Broadcast.req primitive, the TSDU shall be an A_NetworkParameter_Read-PDU.

If the remote Application Layer receives a T_Data_Individual.ind or a T_Data_Broadcast.ind with TSDU = A_NetworkParameter_Read-PDU, it shall map the service primitive to an A_NetworkParameter_Read.ind primitive. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter priority of the A_NetworkParameter_Read.ind primitive.

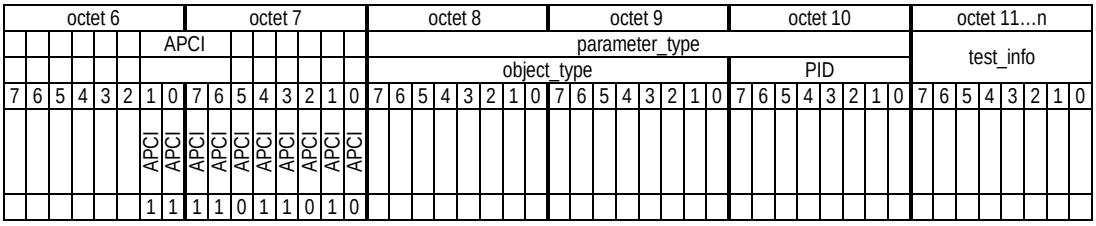


Figure 15 - A_NetworkParameter_Read-PDU (example)

If the remote application process finds the conditions for replying are not fulfilled, this is if it does not support the read network parameter or the check of its investigated parameters against the test information is negative, it shall ignore the service.

If the remote application process accepts the service, it shall respond to the A_NetworkParameter_-Read.ind primitive with an A_NetworkParameter_Read.res primitive after a random wait time in the range of 0 s …10 x T_{media} s [2]. This random wait time is per parameter_type specified in [05]. For parameter types read in point-to-point communication mode, the random wait time is 0 s.The data in the response shall depend on the network parameter type being read.

[2] T_{media} is specified in [05].

The remote Application Layer shall accept the service response. If the parameter comm_mode equals “point-to-point, connectionless”, it shall pass the service with a T_Data_Individual.req to the local Transport Layer; the parameter ASAP shall be mapped to the corresponding parameter TSAP of the T_Data_Individual.req primitive. If the parameter comm_mode equals “point-to-all-points connectionless”, it shall pass the service with a T_Data_Broadcast.req to the local Transport Layer. The TSDU shall always be an A_NetworkParameter_Response-PDU.

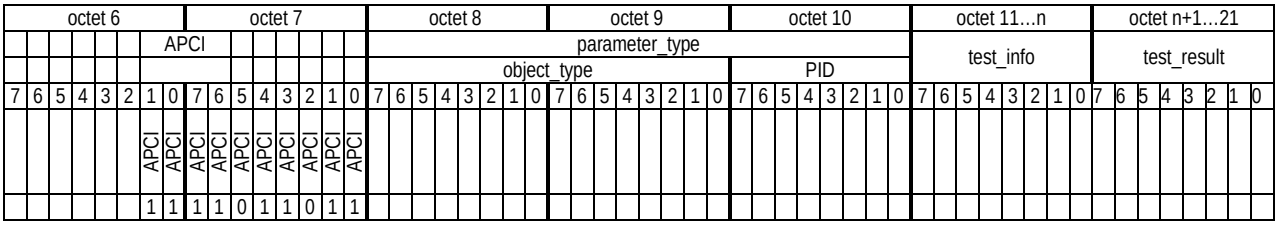


Figure 16 - A_NetworkParameter_Response-PDU (example)

On reception of a T_Data_Broadcast.ind primitive or a T_Data_Individual.ind, with TSDU = A_NetworkParameter_Response-PDU the local Application Layer shall map this service to an A_NetworkParameter_Read.Acon primitive. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter priority of the A_NetworkParameter_Read.Acon primitive. The parameter TSAP shall be mapped to the corresponding parameter ASAP of the A_NetworkParameter_Read.Acon primitive.

Error and exception handling

The exception handling shall depend on the communication mode of the request as specified in Table 2.

Table 2 – Exceptions for A_NetworkParameter_Read and handling

Exception:	**communication mode**	
	point-to-all-points (broadcast)	**point-to-point (unicast)**
• the network parameter to be read is not supported	ignore	negative response
• the network parameter to be read is supported and the check is negative	ignore	negative response

The negative response shall be as follows.

- If the remote application process receives an A_NetworkParameter_Read-PDU with an unknown object_type within parameter_type, this is, object_type is not supported by the remote application, then it shall respond with an A_NetworkParameter_Response-PDU with object_type = FFFFh, PID = FFh and without test_info and test_result fields.
- If the remote application process receives an A_NetworkParameter_Read-PDU with an known object_type but with unknown PID within parameter_type, this is, the requested PID is not supported by the remote application, then it shall respond with an A_NetworkParameter_Response-PDU with object_type = requested object_type, PID = FFh and without test_info and test_result fields.
- The negative response for a negative check of the investigated parameters against the test information is specified per parameter_type either in [05] or [06].

A_NetworkParameter_Read.req(ASAP, comm_mode_req, hop_count_type, parameter_type, priority, test_info)

ASAP:		local reference of the Service Access Point or Individual Address
comm_mode_req:		point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:		hop count 7 or standard
parameter_type:		network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
test_info:		value against which the resource indicated by parameter_type is tested

A_NetworkParameter_Read.Lcon(ASAP, comm_mode_req, hop_count_type, parameter_type, priority, test_info, a_status)

ASAP:		local reference of the Service Access Point or Individual Address
comm_mode_req:		point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:		hop count 7 or standard
parameter_type:		network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
test_info		value against which the resource indicated by parameter_type is tested
a_status:	ok:	A_NetworkParameter_Read.req sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_NetworkParameter_Read.ind(ASAP, comm_mode_req, hop_count_type, parameter_type, priority, test_info)

ASAP:		local reference of the Service Access Point or individual address
comm_mode_req:		point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:		hop count 7 or standard
parameter_type:		network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
test_info:		value against which the resource indicated by parameter_type is tested

A_NetworkParameter_Read.res(ASAP, comm_mode, hop_count_type, parameter_type, priority, test_info, test_result)

ASAP:	local reference of the Service Access Point or individual address
comm_mode:	point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:	hop count 7 or standard
individual_address:	the destination address for the service
parameter_type:	network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	value against which the resource indicated by parameter_type is tested
test_result:	parameter_type dependent response

A_NetworkParameter_Read.Rcon(ASAP, comm_mode, hop_count_type, parameter_type, priority, test_info, test_result, a_status)

ASAP:		local reference of the Service Access Point or individual address
comm_mode:		point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:		hop count 7 or standard
individual_address:		the destination address for the service
parameter_type:		network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
test_info:		value against which the resource indicated by parameter_type is tested
test_result:		parameter_type dependent response
a_status:	ok:	A_NetworkParameter_Read.res sent successfully with T_Data_Broadcast or T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Broadcast or T_Data_Individual request frame did not succeed

A_NetworkParameter_Read.Acon(ASAP, hop_count_type, individual_address, parameter_type, priority, test_info, test_result)

ASAP:	local reference of the Service Access Point or individual address
hop_count_type:	hop count 7 or standard
individual_address:	the individual address of the device that has sent the response
parameter_type:	network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	value against which the resource indicated by parameter_type is tested
test_result:	parameter_type dependent response

3.2.7 A_NetworkParameter_Write-service

The A_NetworkParameter_Write.req primitive shall be applied by the user of Application Layer of the management client to set network configuration information in one or multiple management servers. The service shall be transmitted on either point-to-point connection-less or on point-to-all points (broadcast) communication mode. This shall be specified in the parameter comm_mode of the A_NetworkParameter_Write.req primitive.

The local Application Layer shall accept the service request. If the parameter comm_mode equals “point-to-point, connectionless”, it shall pass the service with a T_Data_Individual.req to the local Transport Layer; the parameter ASAP shall be mapped to the corresponding parameter TSAP of the T_Data_Individual.req primitive. If the parameter comm_mode equals “point-to-all-points connectionless”, it shall pass the service with a T_Data_Broadcast.req to the local Transport Layer. The TSDU shall in both cases always be an A_NetworkParameter_Write-PDU.

If the remote Application Layer receives a T_Data_Individual.ind or a T_Data_Broadcast.ind with TSDU = A_NetworkParameter_Write-PDU, it shall map the service primitive to an A_Network-Parameter_Write.ind primitive. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter priority of the A_NetworkParameter_Write.ind primitive.

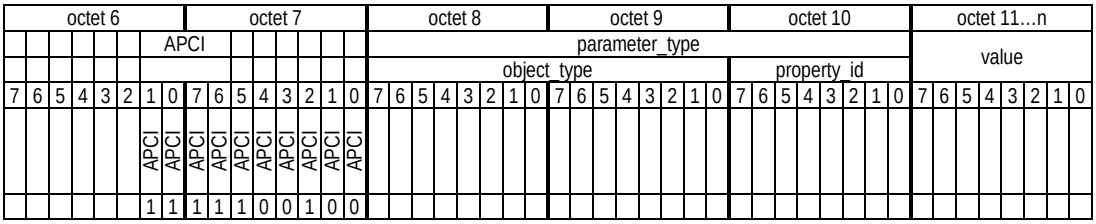


Figure 17 - A_NetworkParameter_Write-PDU (example)

Prior to passing an A_NetworkParameter_Write.Lcon primitive to the local application process, the local Application Layer shall need a T_Data_Individual.con or a T_Data_Broadcast.con, depending on the parameter comm_mode of the service request, from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_Network-Parameter_Write.Lcon(a_status = ok) to the local application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_NetworkParameter_Write.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated Transport Layer specification did not succeed.

NOTE This service is as such not confirmed by the remote partner. The A_NetworkParameter_Write.Lcon is only a local confirmation caused by the local Transport Layer confirmation, basically caused by the Data Link Layer confirmation (ok, not_ok).

A_NetworkParameter_Write.req(ASAP, comm_mode, hop_count_type, parameter_type, priority, value)

ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:	hop count 7 or standard
parameter_type:	the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
value:	value to which the network parameter indicated by parameter_type shall be set

A_NetworkParameter_Write.Lcon(ASAP, comm_mode, hop_count_type, parameter_type, priority, value, a_status)

ASAP:		local reference of the Service Access Point or Individual Address
comm_mode:		point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:		hop count 7 or standard
parameter_type:		the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
value:		value to which the network parameter indicated by parameter_type shall be set
a_status:	ok:	A_NetworkParameter_Write-PDU sent successfully with the requested Transport Layer service
	not_ok:	transmission of the requested Transport Layer service did not succeed

A_NetworkParameter_Write.ind(ASAP, parameter_type, priority, value)

ASAP:	local reference of the Service Access Point or Individual Address
parameter_type:	the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
value:	value to which the network parameter indicated by parameter_type shall be set

Error and exception handling

- **Unknown parameter_type**

If an unknown parameter_type is received, the receiver shall neglect the A_NetworkParameter_Write-PDU without further action.

- **Data consistency**

In general, if over this service data is accessed that is also accessible over additional mechanisms, like over the A_PropertyValue_Read or the A_PropertyValue_Write, via dedicated services like the A_IndividualAddress_Read or the A_IndividualAddress_Write, as a memory mapped Resource or via local access using EMI, the receiver shall take care of consistency between these access modes.

The network – and device Management Procedures (see [05]) that base on this Application Layer services may specify procedure specific error handling.

3.2.8 A_NetworkParameter_InfoReport

The A_NetworkParameter_InfoReport.req primitive shall be applied by the user of Application Layer, which shall be a Property Server, to inform one or all Management Clients about the value of an Interface Object Property. This Interface Object Property shall be identified by the field parameter_type, which shall contain the Object Type and the Property Identifier. The field test_info shall have a value that shall be specified for each use case of this service, that is, for each specific parameter_type.

- If the Management Server may hold more than one instance of the Object Type, then it shall be specified for the specific use of this service if the Object Instance shall be encoded. If it is to be encoded, then the Object Instance shall be encoded as part of the field test_result.
- The Management Server may typically use this service to announce the value of one of its Properties. If it is to be encoded, then the Property Value shall be encoded as part of the field test_result.

NOTE 2 For possible use of the service A_NetworkParameter_InfoReport, please refer to the specification of the Management Procedure DMP_InterfaceObjectInfoReport_RCl in [05] and its examples.

The service request shall be communicated on either point-to-point connectionless or on point-to-all points (broadcast) communication mode. This shall be specified in the service parameter comm_mode_req of the A_NetworkParameter_InfoReport.req primitive. The APCI shall be the APCI_NetworkParameter_Response.

The local Application Layer shall accept the service request.

- If the parameter comm_mode_req equals “point-to-point, connectionless”, it shall pass the service with a T_Data_Individual.req to the local Transport Layer; the parameter ASAP shall be mapped to the corresponding parameter TSAP of the T_Data_Individual.req primitive.
- If the parameter comm_mode_req equals “point-to-all-points connectionless”, it shall pass the service with a T_Data_Broadcast.req to the local Transport Layer.

The TSDU shall in both cases always be an A_NetworkParameter_InfoReport-PDU.

The arguments hop_count_type and priority shall be mapped to the corresponding arguments hop_count_type and priority of the T_Data_Individual.req respectively the T_Data_Broadcast.req primitive.

If the remote Application Layer receives a T_Data_Individual.ind or a T_Data_Broadcast.ind with TSDU = A_NetworkParameter_InfoReport-PDU, it shall map the service primitive to an A_NetworkParameter_InfoReport.ind primitive.

The arguments priority shall be mapped to the corresponding argument priority of the A_NetworkParameter_InfoReport.ind primitive.

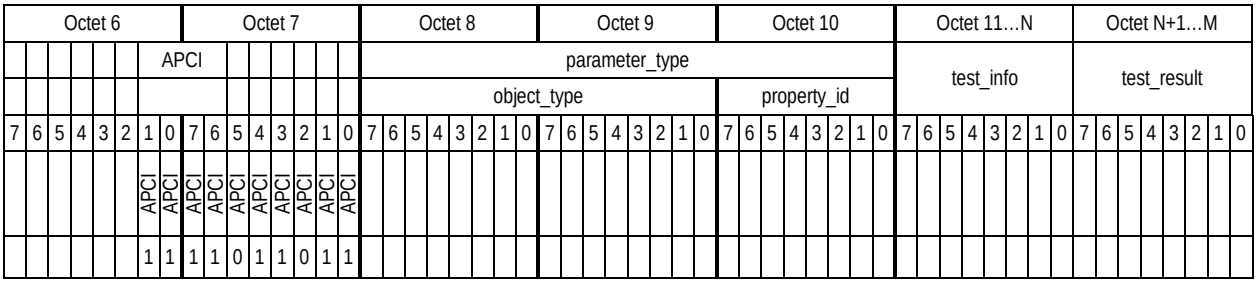


Figure 18 - A_NetworkParameter_InfoReport-PDU (example)

A_NetworkParameter_InfoReport.req(ASAP, comm_mode_req, hop_count_type, parameter_type, priority, test_info, test_result)

ASAP:	local reference of the Service Access Point or Individual Address
comm_mode_req:	point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:	hop count 0, 7 or Network Layer Parameter
parameter_type:	the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	parameter type dependent value
test_result:	parameter type dependent value

A_NetworkParameter_InfoReport.Lcon(ASAP, comm_mode_req, hop_count_type, parameter_type, priority, test_info, test_result, a_status)

ASAP:	local reference of the Service Access Point or Individual Address	
comm_mode_req	point-to-all-points connectionless communication mode or point-to-point connectionless communication mode	
hop_count_type:	hop count 0, 7 or Network Layer Parameter	
parameter_type:	the network parameter that shall be set, structured as Interface Object Type and Property Identifier	
priority:	system, urgent, normal or low priority	
test_info:	parameter type dependent value	
test_result	parameter type dependent value	
a_status:	ok:	A_NetworkParameter_InfoReport sent successfully with T_Data_Individual - or T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Individual – or T_Data_Broadcast request frame did not succeed

A_NetworkParameter_InfoReport.ind(ASAP, comm_mode_req, hop_count_type, parameter_type, priority, test_info, test_result)

ASAP:	the Individual Address of the device that has sent the A_NetworkParameter_InfoReport-PDU.
comm_mode_req	point-to-all-points connectionless communication mode or point-to-point connectionless communication mode
hop_count_type:	hop count 0, 7 or Network Layer Parameter
parameter_type:	the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	parameter type dependent value
test_result:	parameter type dependent value

Error - and – exception handling

No error handling shall be applied. The service shall not be remote confirmed.

