Application Note 161/13 v01

Title:	**