3.3 Application Layer Services on System Broadcast communication mode

3.3.1 Introduction

The Application Layer PDUs specified in this clause 3.2 shall use the point-to-all-points, connectionless (system broadcast) communication mode (T_Data_SystemBroadcast) when transmitted on a Physical Layer connection to a KNX open medium (Powerline, Radio Frequency…). If these PDUs are to be transmitted on a KNX closed medium (Twisted Pair…) then the point-to-*domain*, connectionless (broadcast) communication mode (T_Data_Broadcast) shall be used.

This may mean that a Coupler between a closed medium and an open medium may have to convert an incoming frame on broadcast communication to an outgoing frame on system broadcast communication and vice versa. This is specified in [01] clause “System Broadcast conversion”.

3.3.2 A_DeviceDescriptor_InfoReport-service

NOTE 3 This service uses the same 4 bit APCI as APCI_DeviceDescriptor_Response.

The A_DeviceDescriptor_InfoReport.req primitive shall be applied by the local user of Application Layer to send the Device Descriptor to remote communication partners. The remote communication partner shall not be identified in the service, this is, the destination shall be defined by setting it in “teaching mode”. Only remote devices in this mode shall accept the A_DeviceDescriptor_InfoReport-PDU, other device shall ignore it.

The local Application Layer shall accept the service request and pass it with a T_Data_System-Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value “system”, shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req primitive, the TSDU shall be an A_DeviceDescriptor_InfoReport-PDU.

octet 6								octet 7								octet 8								…octet n							
						APCI				descriptor_type						device_descriptor															
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI																						
						1	1	0	1																						

Figure 19 - A_DeviceDescriptor_InfoReport-PDU (example)

The remote Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DeviceDescriptor_InfoReport-PDU to an A_DeviceDescriptor_InfoReport.ind primitive. The argument priority, implicitly with value “system”, shall be mapped to the corresponding argument priority of the A_DeviceDescriptor_InfoReport.ind primitive.

The remote application process shall ignore the A_DeviceDescriptor_InfoReport.ind primitive if the remote device is not in “teaching mode”. Otherwise the received Device Descriptor shall be processed.

Prior to passing an A_DeviceDescriptor_InfoReport.Lcon primitive to the local application process, the local Application Layer shall need a T_Data_SystemBroadcast.con from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_Device-Descriptor_InfoReport.Lcon (a_status = ok) to the local Application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_DeviceDescriptor_-InfoReport.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated T_Data_SystemBroadcast.req did not succeed.

A_DeviceDescriptor_InfoReport.req(ack_request, descriptor_type, device_descriptor, hop_count_type, priority)

ack_request:	Data Link Layer Acknowledge requested or don’t care
descriptor_type:	type of the device descriptor
device_descriptor:	the device descriptor of the device
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority

A_DeviceDescriptor_InfoReport.Lcon(ack_request, descriptor_type, device_descriptor, hop_count_type priority, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
descriptor_type:		type of the descriptor
device_descriptor:		the device descriptor of the device
hop_count_type:		hop count 7 or standard
priority:		system, urgent, normal or low priority
a_status:	ok:	A_DeviceDescriptor_InfoReport sent successfully with T_Data_SystemBroadcast service
	not_ok:	transmission of the associated T_Data_SystemBroadcast request frame did not succeed

A_DeviceDescriptor_InfoReport.ind(descriptor_type device_descriptor, hop_count_type, priority)

descriptor_type:	type of the descriptor
device_descriptor:	the device descriptor of the device
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority

Error and Exception handling

No error handling is applied. The service is not remote confirmed.

3.3.3 A_DomainAddress_Write-service

The A_DomainAddress_Write.req primitive shall be applied by the user of Application Layer, to modify the Domain Address in a communication partner. The Domain Address shall be encoded according the Domain Address format used on the medium of the communication partner either as a 2 octet value (KNX-PL110) or a 6 octet value (KNX-RF). The communication partner shall not be identified in the service, i.e. the destination must be defined by selecting a destination manually. This can be done by pressing a button on exactly one device that brings this device into a ´programming´ mode, i.e. only the device where the button is pressed shall accept the A_DomainAddress_Write.ind, others shall ignore it. The way that a product is set to ´programming´ mode may be manufacturer specific.

The local Application Layer shall accept the service request and shall pass it with a T_Data_System-Broadcast.req to the local Transport Layer. The parameter priority shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req primitive. The TSDU shall be an A_Domain-Address_Write-PDU.

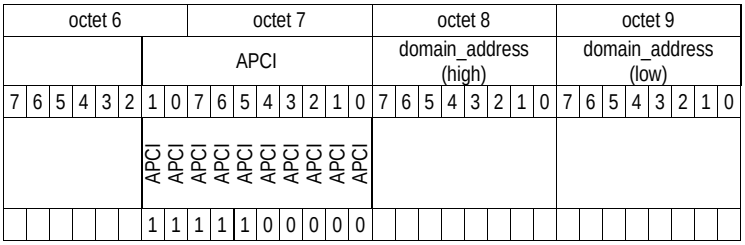


Figure 20 - A_DomainAddress_Write-PDU for a 2 octet DoA format (example)

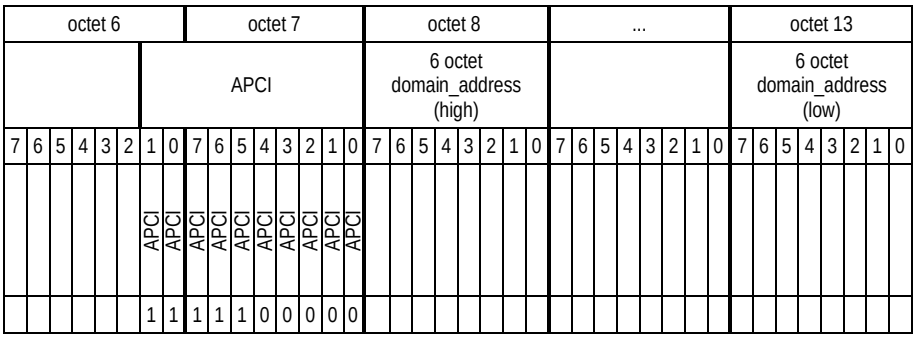


Figure 21 - A_DomainAddress_Write-PDU for a 6 octet DoA format (example)

The remote Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddress_Write-PDU to an A_DomainAddress_Write.ind primitive. The arguments priority and domain_address_new shall be mapped to the corresponding arguments of the A_DomainAddress_Write.ind primitive.

The application process shall ignore the A_DomainAddress_Write.ind primitive if the device is not in ´programming´ mode.

Prior to passing an A_DomainAddress_Write.Lcon primitive to the local application process, the local Application Layer needs a T_Data_SystemBroadcast.con from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_DomainAddress_Write.Lcon(a_status = ok) to the local application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_DomainAddress_Write.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated T_Data_SystemBroadcast.req did not succeed.

A_DomainAddress_Write.req(ack_request, priority, hop_count_type, domain_address_new)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
domain_address_new:	the new value of the Domain Address

A_DomainAddress_Write.Lcon(ack_request, priority, hop_count_type, domain_address_new, a_status)

ack_request:		Data Link Layer Acknowledge requested or don’t care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
domain_address_new:		the new value of the Domain Address
a_status:	ok:	A_DomainAddress_Write sent successfully with T_Data_SystemBroadcast service
	not_ok:	transmission of the associated T_Data_SystemBroadcast request frame did not succeed

A_DomainAddress_Write.ind(priority, hop_count_type, domain_address_new)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
domain_address_new:	the new value of the Domain Address

3.3.4 A_DomainAddress_Read-service

The A_DomainAddress_Read.req primitive shall be applied by the user of Application Layer, to read the Domain Address from a communication partner. The communication partner shall not be identified in the service, i.e. the destination must be defined by selecting a destination manually. This can be done by pressing a button on one or more devices that brings these devices into a ´programming´ mode, i.e. only a device where the button is pressed shall accept the A_DomainAddress_Read.ind, others shall ignore it. The way that a product is set to ´programming´ mode may be manufacturer specific.

The local Application Layer shall accept the service request and shall pass it with a T_Data_System-Broadcast.req to the local Transport Layer. The parameter priority shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req primitive, the TSDU shall be an A_DomainAddress_-Read-PDU.

octet 6								octet 7							
						APCI									
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI
						1	1	1	1	1	0	0	0	0	1

Figure 22 - A_DomainAddress_Read-PDU (example)

The remote Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddress_Read-PDU to an A_DomainAddress_Read.ind primitive. The argument priority shall be mapped to the corresponding argument priority of the A_DomainAddress_Read.ind primitive.

The remote application process shall respond to the A_DomainAddress_Read.ind primitive with an A_DomainAddress_Read.res primitive only if the device is in ´programming´ mode. The Domain Address shall be encoded according the Domain Address format used on the medium of the remote application process either as a 2 octet value (KNX-PL110) or a 6 octet value (KNX-RF).

The remote Application Layer shall accept the service response and shall pass it with a T_Data_System-Broadcast.req to the Transport Layer; the TSDU shall be an A_DomainAddress_Response-PDU.

octet 6						octet 7								octet 8								octet 9							
						APCI								domain_address (high)								domain_address (low)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																
						1	1	1	1	1	0	0	0	1	0																

Figure 23 - A_DomainAddress_Response-PDU for a 2 octet DoA format (example)

octet 6						octet 7								octet 8								…								octet 13							
						APCI								6 octet domain_address (high)								…								6 octet domain_address (low)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																								
						1	1	1	1	1	0	0	0	1	0																								

Figure 24 - A_DomainAddress_Response-PDU for a 6 octet DoA format (example)

The local Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddress_Response-PDU to an A_DomainAddress_Read.Acon primitive. The argument priority shall be mapped to the corresponding argument priority of the A_DomainAddress_Read.Acon primitive.

A_DomainAddress_Read.req(ack_request, priority, hop_count_type)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard

A_DomainAddress_Read.Lcon(ack_request, priority, hop_count_type, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
a_status:	ok:	A_DomainAddress_Read.req sent successfully with T_Data_SystemBroadcast service
	not_ok:	transmission of the associated T_Data_SystemBroadcast request frame did not succeed

A_DomainAddress_Read.ind(priority, hop_count_type)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard

A_DomainAddress_Read.res(ack_request, priority, hop_count_type, domain_address)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
domain_address:	the value of the Domain Address

A_DomainAddress_Read.Acon(priority, hop_count_type, domain_address)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
domain_address:	the value of the Domain Address

3.3.5 A_DomainAddressSelective_Read-service

The A_DomainAddressSelective_Read.req primitive shall be applied by the user of Application Layer, to read the Domain Address from the communication partner that shall be identified within the service. This service is particularly used to check the existence of any open media devices with the specified Domain Address in possibly neighbouring installations.

NOTE 4 The A_DomainAddressSelective_Read-service is only specified for the 2 octet format DoA. It is not available for the 6 octet format DoA.

The local Application Layer shall accept the service request and shall pass it with a T_Data_System-Broadcast.req to the local Transport Layer. The parameter priority shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req primitive, the TSDU shall be an A_DomainAddress-Selective_Read-PDU.

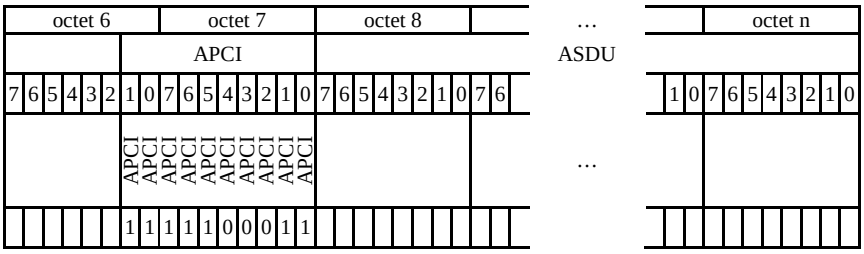


Figure 25 - A_DomainAddressSelective_Read-PDU (example)

The remote Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddressSelective_Read-PDU to an A_DomainAddressSelective_Read.ind primitive. The priority and the ASDU shall be passed to the remote Application Layer user.

If the remote Application Layer user confirms the A_DomainAddressSelective_Read.ind primitive, with ASDU specific data and wait times, with an A°DomainAddress_Read.res, then the remote Application Layer shall accept the service response and shall pass it with a T_Data_System¬Broadcast.req to the remote Transport Layer; the TSDU shall be an A_DomainAddress_Response PDU.

A_DomainAddressSelective_Read.req(priority, hop_count_type, ASDU)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASDU:	service specific parameters

A_DomainAddressSelective_Read.Lcon(priority, hop_count_type, domain_address, start_address, range, a_status)

priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASDU:		service specific parameters
a_status:	ok:	A_DomainAddressSelective_Read-PDU sent successfully with T_Data_SystemBroadcast service
	not_ok:	transmission of the associated T_Data_SystemBroadcast request frame did not succeed.

A_DomainAddressSelective_Read.ind(priority, hop_count_type, domain_address, start_address, range)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASDU:	service specific parameters

3.3.6 A_DomainAddressSerialNumber_Read-service

The A_DomainAddressSerialNumber_Read.req primitive shall be applied by the user of the Application Layer to read the Domain Address in a communication partner. The communication partner shall be identified using the unique KNX Serial Number (6 octet) of the communication partner.

The local Application Layer shall accept the service request and shall pass it with a T_Data_System-Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req primitive; the TSDU shall be an A_DomainAddressSerialNumber_Read-PDU.

The remote Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddressSerialNumber_Read-PDU to an A_DomainAddressSerialNumber_Read.ind primitive. The argument priority, implicitly with value ‘system’, shall be mapped to the corresponding argument priority of the A_DomainAddressSerialNumber_Read.ind primitive.

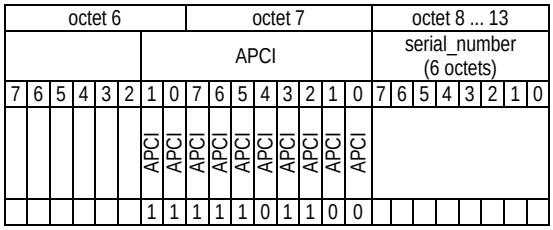


Figure 26 - A_DomainAddressSerialNumber_Read-PDU (example)

The application process shall respond to the A_DomainAddressSerialNumber_Read.ind primitive with an A_DomainAddressSerialNumber_Read.res primitive, if the KNX Serial Number received is equal to the KNX Serial Number of the device. The Domain Address shall be encoded according the Domain Address format used on the medium of the remote application process either as a 2 octet value (KNX-PL110) or a 6 octet value (KNX-RF).

The remote Application Layer shall accept the service response and shall pass it with a T_Data_System-Broadcast.req to the remote Transport Layer; the TSDU shall be an A_DomainAddressSerial-Number_Response-PDU.

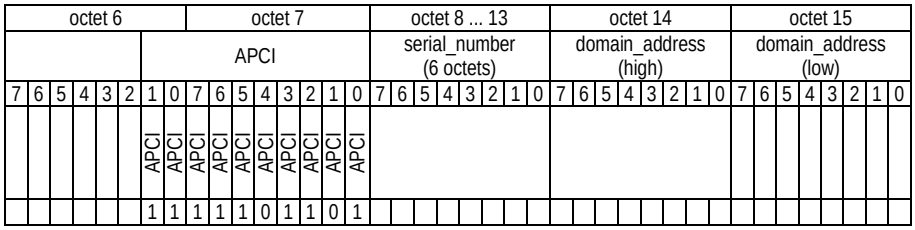


Figure 27 - A_DomainAddressSerialNumber_Response-PDU for a 2 octet DoA format (example)

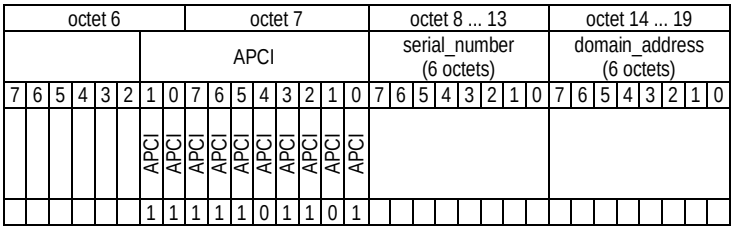


Figure 28 - A_DomainAddressSerialNumber_Response-PDU for a 6 octet DoA format (example)

The local Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddressSerialNumber_Response-PDU to an A_DomainAddressSerialNumber_Read.con primitive. The argument priority, implicitly with value ‘system’, shall be mapped to the corresponding argument priority of the A_DomainAddress-SerialNumber_Read.con primitive.

A_DomainAddressSerialNumber_Read.req (ack_request, hop_count_type, priority, serial_number)

ack_request:	Data Link Layer Acknowledge requested or don’t care
hop_count_type:	hop count 7 or standard
priority:	system
serial_number:	the KNX Serial Number of the remote device

A_DomainAddressSerialNumber_Read.Lcon (ack_request, hop_count_type, priority, serial_number, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
hop_count_type:		hop count 7 or standard
priority:		system
serial_number:		the KNX Serial Number of the remote device
a_status	ok:	A_DomainAddressSerialNumber_Read-PDU sent successfully with T_Data_Broadcast service
	not_ok:	transmission of the associated T_Data_Broadcast request frame did not succeed

A_DomainAddressSerialNumber_Read.ind (hop_count_type, priority, serial_number)

hop_count_type:	hop count 7 or standard
priority:	system
serial_number:	the KNX Serial Number of the remote device

A_DomainAddressSerialNumber_Read.res (ack_request, domain_address, hop_count_type, priority, serial_number)

ack_request:	Data Link Layer Acknowledge requested or don't care
domain_address:	the Domain Address of the remote device
hop_count_type:	hop count 7 or standard
priority:	system
serial_number:	the KNX Serial Number of the remote device

A_DomainAddressSerialNumber_Read.con (domain_address, hop_count_type, priority, serial_number)

domain_address:	the Domain Address of the remote device
hop_count_type:	hop count 7 or standard
priority:	system
serial_number:	the KNX Serial Number

3.3.7 A_DomainAddressSerialNumber_Write-Service

The A_DomainAddressSerialNumber_Write.req primitive shall be applied by the user of Application Layer, to modify the Domain Address in a communication partner. The communication partner shall be identified using the unique KNX Serial Number (6 octet) of the device. The Domain Address shall be encoded according the Domain Address format used on the medium of the remote application process either as a 2 octet value (KNX-PL110) or a 6 octet value (KNX-RF).

The local Application Layer shall accept the service request and shall pass it with a T_Data_System-Broadcast.req to the local Transport Layer. The parameter priority, implicitly with value 'system', shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req primitive; the TSDU shall be an A_DomainAddressSerialNumber_Write-PDU.

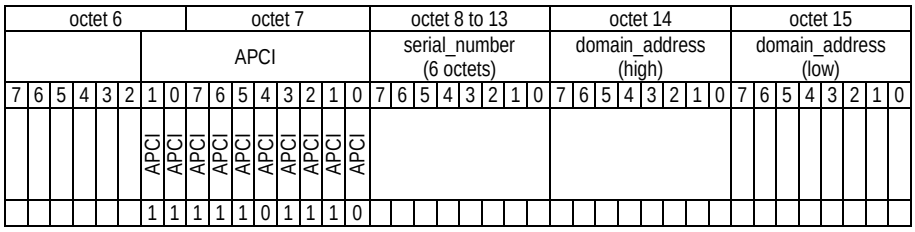


Figure 29 - A_DomainAddressSerialNumber_Write-PDU for a 2 octet DoA format (example)

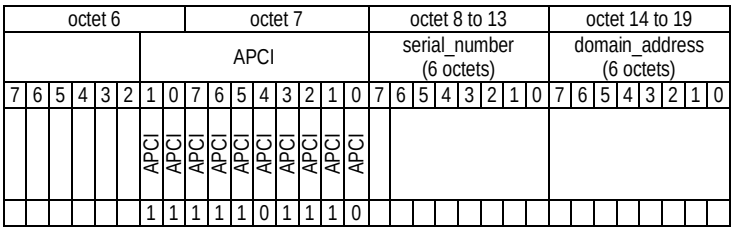


Figure 30 - A_DomainAddressSerialNumber_Write-PDU for a 6 octet DoA format (example)

The remote Application Layer shall map a T_Data_SystemBroadcast.ind primitive with TSDU = A_DomainAddressSerialNumber_Write-PDU to an A_DomainAddressSerialNumber_Write.ind primitive. The argument priority shall be mapped to the corresponding argument priority of the A_DomainAddressSerialNumber_Write.ind primitive.

If the KNX Serial Number contained in the A_DomainAddressSerialNumber_Write-PDU is equal to the KNX Serial Number of the device, then the remote application process shall change its Domain Address to the value contained in the A_DomainAddressSerialNumber_Write-PDU; otherwise, it shall ignore the A_DomainAddressSerialNumber_Write.ind.

A_DomainAddressSerialNumber_Write.req(ack_request, domain_address, hop_count_type, priority, serial_number)

ack_request:	Data Link Layer Acknowledge requested or don't care
domain_address:	the new value of the Domain Address
hop_count_type:	hop count 7 or standard
priority:	system
serial_number:	the KNX Serial Number of the remote device

A_DomainAddressSerialNumber_Write.ind(domain_address, hop_count_type, priority, serial_number)

domain_address:	the new value of the Domain Address
hop_count_type:	hop count 7 or standard
priority:	system, urgent, normal or low priority
serial_number:	the KNX Serial Number of the remote device

3.3.8 A_SystemNetworkParameter_Read

The A_SystemNetworkParameter_Read.req primitive shall be applied by the user of Application Layer of the MaC to check about the configuration of a network parameter.

- On an open communication medium, the service request shall be communicated on point-to-all-points (system broadcast) communication mode.
- On a closed communication medium, the service request shall be communicated on point-to-domain, connectionless (broadcast) communication mode.

A device shall respond to the service if it complies with the conditions specified in the service parameters.

The local Application Layer shall accept the service request.

- If the local medium is an open medium, then it shall pass the service request with a T_Data_SystemBroadcast.req to the local Transport Layer.
- If the local medium is a closed medium, then it shall pass the service request with a T_Data_Broadcast.req to the local Transport Layer.

The TSDU shall in both cases always be an A_SystemNetworkParameter_Read-PDU. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req respectively the T_Data_Broadcast.req primitive; the TSDU shall be an A_SystemNetworkParameter_Read-PDU.

If the remote Application Layer receives a T_Data_SystemBroadcast.ind or a T_Data_Broadcast.ind with TSDU = A_SystemNetworkParameter_Read-PDU, it shall map the service primitive to an A_SystemNetworkParameter_Read.ind primitive. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter priority of the A_SystemNetworkParameter_Read.ind primitive.

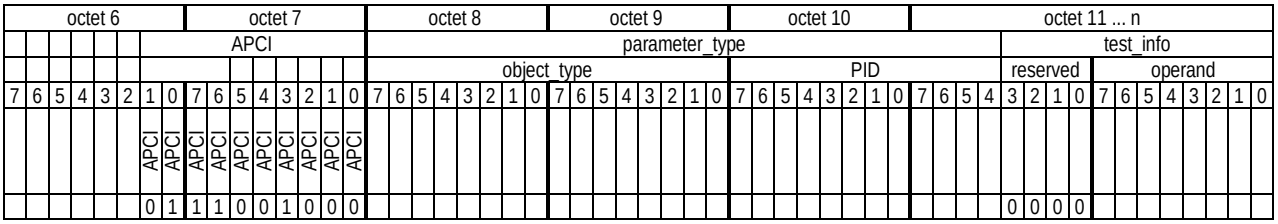


Figure 31 - A_SystemNetworkParameter_Read-PDU (example)

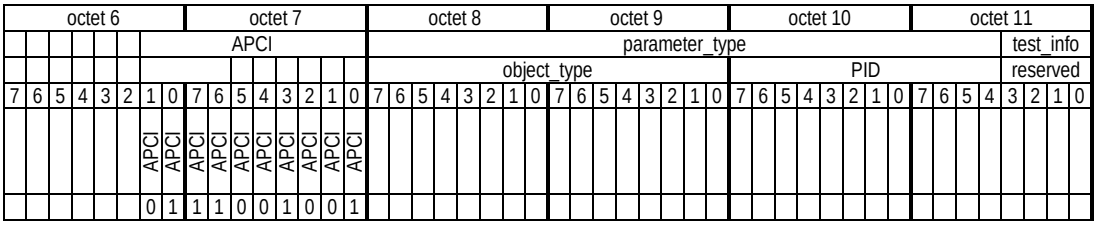
If the remote application process finds the conditions for replying are not fulfilled, this is if it does not support the read network parameter or the check of its investigated parameters against the test information is negative, it shall ignore the service.

If the remote application process accepts the service, it shall respond to the A_SystemNetwork-Parameter_Read.ind primitive with an A_SystemNetworkParameter_Read.res primitive after a random wait time that is per parameter_type specified in [05]. The data in the response shall depend on the network parameter type being read.

The remote Application Layer shall accept the service response.

- If the local medium is an open medium, then it shall pass the service response with a T_Data_SystemBroadcast.req to the local Transport Layer
- If the local medium is a closed medium, then it shall pass the service response with a T_Data_Broadcast.req to the local Transport Layer

The TSDU shall always be an A_SystemNetworkParameter_Response-PDU.



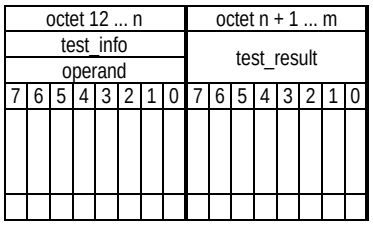


Figure 32 - A_SystemNetworkParameter_Response-PDU (example)

On reception of a T_Data_SystemBroadcast.ind primitive or a T_Data_Broadcast.ind, with TSDU = A_SystemNetworkParameter_Response-PDU the local Application Layer shall map this service to an A_SystemNetworkParameter_Read.Acon primitive. The parameter priority, implicitly with value 'system', shall be mapped to the corresponding parameter priority of the A_SystemNetworkParameter_Read.Acon primitive. The parameter TSAP shall be mapped to the corresponding parameter ASAP of the A_SystemNetworkParameter_Read.Acon primitive.

Error and exception handling

- If the remote application process receives an A_SystemNetworkParameter_Read-PDU with an unknown object_type within parameter_type, this is, object_type is not supported by the remote application, then it shall respond with an A_SystemNetworkParameter_Response-PDU with object_type = FFFFh, PID = FFh and without test_info and test_result fields.
- If the remote application process receives an A_SystemNetworkParameter_Read-PDU with a known object_type but with unknown PID within parameter_type, this is, the requested PID is not supported by the remote application, then it shall respond with an A_SystemNetworkParameter_Response-PDU with object_type = requested object_type, PID = FFh and without test_info and test_result fields.
- The negative response for a negative result of a check of the investigated parameters against the test information is specified per parameter_type either in [05] or [06].

A_SystemNetworkParameter_Read.req(hop_count_type, parameter_type, priority, test_info)

hop_count_type:	hop count 7 or standard
parameter_type:	network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	value against which the resource indicated by parameter_type is tested

A_SystemNetworkParameter_Read.Lcon(hop_count_type, parameter_type, priority, test_info, a_status)

hop_count_type:		hop count 7 or standard
parameter_type:		network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
test_info		value against which the resource indicated by parameter_type is tested
a_status:	ok:	A_NetworkParameter_Read.req sent successfully with T_Data_Broadcast – or with T_Data_SystemBroadcast-service
	not_ok:	transmission of the associated T_Data_Broadcast – or T_DataSystem_Broadcast request frame did not succeed

A_SystemNetworkParameter_Read.ind(hop_count_type, parameter_type, priority, test_info)

hop_count_type:	hop count 7 or standard
parameter_type:	network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	value against which the resource indicated by parameter_type is tested

A_SystemNetworkParameter_Read.res(hop_count_type, parameter_type, priority, test_info, test_result)

hop_count_type:	hop count 7 or standard
parameter_type:	network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	value against which the resource indicated by parameter_type is tested
test_result:	parameter_type dependent response

A_SystemNetworkParameter_Read.Rcon(hop_count_type, parameter_type, priority, test_info, test_result, a_status)

hop_count_type:		hop count 7 or standard
parameter_type:		network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
test_info:		value against which the resource indicated by parameter_type is tested
test_result:		parameter_type dependent response
a_status:	ok:	A_NetworkParameter_Read.res sent successfully with T_Data_Broadcast or T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Broadcast or T_Data_Individual request frame did not succeed

A_SystemNetworkParameter_Read.Acon(ASAP, hop_count_type, parameter_type, priority, test_info, test_result)

ASAP:	local reference of the Service Access Point or individual address
hop_count_type:	hop count 7 or standard
parameter_type:	network parameter type that is verified, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
test_info:	value against which the resource indicated by parameter_type is tested
test_result:	parameter_type dependent response

3.3.9 A_SystemNetworkParameter_Write

The A_SystemNetworkParameter_Write.req primitive shall be applied by the user of Application Layer of the MaC to set network configuration information in one or multiple management servers.

- On an open communication medium, the service request shall be communicated on point-to-all-points (system broadcast) communication mode.
- On a closed communication medium, the service request shall be communicated on point-to-domain, connectionless (broadcast) communication mode.

The local Application Layer shall accept the service request.

- If the local medium is an open medium, then it shall pass the service request with a T_Data_SystemBroadcast.req to the local Transport Layer.
- If the local medium is a closed medium, then it shall pass the service request with a T_Data_Broadcast.req to the local Transport Layer.

The TSDU shall in both cases always be an A_SystemNetworkParameter_Write-PDU. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter of the T_Data_SystemBroadcast.req respectively the T_Data_Broadcast.req primitive; the TSDU shall be an A_SystemNetworkParameter_Write-PDU.

If the remote Application Layer receives a T_Data_SystemBroadcast.ind or a T_Data_Broadcast.ind with TSDU = A_SystemNetworkParameter_Write-PDU, it shall map the service primitive to an A_SystemNetworkParameter_Write.ind primitive. The parameter priority, implicitly with value ‘system’, shall be mapped to the corresponding parameter priority of the A_SystemNetworkParameter_Write.ind primitive.

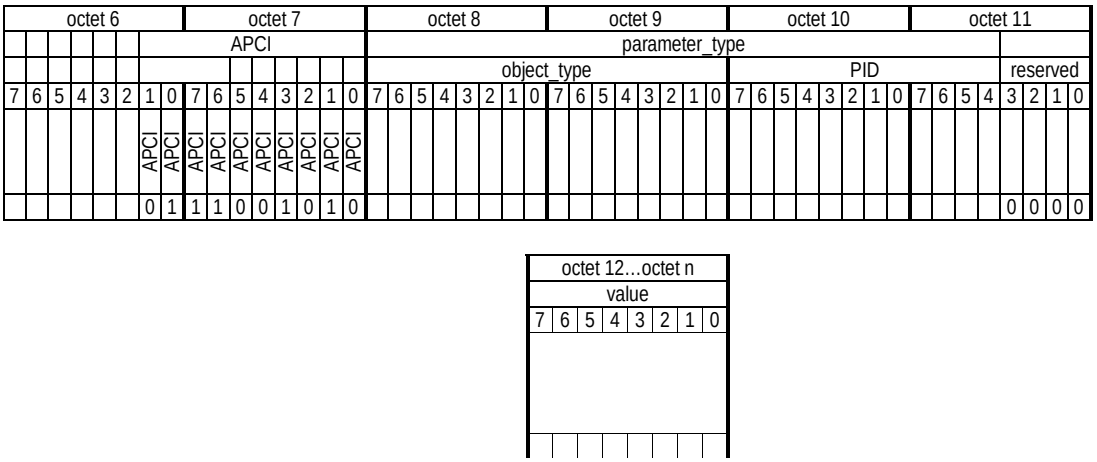


Figure 33 - A_SystemNetworkParameter_Write-PDU (example)

Prior to passing an A_SystemNetworkParameter_Write.Lcon primitive to the local application process, the local Application Layer shall need a T_Data_SystemBroadcast.con or a T_Data_Broadcast.con, depending on the communication mode of the service request, from the local Transport Layer. If the confirmation is positive (t_status = ok), the local Application Layer shall pass a positive A_SystemNetworkParameter_Write.Lcon(a_status = ok) to the local application process. If the confirmation is negative (t_status = not_ok), the local Application Layer shall pass an A_SystemNetworkParameter_Write.Lcon (a_status = not_ok) to the local user indicating that the transmission of the associated Transport Layer specification did not succeed.

NOTE 5 This service is as such not confirmed by the remote partner. The A_SystemNetworkParameter_Write.Lcon is only a local confirmation caused by the local Transport Layer confirmation, basically caused by the Data Link Layer confirmation (ok, not_ok).

A_SystemNetworkParameter_Write.req(hop_count_type, parameter_type, priority, value)

hop_count_type:		hop count 7 or standard
parameter_type:		the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
value:		value to which the network parameter indicated by parameter_type shall be set

A_SystemNetworkParameter_Write.Lcon(hop_count_type, parameter_type, priority, value, a_status)

hop_count_type:		hop count 7 or standard
parameter_type:		the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:		system, urgent, normal or low priority
value:		value to which the network parameter indicated by parameter_type shall be set
a_status:	ok:	A_SystemNetworkParameter_Write.req sent successfully with T_Data_Broadcast – or with T_Data_SystemBroadcast- service
	not_ok:	Transmission of the associated T_Data_Broadcast – or with T_Data_System-Broadcast request frame did not succeed

A_SystemNetworkParameter_Write.ind(hop_count_type, parameter_type, priority, value)

hop_count_type:	hop count 7 or standard
parameter_type:	the network parameter that shall be set, structured as Interface Object Type and Property Identifier
priority:	system, urgent, normal or low priority
value:	value to which the network parameter indicated by parameter_type shall be set

Error and exception handling

- **Unknown parameter_type**

If an unknown parameter_type is received, the receiver shall neglect the A_SystemNetwork-Parameter_Write-PDU without further action.

- **Object Type and PID not accessible through A_SystemNetworkParameter_Write**

The A_SystemNetworkParameter_Write-service is no generic function to access any Property. Instead, the accessibility of a Property through this service is limited to those Properties for which this is indicated in [06].

- **Property specific use of A_SystemNetworkParameter_Write**

The service parameters Object Type, PID and Value are inherited from the access to Data Properties using e.g. A_PropertyValue_Write. Yet, this service may not only be used to simply set the Property Value, but may instead have a use case specific meaning and interpretation in the field “value”, similar to the service A_FunctionProperty_Command. It is for each use of this service per Property specified in Chapter 3/5/1 “Resources” ([04]) or in Chapter 3/5/2 “Management Procedures” ([05]) how the field “value” shall be interpreted.

- **Data consistency**

In general, if over this service data is accessed that is also accessible over additional mechanisms, like over the A_PropertyValue_Read or the A_PropertyValue_Write, via dedicated services like the A_IndividualAddress_Read or the A_IndividualAddress_Write, as a memory mapped Resource or via local access using EMI, the receiver shall take care of consistency between these access modes.

The network – and device Management Procedures (see [05]) that base on this Application Layer service may specify procedure specific error handling.

3.4 Application Layer Services on Point-to-point Connectionless Communication Mode

3.4.1 Introduction

A point-to-point connectionless communication mode shall connect one device with another device. The following services can be applied on the point-to-point connectionless communication mode as well as on the point-to-point connection-oriented communication mode. The following clauses describe the mapping of the services on the point-to-point connectionless communication mode. For using these services on a connection oriented communication mode the T_Data_Connected service of Transport Layer shall be applied instead of the T_Data_Individual service.

- For the connectionless communication mode the ASAP parameter shall be the Individual Address of the communication partner.
- For the connection-oriented communication mode the ASAP parameter shall be the identifier of the communication relationship

The services on the point-to-point connectionless communication mode allow accessing Properties of Interface Objects in the communication partner. See [03] for a description of the Interface Objects and their structure.

3.4.2 Common services

3.4.2.1 A_DeviceDescriptor_Read-service

The A_DeviceDescriptor_Read.req primitive shall be applied by the user of Application Layer to read the Device Descriptor of the remote Management Server. The service shall be confirmed by the remote Management Server containing the Device Descriptor information.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Individual.req primitive, the TSDU shall be an A_Device-Descriptor_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Device-Descriptor_Read-PDU to an A_DeviceDescriptor_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_DeviceDescriptor_Read.ind primitive.

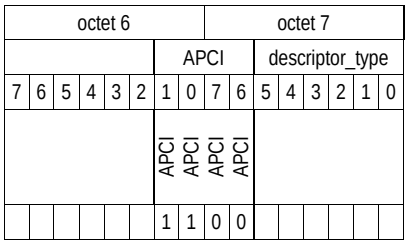


Figure 34 - A_DeviceDescriptor_Read-PDU (example)

The remote Management Server shall respond to the A_DeviceDescriptor_Read.ind primitive with an A_DeviceDescriptor_Read.res primitive containing the Device Descriptor information.

In this, the following shall apply.

On reception of an A_DeviceDescriptor_Read-PDU with descriptor_type = 0 the Management Server shall respond with DD0 if DD0 is supported. If DD0 is not supported by the Management Server then it shall respond with the lowest supported Descriptor Type and corresponding data.

On reception of an A_DeviceDescriptor_Read-PDU with descriptor_type ≠ 0 the Management Server shall respond with the corresponding Descriptor Type and data.

On reception of an A_DeviceDescriptor_Read-PDU with descriptor_type ≠ 0 and this Descriptor Type is not supported by the Management Server, then the Management Server shall respond with the Descriptor Type 3Fh (error) without any data.

This definition is independent of the medium used and whether connection-oriented - or connectionless communication is used.

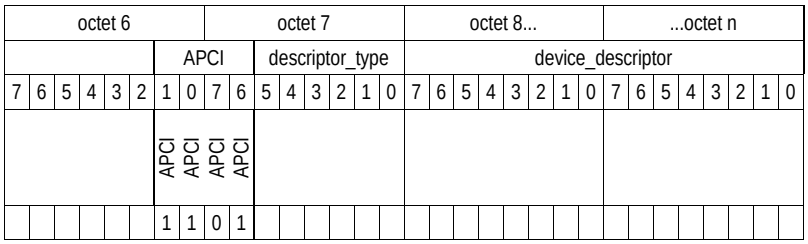


Figure 35 - A_DeviceDescriptor_Response-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Individual.req primitive, the TSDU shall be an A_DeviceDescriptor_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Device-Descriptor_Response-PDU to an A_DeviceDescriptor_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_DeviceDescriptor_Read.Acon primitive. For definition of the Device Descriptors see [04].

A_DeviceDescriptor_Read.req(ack_request, priority, hop_count_type, ASAP, descriptor_type)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
descriptor_type	type of the Device Descriptor

A_DeviceDescriptor_Read.Lcon(ack_request, priority, hop_count_type, ASAP, descriptor_type, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
descriptor_type		type of the Device Descriptor
a_status:	ok:	A_DeviceDescriptor_Read-PDU sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_DeviceDescriptor_Read.ind(priority, hop_count_type, ASAP, descriptor_type)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
descriptor_type	type of the Device Descriptor

A_DeviceDescriptor_Read.res(ack_request, priority, hop_count_type, ASAP, descriptor_type, device_descriptor)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
descriptor_type	type of the Device Descriptor
device_descriptor:	the Device descriptor of the communication controller

A_DeviceDescriptor_Read.Acon(priority, hop_count_type, ASAP, descriptor_type, device_descriptor)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
descriptor_type	type of the Device Descriptor
device_descriptor:	the Device Descriptor of the communication controller

3.4.2.2 A_Restart-service

The following service can be applied on the point-to-point connectionless communication mode as well as on the point-to-point connection-oriented communication mode. The following clause specifies the mapping of A_Restart on the point-to-point connectionless communication mode. For using this service on a connection-oriented communication mode T_Data_Connected of Transport Layer shall be applied instead of T_Data_Individual.

- For the connectionless communication mode the ASAP parameter shall be the Individual Address of the communication partner.
- For the connection-oriented communication mode the ASAP parameter shall be the identifier of the communication relationship.

The local Application Layer user shall apply the A_Restart.req service primitive with service parameter restart_type encoded as follows.

0: To request a Basic Restart of the communication partner.
The A_Restart-PDU shall contain no further data.

1: To request a Master Reset of the communication partner.
The A_Restart-PDU shall contain the fields Erase Code and Channel Number.

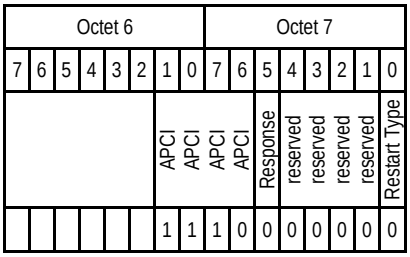


Figure 36 - A_Restart-PDU (example with restart_type = 0)

Octet 6								Octet 7								Octet 8								Octet 9							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	Response	reserved	reserved	reserved	reserved	Restart Type	Erase Code								Channel Number							
						1	1	1	0	0	0	0	0	0	1																

Figure 37 - A_Restart-PDU (example with restart_type = 1)

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req (T_Data_Connected.req) to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Individual.req (T_Data_Connected.req) primitive, the TSDU shall be an A_Restart-PDU.

The remote Application Layer shall map a T_Data_Individual.ind (T_Data_Connected.ind) primitive with TSDU = A_Restart-PDU to an A_Restart.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Restart.ind primitive.

If the remote Application Layer accepts the service and the restart_type indicates a Basic Restart then the remote application process shall not confirm the service.

If the service is not confirmed by the remote application process, then

- for the connectionless communication mode the local Application Layer confirmation shall be caused by the L2-acknowledge of the local Data Link Layer, and
- for the connection-oriented communication mode the local Application Layer confirmation shall be caused by the confirmation of the local Transport Layer.

If the restart_type equals a Master Reset then the remote Application Process shall respond to the A_Restart_ind primitive with an A_Restart.res primitive containing the Error Code and the Process Time. The Process Time shall be encoded as specified in [05] clause “DM_Restart” under “Timing (Management Client and Management Server)”.

Octet 6								Octet 7								Octet 8								Octet 9								Octet 10							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	Response	reserved	reserved	reserved	reserved	Restart Type	Error Code								Process Time																	
						1	1	1	0	1	0	0	0	0	1																										

Figure 38 - A_Restart_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Individual.req (T_Data_Connect.req) to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_Restart_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind (T_Data_Connected.ind) primitive with TSDU = A_Restart_Response-PDU to an A_Restart.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Restart.Acon primitive.

A_Restart.req(ack_request, channel_number, erase_code, priority, hop_count_type, restart_type, ASAP)

ack_request	Data Link Layer acknowledge requested or don't care
channel_number	number of the application channel that shall be reset or 00h
erase_code	indication of Resources that shall be reset prior to resetting the device
priority:	system, urgent, normal or low priority
hop_count_type	hop count 7 or standard
restart_type	Basic Restart or Master Reset
ASAP:	local reference of the service access point or Individual Address

A_Restart.ind(erase_code, channel_number, priority, hop_count_type, ASAP)

channel_number	number of the application channel that shall be reset or 00h
erase_code	indication of Resources that shall be reset prior to resetting the device
priority:	system, urgent, normal or low priority
hop_count_type	hop count 7 or standard
restart_type	Basic Restart or Master Reset
ASAP:	local reference of the service access point or Individual Address

A_Restart.res(error_code, priority, process time, hop_count_type, ASAP)

error_code	indication about success or failure of the Master Reset request
priority:	system, urgent, normal or low priority
process time:	worst case time required by the remote Application Layer user needs for the execution of the requested Master Reset
hop_count_type	hop count 7 or standard
restart_type	Basic Restart or Master Reset
ASAP:	local reference of the service access point or Individual Address

Error and exception handling

The connectionless usage of A_Restart simplifies the error handling of the restart procedure.

In case the local Application Layer user (Management Client) applies the A_Restart (Basic Restart or Master Reset) using the connection-oriented communication mode, it shall take into account the following.

- The Management Server (device) may or may not break down the Transport Layer connection. If the Management Server breaks down the Transport Layer connection, it may or may not send a T_Disconnect-PDU to the Management Server.

The Management Client shall set the value of the reserved fields in octet 7 to 0. The Management Server (device) shall check these fields to be 0; if these fields differ from 0 then it shall ignore the service totally. (No negative response shall be sent.)

If the Management Server (device) receives an A_Restart-PDU with restart_type = Master Reset and with a value of the Erase Code that is reserved or that it does not support, then it shall respond with an A_Restart_Response-PDU with Error Code = "Unsupported Erase Code".

If the Management Server (device) receives an A_Restart-PDU with restart_type = Master Reset and with a value of the Channel Number that is not valid, then it shall respond with an A_Restart_Response-PDU with Error Code = “Invalid Channel Number”. The Channel Number shall be invalid in the following cases.

1. The Channel Number is not 00h for an Erase Code for which the Channel Number shall be 00h according to “Definition of Erase Code and Channel Number” in [05] clause “DM_Restart”.
2. The Channel Number is not 00h but the device does not support application channels.
3. The Channel Number is not 00h and the device does support application Channels but the device does not have an application channel with the requested Channel Number.

3.4.2.3 A_FileStream_InfoReport

The A_FileStream_InfoReport.req primitive shall be applied by the user of Application Layer to send one file block to the remote partner. The file of the File Server shall be identified by the file handle and the file block shall be identified by the file block sequence number. The APCI shall be the APCI_FileStream_InfoReport. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. This service shall not be confirmed.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Individual.req to the local Transport Layer. The parameter ASAP shall be mapped to the corresponding ASAP of the T_Data_Individual.req primitive. The priority shall be set to low. The TSDU shall be an A_FileStream_InfoReport-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_FileStream_InfoReport-PDU to an A_FileStream_InfoReport.ind primitive. The argument TSAP shall be mapped to the corresponding argument ASAP of the A_FileStream_InfoReport.ind primitive.

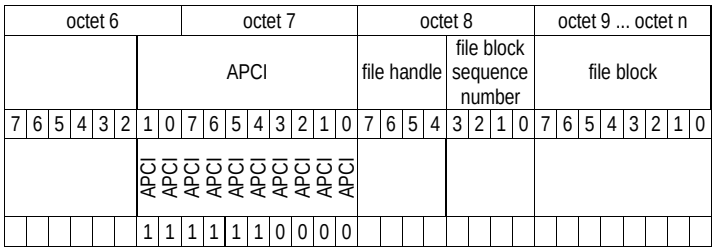


Figure 39 - A_FileStream_InfoReport-PDU (example)

A_FileStream_InfoReport.req(ack_request, ASAP, file_block, file_block_sequence_number, file_handle, hop_count_type)

ack_request:	Data Link Layer Acknowledge requested or don’t care
ASAP:	local reference of the Service Access Point or Individual Address
file_block:	part of the file that shall be transferred to the communication partner
file_block_sequence_number:	rolling counter to denote the relative position of the file block in the stream transfer
file_handle:	reference to the file path of file stream to which the file block belongs
hop_count_type:	hop count 7 or standard

A_FileStream_InfoReport.Lcon(ack_request, ASAP, file_block, file_block_sequence_number, file_handle, hop_count_type, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
file_block:		part of the file that shall be transferred to the communication partner
file_block_sequence_number:		rolling counter to denote the relative position of the file block in the stream transfer
file_handle:		reference to the file path of file stream to which the file block belongs
ASAP:		local reference of the Service Access Point or Individual Address
hop_count_type:		hop count 7 or standard
a_status:	ok:	A_FileStream_InfoReport–PDU is sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_FileStream_InfoReport.ind(ASAP, file_block, file_block_sequence_number, file_handle, hop_count_type)

ASAP:	local reference of the Service Access Point or Individual Address
file_block:	part of the file that shall be transferred to the communication partner
file_block_sequence_number:	rolling counter to denote the relative position of the file block in the stream transfer
file_handle:	reference to the file path of file stream to which the file block belongs
hop_count_type:	hop count 7 or standard

Error and exception handling

There is no Application Layer error handling specified for this service. This service is not confirmed.

3.4.3 Data Property services

3.4.3.1 A_PropertyValue_Read-service

The A_PropertyValue_Read.req primitive shall be applied by the user of Application Layer to read the value of a Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by Transport Layer. The Interface Object of the partner shall be addressed with an object_index and the Property of the Interface Object shall be addressed with a property_id. The nr_of_elem and start_index shall indicate the number of array elements starting with the given start_index in the Property value that the user wants to read. The user of Application Layer in the partner device shall respond with an A_PropertyValue_Read.res, i.e. the service shall be confirmed by the remote application process.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters ASAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyValue_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Property-Value_Read-PDU to an A_PropertyValue_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyValue_Read.ind primitive.

The application process shall respond to the A_PropertyValue_Read.ind primitive with an A_Property-Value_Read.res primitive containing the requested number of elements of the Property value of the Property of the associated Interface Object. If the remote application process has a problem, e.g. object or Property does not exist or the data does not fit in a PDU or the requester has not the required access rights, then the nr_of_elem of the A_PropertyValue_Response-PDU shall be zero and shall contain no data.

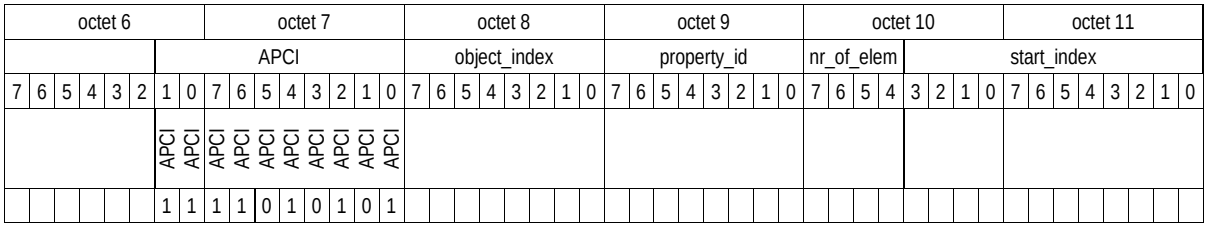


Figure 40 - A_PropertyValue_Read-PDU (example)

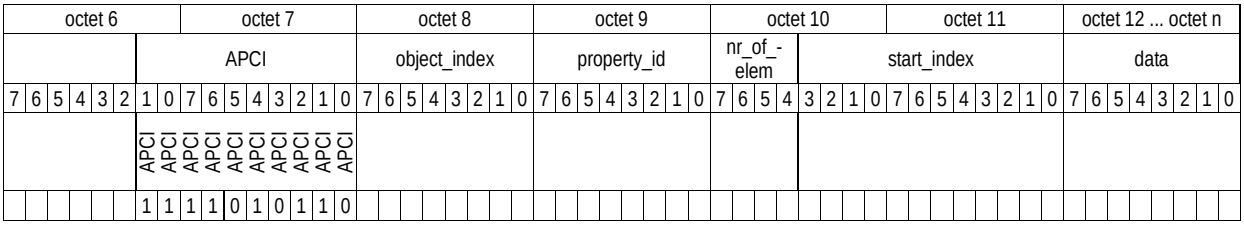


Figure 41 - A_PropertyValue_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU is an A_PropertyValue_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyValue_Response-PDU to an A_PropertyValue_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyValue_Read.Acon primitive.

Error and exception handling

If the remote application process receives an A_PropertyValue_Read-PDU with start_index = 0, this is, reading the current number of elements of the Property Value array, but with a nr_of_elem greater than 1, then it shall respond with an A_PropertyValue_Response-PDU with start_index = 0 and nr_of_elem = 1 and the field data shall contain the current number of elements of the Property array.

A_PropertyValue_Read.req(ack_request, priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the Property_id of the Property of the object
nr_of_elem:	the number of array elements to be read in the Property value
start_index:	the array index of the first array element to be read

A_PropertyValue_Read.Lcon(ack_request, priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_index:		the object_index of the object addressed
property_id:		the property_id of the Property of the object addressed
nr_of_elem:		the number of array elements to be read in the Property value
start_index:		the array index of the first array element to be read
a_status:	ok:	A_PropertyValue_Read sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyValue_Read.ind(priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object addressed
nr_of_elem:	the number of array elements to be read in the Property value
start_index:	the array index of the first array element to be read

A_PropertyValue_Read.res(ack_request, priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object addressed
nr_of_elem:	the number of array elements read in the Property value or zero if problem occurred
start_index:	the array index of the first array element read
data:	the value of the array elements read, or no data, if a problem occurred

A_PropertyValue_Read.Acon(priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object addressed
nr_of_elem:	the number of array elements read in the Property value or zero if problem occurred
start_index:	the array index of the first array element read
data:	the value of the array elements read, or no data, if a problem occurred

3.4.3.2 A_PropertyValue_Write-service

The A_PropertyValue_Write.req primitive shall be applied by the user of Application Layer to modify the value of a Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by Transport Layer. The Interface Object of the partner shall be addressed with the object_index and the Property of the Interface Object shall be addressed with the property_id. The nr_of_elem and start_index shall indicate the number of array elements starting with the given start_index in the Property value that the user wants to write to.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyValue_Write-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyValue_Write-PDU to an A_PropertyValue_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyValue_Write.ind primitive.

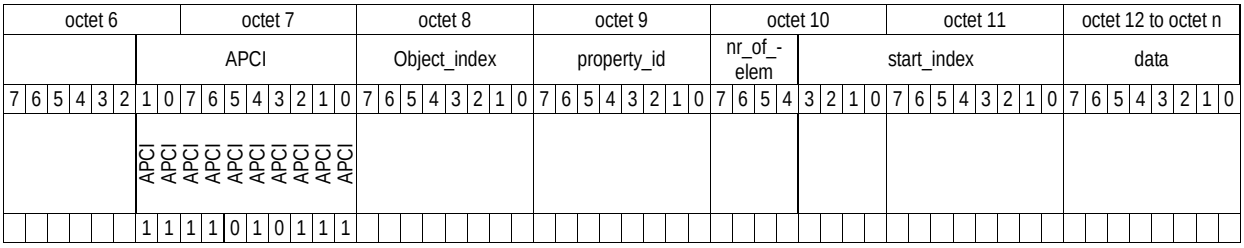


Figure 42 - A_PropertyValue_Write-PDU (example)

The remote application process shall respond to the A_PropertyValue_Write.ind primitive with an A_PropertyValue_Read.res primitive containing the requested number of elements of the Property value of the Property of the associated Interface Object. The value of the Property of the associated Interface Object shall be explicitly read back after writing to it.

Error handling

If the remote application process has a problem, e.g. Interface Object or Property doesn't exist or the requester does not have the required access rights, then the nr_of_elem of the A_PropertyValue_Response-PDU (see Figure 41) shall be zero and shall contain no data.

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyValue_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Property-Value_Response-PDU to an A_PropertyValue_Write.Acon primitive if an A_PropertyValue_Write-PDU is sent before to this communication partner to this Interface Object and Property. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyValue_Read.Acon primitive.

A_PropertyValue_Write.req(ack_request, priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index, data)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_index:		the object_index of the object addressed
property_id:		the property_id of the Property of the object addressed
nr_of_elem:		the number of array elements to be written in the Property value
start_index:		the array index of the first array element to be written
data:		the data to write to the array elements

A_PropertyValue_Write.Lcon(ack_request, priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_index:		the object_index of the object addressed
property_id:		the property_id of the Property of the object addressed
nr_of_elem:		the number of array elements to be written in the Property value
start_index:		the array index of the first array element to be written
data:		the data to write to the array elements
a_status:	ok:	A_PropertyValue_Write sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyValue_Write.ind(priority, hop_count_type, ASAP, object_index, property_id, nr_of_elem, start_index, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object addressed
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
data:	the data to write to the array elements

3.4.3.3 A_PropertyDescription_Read-service

The A_PropertyDescription_Read.req primitive shall be applied by the user of Application Layer, to read the description of the Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by Transport Layer. The Interface Object of the partner shall be addressed with an object_index and the Property of the object shall be addressed with a property_id or with a property_index. The property_index shall be used only if the property_id is zero. The property_index, if evaluated, shall address the Property of the Interface Object with a sequential number, i.e. property_index = 0 shall mean the first Property of the associated Interface Object, property_index = 1 shall mean the second Property. The service shall be confirmed by the remote application process.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyDescription_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Property-Description_Read-PDU to an A_PropertyDescription_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Property-Description_Read.ind primitive.

The application process shall respond to the A_PropertyDescription_Read.ind primitive with an A_PropertyDescription_Read.res primitive containing the description of the Property of the associated Property of the Interface Object addressed. The fields of the description of the Property shall be encoded as defined in [03].

If the property_id in the A_PropertyDescription_Read-PDU is zero, the remote application process shall use the indicated property_index to access the Property description. The property_index in the A_PropertyDescription_Response-PDU shall be the value of the field property_index of the received A_PropertyDescription_Read-PDU

If the property_id in the A_PropertyDescription_Read-PDU is not zero, then the field property_index shall be ignored; the remote application process shall use the indicated property_id to access the Property description. The property_index in the A_PropertyDescription_Response-PDU shall in this case be:

- the correct value of the Property index of the addressed Property, or
- the value of the field property_index of the received A_PropertyDescription_Read-PDU.

For new implementations the property_index shall contain the correct value of the addressed Property.

If the remote application process has a problem, e.g. Interface Object or Property does not exist, then the max_nr_of_elem of the A_PropertyDescription_Response-PDU shall be zero.

The service shall not be confirmed negative for authorization reasons (see A_Authorize_Request-service).

octet 6								octet 7								octet 8								octet 9								octet 10							
								APCI								object_index								Property_id								property_index							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
								APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																								
								1	1	1	1	0	1	1	0	0	0																						

Figure 29 - A_PropertyDescription_Read-PDU (example)

octet 6								octet 7								octet 8								octet 9								octet 10							
								APCI								object_index								property_id								property_index							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
								APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																								
								1	1	1	1	0	1	1	0	0	1																						

octet 11								octet 12								octet 13								octet 14							
w		type										max_nr_of_elem								access							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	r							r	r	r	r													read_level				write_level			
	0							0	0	0	0																				

Figure 43 - A_PropertyDescription_Response-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyDescription_Response-PDU.

Unused fields in the A_PropertyDescription_Response-PDU are marked with ‘r’ in Figure 43 above and shall be set to 0 [3)].

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Property-Description_Response-PDU to an A_PropertyDescription_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyDescription_Read.Acon primitive.

3) Existing implementations may set bit 6 of octet 11 if the Property Value is an array; this is not allowed for new implementations. This bit shall not be evaluated by a management client; instead the contents of the field max_nr_of_elem shall be used.

A_PropertyDescription_Read.req(ack_request, priority, hop_count_type, ASAP, object_index, property_id, property_index)

ack_request:		Data Link Layer Acknowledge requested or don’t care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_index:		the object_index of the object addressed
property_id:		the property_id of the Property of the object
property_index:		sequential Property number

A_PropertyDescription_Read.Lcon(ack_request, priority, hop_count_type, ASAP, object_index, property_id, property_index, a_status)

ack_request:		Data Link Layer Acknowledge requested or don’t care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_index:		the object_index of the object addressed
property_id:		the property_id of the Property of the object
property_index:		sequential Property number
a_status:	ok:	A_PropertyDescription_Read sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyDescription_Read.ind(priority, hop_count_type, ASAP, object_index, property_id, property_index)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object
property_index:	sequential Property number

A_PropertyDescription_Read.res(ack_request, priority, hop_count_type, ASAP, object_index, property_id, property_index, write_enable, type, max_nr_of_elem, access)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object
property_index:	the sequential Property number of the addressed Property [4)]
write_enable:	specifies whether the Property value can be written or not
type:	the Property DataType according to which the Property value is encoded
max_nr_of_elem:	maximum number of elements of the array or zero to indicate a problem
access:	access level to read or write to the Property value

A_PropertyDescription_Read.Acon(priority, hop_count_type, ASAP, object_index, property_id, property_index, write_enable, type, max_nr_of_elem, access)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_index:	the object_index of the object addressed
property_id:	the property_id of the Property of the object
property_index:	the sequential Property number of the addressed Property 4)
write_enable:	specifies whether the Property value can be written or not
type:	the Property DataType according to which the Property value is encoded
max_nr_of_elem:	maximum number of elements of the array or zero to indicate a problem
access:	access level to read or write to the Property value

3.4.4 Link services

3.4.4.1 A_Link_Read-service

The A_Link_Read.req primitive shall be applied by the local Application Layer user to read the links to a given Group Object in a communication partner. The communication partner shall be addressed with a local ASAP that shall me mapped to an Individual Address by the Transport Layer. The Group Object of the communication partner shall be addressed with a Group Object Number. The index in the list of links from which the reading shall start shall be indicated with the start index. The remote Application Layer user in the communication partner shall respond with an A_Link_Read.res containing the requested link information; this is, the service shall be confirmed by the remote Application Layer.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Individual.req primitive, the TSDU shall be an A_Link_Read-PDU.

4) Existing implementations may respond with the value of the field property_index contained in the A_PropertyDescription_Read-PDU. This is not allowed for new implementations.

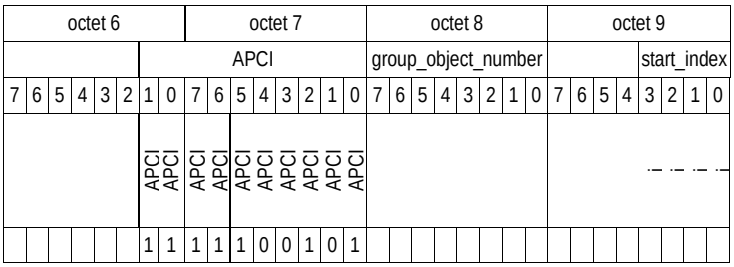


Figure 44 - A_Link_Read-PDU (example)

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Link_-Read-PDU to an A_Link_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Link_Read.ind primitive.

The remote management process shall respond to the A_Link_Read.ind primitive with an A_Link_-Read.res primitive containing the current list of links on the addressed Group Object and the sending Group Address if any. The TSDU shall be an A_Link_Response-PDU.

- If more Group Addresses are assigned to the Group Object than what can be transported in the maximal frame length of the device, then the group_address_list may be limited to the first n Group Addresses association to this Group Object to build a maximal length A_Link_Response-PDU. For devices not supporting the L_Data_Extended frame format, this value n shall be 6.
- If less Group Addresses are assigned to the Group Object than what can be transported in the maximal frame length of the device, then the group_address_list shall only contain the assigned Group Addresses and the A_Link_Response-PDU shall be stopped after the last linked Group Address. The A_Link_Response-PDU may thus have a variable length.
- If the remote management process has a problem, e.g. addressed Group Object does not exist or no Group Addresses are assigned to the Group Object from the request start index, then it shall it shall respond with a negative response.

Positive response

sending_address:	1 to 15, 0 means none.
start_index:	1 to 15
group_address_list:	The A_Link_Response-PDU can contain 0 [5)] to 6 Group Addresses in the ASDU; each Group Address shall occupy 2 octets.

5) "0" is the typical response in case no association exists for the considered Group Object.

Negative response

sending_address: 1 to 15, 0 means none

start_index: 0 [6)]

group_address_list: Group Addresses (each on 16 bit) : none

octet 6						octet 7								octet 8								octet 9								octet 10								...								octet n ≤ 21							
						APCI								group_object_number								sending_-address				start_-address				group_address_list																							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI									S	S	S	S	–	–	–	–																							
						1	1	1	1	1	0	0	1	1	0																																						

Figure 45 - A_Link_Response-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Individual.req to the remote Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive; the TSDU shall be an A_Link_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind primitive with a TSDU = A_Link_-Response-PDU to an A_Link_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Link_Read.res primitive.

A_Link_Read.req (ASAP, group_object_number, priority, start_index)

ASAP:	local reference of the Service Access Point or Individual Address.
group_object_number:	local index of the intended Group Object. The counting shall start with 1.
priority:	system, urgent, normal or low priority
start_index	index of first GA from which the GAs assigned to the address GO in the communication partner shall be read. The counting shall start with 1. The range shall be 1 to 15.

A_Link_Read.res (ASAP, group_address_list, group_object_number, priority, sending_address, start_index)

ASAP:	local reference of the Service Access Point or Individual Address.
group_address_list:	list of Group Addresses connected to the Group Object; 0 to 6 [7)]Group Addresses can be contained, each shall occupy 2 octets.
group_object_number:	local index of the addressed Group Object. The counting shall start with 1.
priority:	system, urgent, normal or low priority
sending_address:	index of the sending address in the list of GAs attached to the Group Object. The counting shall start with 1.
start_index:	index of first transmitted GA in the list of GAs attached to the Group Object. The counting shall start with 1.

6) Sufficient to match negative response and request, as long as response is awaited before sending any other request to same device.

7) This list is limited to the 6 first Group Addresses (for a given Group Object) starting at the start_index. It is expected however that devices with more than 6 links to a single Group Object will use more powerful mechanisms.

3.4.4.2 A_Link_Write-service

The A_Link_Write.req primitive shall be applied by the user of Application Layer to add or remove a single Group Address to or from a given Group Object in a communication partner. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. The Group Object of the communication partner shall be addressed with a Group Object number. It shall indicate the required action through the argument flags. In case a Group Address is added to a Group Object, the argument flags shall also indicate whether the added Group Address shall be the sending Group Address for the referred Group Object or not.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Individual.req primitive, the TSDU shall be an A_Write_Link-PDU.

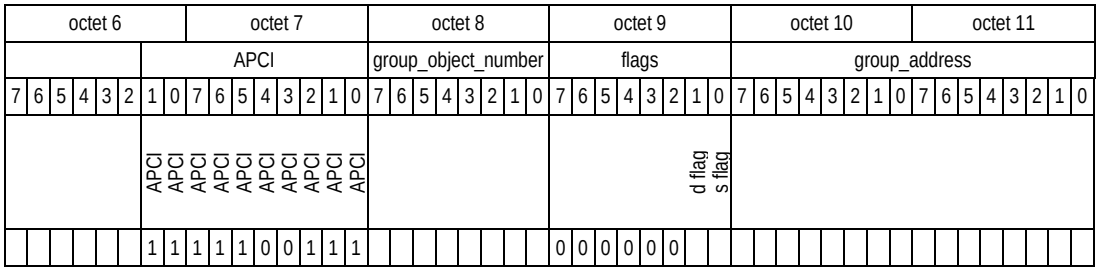


Figure 46 - A_Link_Write-PDU (example)

Flags

d

0: add: Add the contained Group Address to the list of Group Addresses assigned to the referred Group Object.

1: delete: Remove the contained Group Address from the list of Group Addresses assigned to the referred Group Object.

s

The flag s shall only be interpreted in case the flag d equals 0; in case the flag d equals 1, the value of the flag s shall be don't care.

0: not sending: The contained Group Address shall not be the sending Group Address for the referred Group Object.

1: sending: The contained Group Address shall be the sending Group Address for the referred Group Object.

The reception of an A_Link_Write-PDU on an existing association with a different value of the s-flag shall result in an update of the table entry with the new s-flag value.

A Group Address set as "sending" shall take precedence on other previous sending Group Address.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_Link_-Write-PDU to an A_Link_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Link_Write.ind primitive.

The remote management process shall update the link information of the addressed Group Object according the contained parameters. It shall respond to the A_Link_Write.ind primitive with an A_Link_Write.res primitive containing the resulting current list of links on the addressed Group Object and the indication of the sending address if any. The TSDU shall be an A_Link_Response-PDU.

Error and exception handling

- If the remote management process has a problem, e.g. addressed Group Object does not exist or the contained Group Address is not assigned to the addressed Group Object, than it shall respond with a negative response in the A_Link_Response-PDU.
- If after the evaluation of the request more than six Group Addresses are assigned to the addressed Group Object, the remote management process shall only send one single A_Link_Response-PDU containing the six first Group Addresses, thus possibly not containing the Group Address contained in the request.

A_Link_Write.req (ASAP, flags, group_address, group_object_number, priority)

ASAP:	local reference of the Service Access Point
flags:	action to be done (delete/add); sending group address flag
group_address:	Group Address corresponding to the link to add or delete
group_object_number:	local index of the intended Group Object
priority:	system, urgent, normal or low priority

3.4.5 Function Property services

3.4.5.1 A_FunctionPropertyCommand-Service

The A_FunctionPropertyCommand.req primitive shall be applied by the user of Application Layer to call a Function Property of an Interface Object in a communication partner. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the local Transport Layer. The Interface Object of the partner shall be addressed with the object_index and the Function Property of the Interface Object shall be addressed with the property_id.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req or a T_Data_Connected.req (depending whether the service is called connection-oriented or connectionless) to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req or the T_Data_Connected.req primitive; the TSDU shall be an A_FunctionPropertyCommand-PDU.

The remote Application Layer shall map a T_Data_Individual.ind or a T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyCommand-PDU to an A_FunctionPropertyCommand.ind primitive. The parameter comm_mode shall be set to “connection_oriented” in case of a T_Data_Connected.ind and to “connectionless” in case of a T_Data_Individual.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_FunctionPropertyCommand.ind primitive.

The remote application shall check whether the Property Datatype of the Property addressed by this A_FunctionPropertyCommand.ind is PDT_Function. If this is the case it shall call the function and pass the input parameters to it, otherwise the error handling shall apply; see clause 3.4.5.3.

The remote application process shall respond to the A_FunctionPropertyCommand.ind primitive with an A_FunctionPropertyCommand.res primitive containing the return_code and the function specific output data.

octet 6								octet 7								octet 8								octet 9								octet 10 to octet n							
						APCI										object_index								property_id								data							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																										
						1	0	1	1	0	0	0	1	1	1																										

Figure 47 – A_FunctionPropertyCommand-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Individual.req or a T_Data_Connected.req (depending whether the service shall be sent connection-oriented or connectionless) to the local Transport Layer. The TSDU shall be an A_FunctionPropertyState_-Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind or T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyState_Response-PDU to an A_FunctionPropertyCommand.Acon primitive.

A_FunctionPropertyCommand.req(ack_request, ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id,)

ack_request:	Data Link Layer Acknowledge requested or don't care
ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection-oriented or connectionless
data:	input data to the function
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object

A_FunctionPropertyCommand.Lcon(ack_request, ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
ASAP:		local reference of the Service Access Point or Individual Address
comm_mode:		connection-oriented or connectionless
data:		input data to the function
hop_count_type:		hop count 7 or standard
object_index:		the object_index of the addressed Interface Object
priority:		system, urgent, normal or low priority
property_id:		the property_id of the Property of the addressed Interface Object
a_status:	ok:	A_FunctionPropertyCommand-PDU sent successfully with T_Data_Individual or T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Individual or T_Data_Connected request frame did not succeed

A_FunctionPropertyCommand.ind(ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id)

ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection-oriented or connectionless
data:	input data to the function
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object

A_FunctionPropertyCommand.res(ack_request, ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id, return_code)

ack_request:	Data Link Layer Acknowledge requested or don't care
ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection-oriented or connectionless
data:	output data from the function
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object
return_code	error code returned by the function

A_FunctionPropertyCommand.Acon(ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id, return_code)

ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection-oriented or connectionless
data:	output data from the function
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object
return_code	error code returned by the function

3.4.5.2 A_FunctionPropertyState_Read-service

The A_FunctionPropertyState_Read.req primitive shall be applied by the user of Application Layer to read the state of a Function Property of an Interface Object in a remote device. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the local Transport Layer. The Interface Object of the remote device shall be addressed with an object_index and the Property of the Interface Object shall be addressed with a property_id. The user of the Application Layer in the remote device shall respond with an A_FunctionPropertyState_Read.res, this is, the service shall be confirmed by the remote application process.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req or a T_Data_Connected.req (depending whether the service is sent connection-oriented or connectionless) to the local Transport Layer. The TSDU shall be an A_FunctionPropertyState_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive or a T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyState_Read-PDU to an A_FunctionPropertyState_Read.ind primitive.

The remote application shall check whether the Property Datatype of the Property addressed by this A_FunctionPropertyState_Read.ind is PDT_Function. If this is the case it shall call the function and pass the input parameters to it, otherwise the error handling shall apply; see clause 3.4.5.3.

The application process shall respond to the A_FunctionPropertyState_Read.ind primitive with an A_FunctionPropertyState_Read.res primitive containing the function specific return_code and the function specific output data.

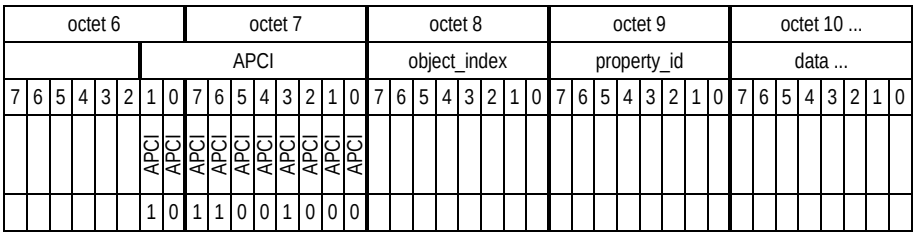


Figure 48 - A_FunctionPropertyState_Read-PDU (example)

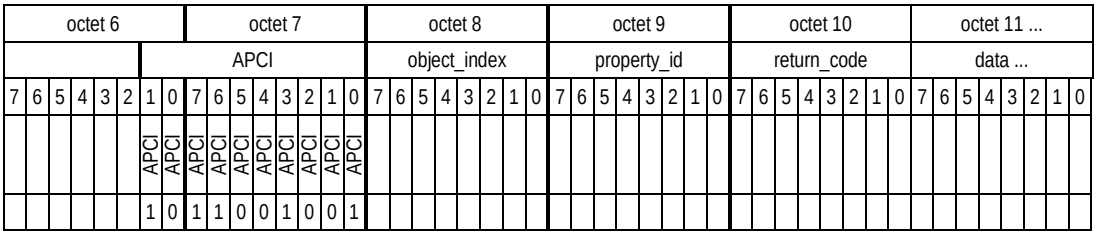


Figure 49 - A_FunctionPropertyState_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Individual.req or a T_Data_Connected.req (depending whether the service shall be sent connection-oriented or connectionless) to the local Transport Layer. The TSDU shall be an A_FunctionPropertyState_-Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind or T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyState_Response-PDU to an A_FunctionPropertyState_Read.Acon primitive.

A_FunctionPropertyState_Read.req(ASAP, ack_request, comm_mode, data, hop_count_type, object_index, priority, property_id)

ASAP:	local reference of the Service Access Point or Individual Address
ack_request:	Data Link Layer Acknowledge requested or don’t care
comm_mode:	connection_oriented or connectionless
data:	input data to the function for reading
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the Interface Object

A_FunctionPropertyState_Read.Lcon(ack_request, ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
ASAP:		local reference of the Service Access Point or Individual Address
comm_mode:		connection_oriented or connectionless
data		input data to the function for reading
hop_count_type:		hop count 7 or standard
object_index:		the object_index of the addressed Interface Object
priority:		system, urgent, normal or low priority
property_id:		the property_id of the Property of the addressed Interface Object
a_status:	ok:	A_FunctionPropertyState_Read-PDU sent successfully with T_Data_Individual or T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Individual or T_Data_Connected request frame did not succeed

A_FunctionPropertyState_Read.ind(ASAP, comm_mode, data, hop_count_type, object_index, priority property_id)

ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection_oriented or connectionless
data	input data to the function for reading
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object

A_FunctionPropertyState_Read.res(ack_request, ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id, return_code)

ack_request:	Data Link Layer Acknowledge requested or don't care
ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection_oriented or connectionless
data:	output data from the function
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object
return_code	error code returned by the function

A_FunctionPropertyState_Read.Acon(ASAP, comm_mode, data, hop_count_type, object_index, priority, property_id, return_code)

ASAP:	local reference of the Service Access Point or Individual Address
comm_mode:	connection_oriented or connectionless
data:	output data from the function
hop_count_type:	hop count 7 or standard
object_index:	the object_index of the addressed Interface Object
priority:	system, urgent, normal or low priority
property_id:	the property_id of the Property of the addressed Interface Object
return_code	error code returned by the function

3.4.5.3 Error and exception handling for Function Property services

If the Interface Object Property accessed by A_FunctionPropertyCommand-PDU or by A_FunctionPropertyState_Read-PDU is not of the Property Datatype PDT_Function, the remote application shall respond with a A_FunctionPropertyState_Response-PDU without the field return_code (this is, the returned PDU shall not contain the field return_code) and without the field data (this is, the returned PDU shall not contain the field data).

In case the remote application is able to successfully call a Function Property, then the Function Property shall deliver a return_code in the field return_code of the PDU. The following rules shall apply for all functions:

- Return_code = 00h: function successfully executed, this is the return code 00h shall be the positive result of the function;
- Return_code ≠ 00h: error.

Error codes are defined in a function specific way.

In case an Interface Object Property of Property Datatype PDT_Function is accessed via the A_PropertyValue_Read-service or the A_PropertyValue_Write-service, the Application Layer shall respond with an A_PropertyValue_Response-PDU with the standard error handling for Data Properties (this is the field nr_of_elem shall be zero and there shall be no data field).

In case an Interface Object Property of Property Datatype PDT_Function is accessed via the A_PropertyDescription_Read, the Application Layer shall respond with an A_PropertyDescription_Response-PDU with type = PDT_Function and max_nr_of_elem = 1; read_level and write_level can have any value.

3.5 Application Layer Services on Point-to-point Connection-Oriented Communication Mode

3.5.1 Introduction

A Point-to-point connection-oriented communication mode shall connect one device with another device. The following services can be applied on point-to-point connection-oriented communication modes if the connection is established (see Transport Layer state machine). Due to the behaviour of the Transport Layer state machine, the user of the Application Layer has to take into account that the connection may be released by the remote communication partner or by an error detected in the communication protocol. Therefore a T_Disconnect.ind primitive may occur at any time, i.e. also if the user of Application Layer is waiting for a confirmation from the Application Layer. The transport layer services T_Connect.ind and T_Disconnect.ind are mapped transparently to A_Connect.ind and A_Disconnect.ind service and passed to the user of Application Layer.

The Application Layer also provides an access protection mechanism on the point-to-point connection-oriented communication mode by an authorization procedure. This procedure is described in clause 3.5.7 "A_Authorize_Request-service".

3.5.2 A_ADC_Read-service

The A_ADC_Read.req primitive shall be applied by the user of Application Layer to read the value of the AD-converter of the communication partner. The service shall be confirmed by the remote application process containing the value of the converter.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_ADC_-Read-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_ADC_-Read-PDU to an A_ADC_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_ADC_Read.ind primitive.

The A_ADC_Read-PDU shall contain the channel number of the AD-converter and the number of consecutive read operations to the AD-converter.

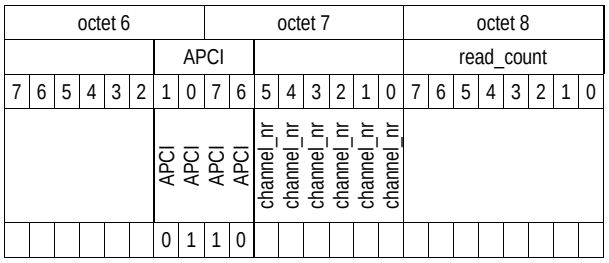


Figure 50 - A_ADC_Read-PDU (example)

The application process shall respond to the A_ADC_Read.ind primitive with an A_ADC_Read.res primitive containing the value of the AD-converter computed by the summation of the consecutive CPU accesses. If the remote application process has a problem, e.g. overflow when computing the summation, or wrong channel number, then the read_count of the A_ADC_Response-PDU shall be zero.

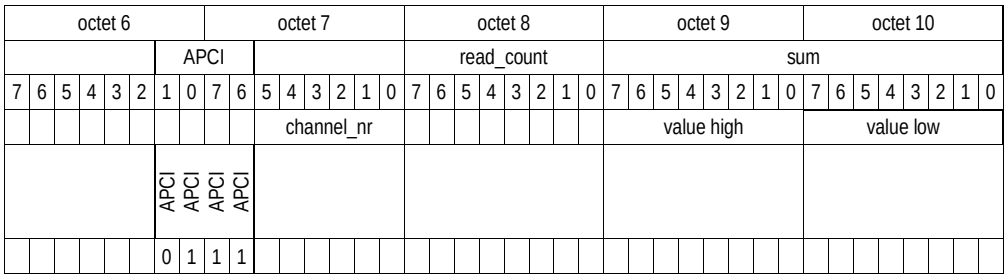


Figure 51 - A_ADC_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_ADC_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_ADC_Response-PDU to an A_ADC_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_ADC_Read.Acon primitive.

A_ADC_Read.req(ack_request, priority, hop_count_type, ASAP, channel_nr, read_count)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
channel_nr:	The channel_nr of the AD-converter
read_count:	number of desired consecutive CPU accesses to the AD-converter

A_ADC_Read.Lcon(ack_request, priority, hop_count_type, ASAP, channel_nr, read_count, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
channel_nr:		The channel_nr of the AD-converter
read_count:		number of desired consecutive CPU accesses to the AD-converter
a_status:	ok:	A_ADC_Read sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_ADC_Read.ind(priority, hop_count_type, ASAP, channel_nr, read_count)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
channel_nr:	The channel_nr of the AD-converter
read_count:	number of desired consecutive CPU accesses to the AD-converter

A_ADC_Read.res(ack_request, priority, hop_count_type, ASAP, channel_nr, read_count, sum)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
channel_nr:	The channel_nr of the AD-converter
read_count:	number of CPU accesses executed to the AD-converter or zero to indicate a problem
sum:	sum of AD-converter values

A_ADC_Read.Acon(priority, hop_count_type, ASAP, channel_nr, read_count, sum)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
channel_nr:	The channel_nr of the AD-converter
read_count:	number of CPU accesses executed to the AD-converter or zero to indicate a problem
sum:	sum of AD-converter values

3.5.3 A_Memory_Read-service

The A_Memory_Read.req primitive shall be applied by the user of Application Layer, to read between 1 and 63 octets in the address space of the remote communication controller. The parameter memory_address shall specify the 16 bit start address and number shall contain the number of octets to be read beginning with the start address to increasing addresses. The service shall be confirmed by the remote application process with the contents of the address space.

The local Application Layer shall accept the service request and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Memory_Read-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Memory_Read-PDU to an A_Memory_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Memory_Read.ind primitive.

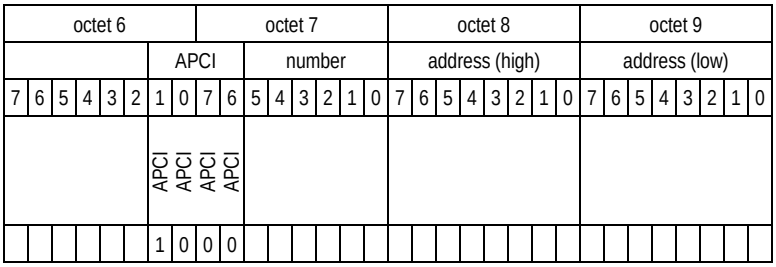


Figure 52 - A_Memory_Read-PDU (example)

The remote application process shall respond to the A_Memory_Read.ind primitive with an A_Memory_Read.res primitive containing the number of octets read beginning with the start address to increasing addresses. If the remote application process has a problem, e.g. address space unreachable or protected or an illegal number of octets are requested, then the parameter number of the A_Memory_Response-PDU shall be zero and shall contain no data.

octet 6								octet 7								octet 8								octet 9								octet 10 ... n							
						APCI			number							address (high)								address (low)								data							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI																																
						1	0	0	1																																

Figure 53 - A_Memory_Response-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_Connected.req to the local Transport Layer. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Memory_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Memory_Response-PDU to an A_Memory_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Memory_Read.Acon primitive.

A_Memory_Read.req(ack_request, priority, hop_count_type, ASAP, number, memory_address)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be read beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address

A_Memory_Read.Lcon(ack_request, priority, hop_count_type, ASAP, number, memory_address, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be read beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address
a_status:	ok:	A_Memory_Read sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_Memory_Read.ind(priority, hop_count_type, ASAP, number, memory_address)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be read beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address

A_Memory_Read.res(ack_request, priority, hop_count_type, ASAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets read beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read

Error handling

If data are to be read from a protected area or from any logical address that is not associated to physical memory then no data shall be returned to indicate an error [8]. The same shall apply if only part of the memory to be read is protected or physically existing.

In addition, the remote Application Layer shall ignore the A_Memory_Read.ind if the value of the parameter "number" is greater than Maximum APDU Length – 3 [9].

3.5.4 A_Memory_Write-service

The A_Memory_Write.req primitive shall applied by the user of Application Layer, to write between 1 octet and 63 octets in the address space of the remote communication controller. The parameter memory_address shall specify the 16 bit start address and the parameter number shall contain the number of octets to be written beginning with the start address to increasing addresses.

The service shall be a confirmed service if Verify Mode is active, otherwise it shall be an acknowledged service.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Memory_Write-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Memory_-Write-PDU to an A_Memory_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments TSAP and priority of the A_Memory_Write.ind primitive.

8) Existing devices may have a different error handling.

9) The maximal APDU length that shall be supported is specified in the specification of PID_MAX_APDULENGTH in [04].

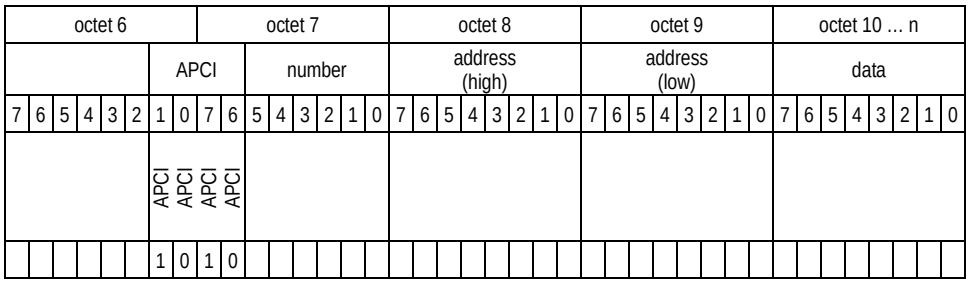


Figure 54 - A_Memory_Write-PDU (example)

With inactive Verify Mode the remote application process shall not respond. Instead the local Application Layer shall map a T_Data_Connected.con primitive to an A_Memory_Write.Lcon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Memory_Write.Lcon primitive; number, memory_address and data shall be don't care.

With active Verify Mode the remote application process shall respond to the A_Memory_Write.ind primitive with an A_Memory_Write.res primitive containing the requested number of octets of the associated memory area. The value of the associated memory area shall be explicitly read back after writing to it. If the remote application process has a problem, e.g. memory area unreachable or protected or an illegal number of octets are requested, then the parameter number shall be zero and shall contain no data.

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Memory_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Memory_-Response-PDU to an A_Memory_Write.Acon primitive if an A_Memory_Write-PDU has been sent before over this connection. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Memory_Write.Acon primitive.

A_Memory_Write.req(ack_request, priority, hop_count_type, TSAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be written beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
data:	the octet(s) to be written

A_Memory_Write.Lcon(ack_request, priority, hop_count_type, ASAP, number, memory_address, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be written beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address
data:		the octet(s) to be written or no data
a_status:	ok:	A_Memory_Write sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_Memory_Write.ind(priority, hop_count_type, TSAP, number, memory_address, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be written beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
data:	the octet(s) to be written

A_Memory_Write.res(ack_request, priority, hop_count_type, ASAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets written beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

A_Memory_Write.Acon(priority, hop_count_type, ASAP, number, memory_address, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets written beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

Error Handling

If data are to be written to a protected area from any logical address that is not associated to physical memory then the service indication shall be ignored. In any case physical memory shall not be addressable via different logical addresses. If only a part of the addressed memory is protected or does not exist, then the complete write operation shall fail.

In addition, the remote Application Layer shall ignore the A_Memory_Write.ind if the value of the parameter “number” is greater than Maximum APDU Length – 3 [10].

If Verify Mode is active, then in case of a failed write operation no data shall be returned to indicate an error [11].

3.5.5 A_MemoryBit_Write-service

The A_MemoryBit_Write.req primitive shall be applied by the user of Application Layer, to modify between 1 bit and 40 bits in a contiguous block of up to 5 octets in the address space of the remote communication controller. The parameter memory_address shall specify the 16 bit start address and number shall contain the number of octets to be modified beginning with the start address to increasing addresses. The A_MemoryBit_Write shall allow to

- set individual bits of the contiguous block to zero, and
- set individual bits of the contiguous block to one, and
- leave individual bits of the contiguous block unmodified, and
- invert individual bits of the contiguous block,

using the parameters and_data and xor_data. Both parameters shall have the same number of octets as the contiguous block indicated in the parameter number. The resulting value for each individual bit in the contiguous block shall be computed using the two associated bits of and_data and xor_data with the following function (see Figure 55):

result_bit(i) = (and_data_bit(i) AND block_bit(i)) XOR xor_data_bit(i)

and_data_bit(i)	xor_data_bit(i)	result_bit(i)
0	0	0
0	1	1
1	0	block_bit(i)
1	1	NOT block_bit(i)

Figure 55 - Function Table for A_MemoryBit_Write-services (example)

The service shall be a confirmed service if Verify Mode is active, otherwise it shall be an acknowledged service.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_MemoryBit_-Write-PDU.

With inactive Verify Mode the remote application process shall not respond. Instead the local Application Layer shall map a T_Data_Connected.con primitive to an A_MemoryBit_Write.Lcon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_MemoryBit_Write.Lcon primitive; number, memory_address and data shall be don't care.

[10] The maximal APDU length that shall be supported is specified in the specification of PID_MAX_APDULENGTH in [04].

[11] Existing devices may have a different error-handling.

With active Verify Mode the remote application process shall respond to the A_MemoryBit_Write.ind primitive with an A_MemoryBit_Write.res primitive containing the requested number of octets of the associated memory area. The value of the associated memory area shall be explicitly read back after writing to it. If the remote application process has a problem, e.g. memory area unreachable or protected or an illegal number of octets are requested, then the parameter number shall be zero and shall contain no data.

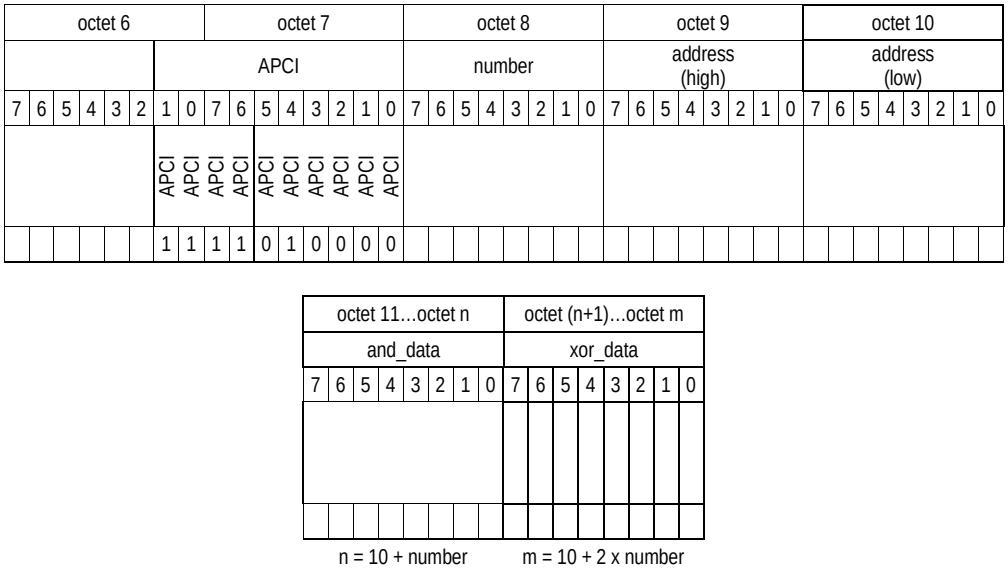


Figure 56 - A_MemoryBit_Write-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU is an A_Memory_-Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Memory_-Response-PDU to an A_MemoryBit_Write.Acon primitive if an A_MemoryBit_Write-PDU is sent before over this connection. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_MemoryBit_Write.Acon primitive.

A_MemoryBit_Write.req(ack_request, priority, hop_count_type, ASAP, number, memory_address, and_data, xor_data)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be modified beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
and_data:	see Figure 55
xor_data:	see Figure 55

A_MemoryBit_Write.Lcon(ack_request, priority, hop_count_type, ASAP, number, memory_address, and_data, xor_data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be modified beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address
and_data:		see Figure 55
xor_data:		see Figure 55
a_status:	ok:	A_MemoryBit_Write sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_MemoryBit_Write.ind(priority, hop_count_type, ASAP, number, memory_address, and_data, xor_data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be modified beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
and_data:	see Figure 55
xor_data:	see Figure 55

A_MemoryBit_Write.res(ASAP, priority, number, memory_address, data)

ASAP:	local reference of the Service Access Point
priority:	system, urgent, normal or low priority
number:	number of octets modified beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

Error handling

If data are to be written to a protected area or from any logical address that is not associated to physical memory then the service indication shall be ignored. In any case physical memory shall not be addressable via different logical addresses. If only a part of the addressed memory is protected or does not exist, then the complete write operation shall fail. If Verify Mode is active, then in case of a failed write operation no data shall be returned to indicate an error [12)].

[12)] Existing devices may have a different error-handling.

3.5.6 A_UserData

3.5.6.1 Definition

The A_UserData-service shall be used for Application Device Management. The Application Device Management is that part of the Device Management that is implemented in the application.

The Application Device Management uses a logical address-space of 1 Mb. The mapping from the logical address space to the physical address and vice versa is task of the Application Device Management.

The Application Device Management may also influence directly or indirectly the application program. But this has to be defined individually for each device.

All services provided by the Application Device Management are based on the A_USER_DATA-messages and the corresponding services.

3.5.6.2 A_UserMemory_Read-service

The A_UserMemory_Read.req primitive shall be applied by the user of Application Layer, to read between 1 octet and 15 octets in the address space of the remote application controller. The parameter memory_address shall specify the 20 bit start address (4 bit address extension + 8 bit address high + 8 bit address low) and the parameter number shall contain the number of octets to be read beginning with the start address to increasing addresses. The service shall be confirmed by the remote application process with the contents of the address space.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_UserMemory_Read-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_User-Memory_Read-PDU to an A_UserMemory_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemory_Read.ind primitive.

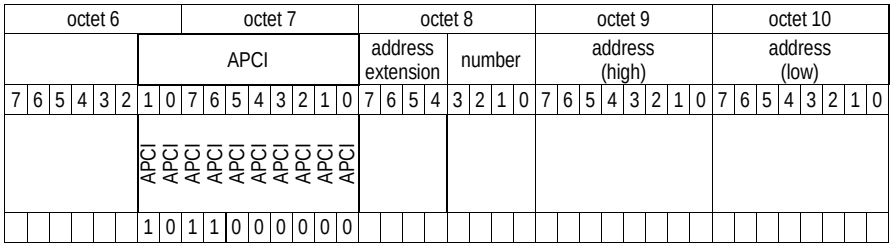


Figure 57 - A_UserMemory_Read-PDU (example)

The remote application process shall respond to the A_UserMemory_Read.ind primitive with an A_UserMemory_Read.res primitive containing the number of octets read beginning with the start address to increasing addresses. If the remote application process has a problem, e.g. address space unreachable or protected or an illegal number of octets are requested, then the parameter number of the A_UserMemory_Response-PDU shall be zero and shall contain no data.

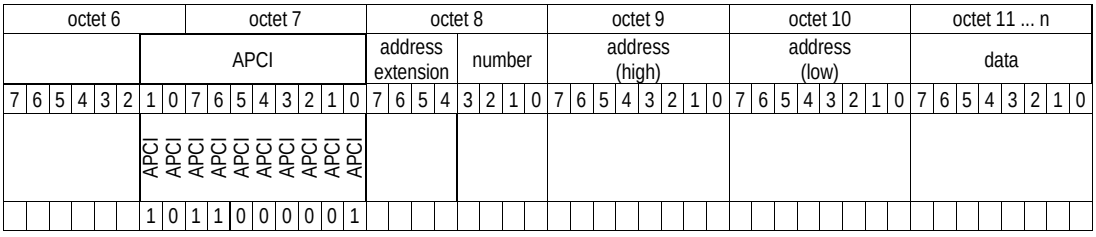


Figure 58 - A_UserMemory_Response-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_User-Memory_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_UserMemory_-Response-PDU to an A_UserMemory_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemory_Read.Acon primitive.

A_UserMemory_Read.req(ack_request, priority, hop_count_type, ASAP, number, memory_address)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be read beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address

A_UserMemory_Read.Lcon(ack_request, priority, hop_count_type, ASAP, number, memory_address, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be read beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address
a_status:	ok:	A_UserMemory_Read sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_UserMemory_Read.ind(priority, hop_count_type, ASAP, number, memory_address)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be read beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address

A_UserMemory_Read.res(ack_request, priority, hop_count_type, ASAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets read beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read

A_UserMemory_Read.Acon(priority, hop_count_type, ASAP, number, memory_address, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets read beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read

Error handling

If data are to be read from a protected area or from any logical address that is not associated to physical memory then no data shall be returned to indicate an error. The same shall apply if only part of the memory to be read is protected or physically existing.

In addition, the remote Application Layer shall ignore the A_UserMemory_Read.ind if the value of the parameter "number" is greater than Maximum APDU Length – 4 [13)].

3.5.6.3 A_UserMemory_Write-service

The A_UserMemory_Write.req primitive shall be applied by the user of Application Layer, to write between 1 octet and 15 octets in the address space of the remote application controller. The parameter memory_address shall specify the 20 bit start address (4 bit address extension + 8 bit address high + 8 bit address low) and the parameter number shall contain the number of octets to be written beginning with the start address to increasing addresses.

The service shall be a confirmed service if Verify Mode is active, otherwise it shall be an acknowledged service.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_UserMemory_Write-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_User-Memory_Write-PDU to an A_UserMemory_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemory_Write.ind primitive.

13) The maximal APDU length that shall be supported is specified in the specification of PID_MAX_APDULENGTH in [04].

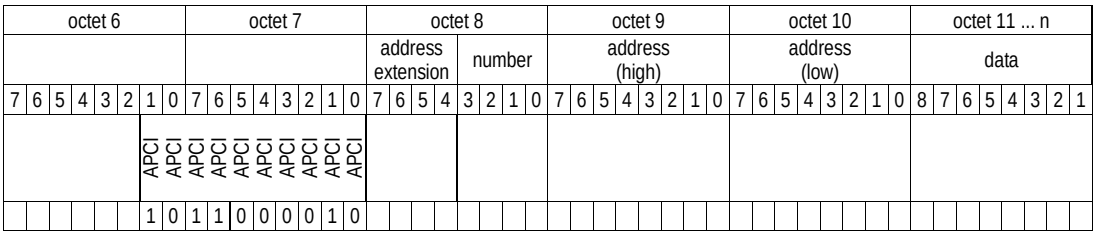


Figure 59 - A_UserMemory_Write-PDU (example)

With inactive Verify Mode the remote application process shall not respond. Instead the local Application Layer shall map a T_Data_Connected.con primitive to an A_UserMemory_Write.Lcon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemory_Write.Lcon primitive; number, memory_address and data shall be don't care.

With active Verify Mode the remote application process shall respond to the A_UserMemory_Write.ind primitive with an A_UserMemory_Write.res primitive containing the requested number of octets of the associated memory area. The value of the associated memory area shall be explicitly read back after writing to it. If the remote application process has a problem, e.g. memory area unreachable or protected or an illegal number of octets are requested, then the parameter number shall be zero and shall contain no data.

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_User-Memory_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_UserMemory_-Response-PDU to an A_UserMemory_Write.Acon primitive if an A_UserMemory_Write-PDU has been sent before over this connection. The arguments TSAP and priority shall be mapped to the corresponding arguments TSAP and priority of the A_UserMemory_Write.Acon primitive.

A_UserMemory_Write.req(ack_request, priority, hop_count_type, ASAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be written beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
data:	the octet(s) to be written

A_UserMemory_Write.Lcon(ack_request, priority, hop_count_type, ASAP, number, memory_address, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be written beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address
data:		the octet(s) to be read back or no data
a_status:	ok:	A_UserMemory_Write sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_UserMemory_Write.ind(priority, hop_count_type, ASAP, number, memory_address, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be written beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
data:	the octet(s) to be written

A_UserMemory_Write.res(ack_request, priority, hop_count_type, TSAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets written beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

A_UserMemory_Write.Acon(priority, hop_count_type, TSAP, number, memory_address, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets written beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

Error handling

If data are to be written to a protected area or from any logical address that is not associated to physical memory then the service indication shall be ignored. In any case physical memory shall not be addressable via different logical addresses. If only a part of the addressed memory is protected or does not exist, then the complete write operation shall fail.

In addition, the remote Application Layer shall ignore the A_Memory_Write.ind if the value of the parameter "number" is greater than Maximum APDU Length – 4 [14)].

If Verify Mode is active, then in case of a failed write operation no data shall be returned to indicate an error.

3.5.6.4 A_UserMemoryBit_Write-service

The A_UserMemoryBit_Write.req primitive shall be applied by the user of Application Layer, to modify between 1 bit and 40 bits in a contiguous block of up to 5 octets in the address space of the remote application controller. The parameter memory_address shall specify the 16 bit start address and number shall contain the number of octets to be modified beginning with the start address to increasing addresses. The A_UserMemoryBit_Write shall allow to

- set individual bits of the contiguous block to zero, and
- set individual bits of the contiguous block to one, and
- leave individual bits of the contiguous block unmodified, and
- invert individual bits of the contiguous block

using the parameters and_data and xor_data. Both parameters shall have the same number of octets as the contiguous block indicated in the parameter number. The resulting value for each individual bit in the contiguous block shall be computed using the two associated bits of and_data and xor_data with the following function (Figure 60):

result_bit(i) = (and_data_bit(i) AND block_bit(i)) XOR xor_data_bit(i)

and_data_bit(i)	xor_data_bit(i)	result_bit(i)
0	0	0
0	1	1
1	0	block_bit(i)
1	1	NOT block_bit(i)

Figure 60 - Function Table for A_UserMemoryBit_Write-service

The service shall be a confirmed service if Verify Mode is active, otherwise it shall be an acknowledged service.

The local Application Layer shall accept the service request and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_UserMemoryBit_Write-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_User-MemoryBit_Write-PDU to an A_UserMemoryBit_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemoryBit_Write.ind primitive.

14) The maximal APDU length that shall be supported is specified in the specification of PID_MAX_APDULENGTH in [04].

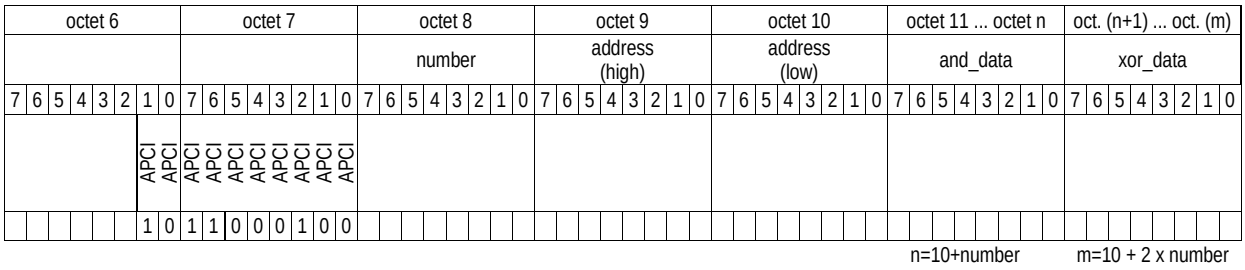


Figure 61 - A_UserMemoryBit_Write-PDU (example)

With inactive Verify Mode the remote application process shall not respond. Instead the local Application Layer shall map a T_Data_Connected.con primitive to an A_UserMemoryBit_Write.Lcon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemoryBit_Write.Lcon primitive; number, memory_address and data shall be don't care.

With active Verify Mode the remote application process shall respond to the A_UserMemoryBit_-Write.ind primitive with an A_UserMemoryBit_Write.res primitive containing the requested number of octets of the associated memory area. The value of the associated memory area shall be explicitly read back after writing to it. If the remote application process has a problem, e.g. memory area unreachable or protected or an illegal number of octets are requested, then the parameter number shall be zero and shall contain no data.

The remote Application Layer shall accept the service response and shall pass it with a T_Data_-Connected.req to the local Transport Layer. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_User-Memory_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_UserMemory_-Response-PDU to an A_UserMemoryBit_Write.Acon primitive if an A_UserMemoryBit_Write-PDU is sent before over this connection. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserMemoryBit_Write.Acon primitive.

A_UserMemoryBit_Write.req(ack_request, priority, hop_count_type, ASAP, number, memory_address, and_data, xor_data)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be modified beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
and_data:	see Figure 60
xor_data:	see Figure 60

A_UserMemoryBit_Write.Lcon(ack_request, priority, hop_count_type, ASAP, number, memory_address, and_data, xor_data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
number:		number of octets to be modified beginning with the start address to increasing addresses
memory_address:		specifies the 16 bit start address
and_data:		see Figure 60
xor_data:		see Figure 60
a_status:	ok:	A_UserMemoryBit_Write sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_UserMemoryBit_Write.ind(priority, hop_count_type, ASAP, number, memory_address, and_data, xor_data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets to be modified beginning with the start address to increasing addresses
memory_address:	specifies the 16 bit start address
and_data:	see Figure 60
xor_data:	see Figure 60

A_UserMemoryBit_Write.res(ack_request, priority, hop_count_type, ASAP, number, memory_address, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets modified beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

A_UserMemoryBit_Write.Acon(priority, hop_count_type, ASAP, number, memory_address, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
number:	number of octets modified beginning with the start address to increasing addresses, or zero to indicate a problem
memory_address:	specifies the 16 bit start address
data:	the octet(s) read back or no data

Error handling

If data are to be written to a protected area or from any logical address that is not associated to physical memory then the service indication shall be ignored. In any case physical memory shall not be addressable via different logical addresses. If only a part of the addressed memory is protected or does not exist, then the complete write operation shall fail. If Verify Mode is active, then in case of a failed write operation no data shall be returned to indicate an error.

3.5.6.5 A_UserManufacturerInfo_Read-service

The A_UserManufacturerInfo_Read.req primitive shall be applied by the user of Application Layer, to read manufacturer information in a communication partner. The manufacturer information shall consist of three octets. Octet zero shall indicate the manufacturer identification of the device. Octets one and two shall be manufacturer specific. The service shall be confirmed by the remote application process.

The local Application Layer shall accept the service request and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_UserManufacturer-Info_Read-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_User-ManufacturerInfo_Read-PDU to an A_UserManufacturerInfo_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_User-ManufacturerInfo_Read.ind primitive.

octet 6								octet 7							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI
						1	0	1	1	0	0	0	1	0	1

Figure 62 - A_UserManufacturerInfo_Read-PDU (example)

The remote application process shall respond to the A_UserManufacturerInfo_Read.ind primitive with an A_UserManufacturerInfo_Read.res primitive containing the manufacturer information.

octet 6								octet 7								octet 8								octet 9								octet 10							
																manufacturer_id								manufact. specific								manufact. specific							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																										
						1	0	1	1	0	0	0	1	1	0																										

Figure 63 - A_UserManufacturerInfo_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_UserManufacturerInfo_-Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_UserManufacturer-Info_Response-PDU to an A_UserManufacturerInfo_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_UserManufacturer-Info_Read.Acon primitive.

Limitations

In the response-PDU of this service only one single octet is specified for the KNX manufacturer code. This does not allow transmitting a KNX manufacturer code with value above 255. Therefore, use of this service is not recommended for new stack implementations.

A_UserManufacturerInfo_Read.req(ack_request, priority, hop_count_type, ASAP)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point

A_UserManufacturerInfo_Read.Lcon(ack_request, priority, hop_count_type, ASAP, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point
a_status:	ok:	A_UserManufacturerInfo_Read sent successfully with T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Connected request frame did not succeed

A_UserManufacturerInfo_Read.ind(priority, hop_count_type, ASAP)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point

A_UserManufacturerInfo_Read.res(ack_request, priority, hop_count_type, ASAP, mfact_info)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
mfact_info:	three octets manufacturer information

A_UserManufacturerInfo_Read.Acon(priority, hop_count_type, ASAP, mfact_info)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
mfact_info:	three octets manufacturer information

3.5.7 A_Authorize_Request-service

The A_Authorize_Request.req primitive shall be applied by the user of Application Layer, to inform the communication partner about the key that shall be four octets long and of data type unsigned32. The remote partner shall know a number of valid keys and shall be able to associate a valid key to an access level. This access level shall be stored as the current access level of this partner and shall be sent back in an A_Authorize_Response-PDU. Access levels (unsigned8) between 0 (maximum level, i.e. maximum access rights) and 3 (minimum level, i.e. minimum access rights) or 0 (maximum level, i.e. maximum access rights) and 15 (minimum level, i.e. minimum access rights) are allowed. If the communication partner doesn't authorize himself or if the key is not a valid key, then the current access level for this communication partner shall be set as specified under "Error and exception handling" below.

The current access level may be used by the remote application process to decide whether or not a communication partner is allowed to request a certain read or write operation.

The remote user may associate different keys to different access levels and handle following services on this connection depending on the current access level.

A current access level shall be valid until the connection is released or a new key is indicated with the A_Authorize_Request service.

The local Application Layer shall accept the service request and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Authorize_Request-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Authorize_-Request-PDU to an A_Authorize_Request.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Authorize_Request.ind primitive.

octet 6								octet 7								octet 8								octet 9								---	octet 12							
																must be 0								key																
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																												
						1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0																		

Figure 64 - A_Authorize_Request-PDU (example)

The remote application process shall respond to the A_Authorize_Request.ind primitive with an A_Authorize_Request.res primitive containing the associated access level.

octet 6								octet 7								octet 8							
																level							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI										
						1	1	1	1	0	1	0	0	1	0								

Figure 65 - A_Authorize_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Authorize_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Authorize_Response-PDU to an A_Authorize_Request.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Authorize_Request.Acon primitive.

Error and exception handling

- If the Remote Management supports authorization and if the communication partner does not authorize itself, the Remote Management shall select the maximum access level protected with FFFFFFFFh as the current access level.
- If the Remote Management supports authorization and if the communication partner authorizes itself with an invalid key, the Remote Management shall select the minimal access level (this is level 3 or level 15) as the current access level.

A_Authorize_Request.req(ack_request, priority, hop_count_type, ASAP, key)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
key:	the key of the requester

A_Authorize_Request.ind(priority, hop_count_type, ASAP, key)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
key:	the key of the requester

A_Authorize_Request.res(ack_request, priority, hop_count_type, ASAP, level)

ack_request:	Data Link Layer Acknowledge requested or don’t care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
level:	the granted access level to the requester

A_Authorize_Request.Acon(priority, hop_count_type, ASAP, level)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
level:	the granted access level to the requester

3.5.8 A_Key_Write-service

The A_Key_Write.req primitive shall be applied by the user of Application Layer, to modify or delete the key associated to a certain access level in the communication partner. The parameter level of the A_Key_Write.req primitive shall indicate the access level that shall be modified; the parameter key shall indicate the new key value.

Every device shall be able to handle exactly one key per access level. The number of access levels supported by a device is Profile dependent.

Key	Access Level
Key for level 0	0
Key for level 1	1
...	...
Key for Level N-1	N-1
None	N (free access)

Figure 66 - Association Table of Keys to Access Levels

If the key indicated in the A_Key_Write.ind primitive is FFFFFFFFh, then the corresponding key entry in the association table of keys to access levels shall be set to invalid, this is, then there shall be no key associated to the corresponding level any more. The current access level shall be less or equal to the access level indicated in the A_Key_Write.ind primitive, otherwise the remote application process shall return FFh in the A_Key_Response-PDU. In all other cases the remote application process shall store the indicated key in the corresponding entry of the association table of keys to access levels and shall respond to the A_Key_Write.ind primitive with an A_Key_Write.res primitive containing the access level set for the corresponding key.

The local Application Layer shall accept the service request and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Key_Write-PDU.

The remote Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Key_Write-PDU to an A_Key_Write.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Key_Write.ind primitive.

octet 6								octet 7								octet 8								octet 9								---	octet 12							
																level								key															
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																										
						1	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0																		

Figure 67 - A_Key_Write-PDU (example)

The remote user shall process the key as described above and shall respond with the A_Key_Write.res primitive.

octet 6								octet 7								octet 8							
												number				level							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI								
						1	1	1	1	0	1	0	1	0	0								

Figure 68 - A_Key_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Connected.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters of the T_Data_Connected.req primitive, the TSDU shall be an A_Key_Response-PDU.

The Application Layer shall map a T_Data_Connected.ind primitive with TSDU = A_Key_Response-PDU to an A_Key_Write.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_Key_Write.Acon primitive.

A_Key_Write.req(ack_request, priority, hop_count_type, ASAP, level, key)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
level:	the access level for which the key shall be modified
key:	the new value of the key or FFFFFFFFh to delete the key

A_Key_Write.ind(priority, hop_count_type, ASAP, level, key)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
level:	the access level for which the key shall be modified
key:	the new value of the key or FFFFFFFFh to delete the key

A_Key_Write.res(ack_request, priority, hop_count_type, ASAP, level)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
level:	the access level for which the associated key has been modified, or the minimum access level if it hasn't been modified

A_Key_Write.Acon(priority, hop_count_type, ASAP, level)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point
level:	the access level for which the associated key has been modified, or the minimum access level if it hasn't been modified

3.6 Coupler specific Application Layer Services on Point-to-Point connection oriented Communication Mode

The defined Coupler-specific Application Layer Services are implementation specific for Router 1.x. Please contact the KNX Association System Department for the up-to-date specifications.

4 Parameters of Application Layer

4.1 Group Object Association Table

The Group Object Association Table shall map TSAPs of multicast communication modes to Application Layer Service Access Points (ASAPs) and vice versa. One TSAP can be mapped to more than one ASAP and one ASAP can have more than one TSAP. The Group Object Association Table may be downloaded using the network management.

Various Realisation Types exist of the Group Object Association Table. Please refer to [04].

4.2 Verify flag

The Verify Flag shall control whether the Verify Mode is enabled or disabled.

The Verify Mode Control is specified in [04].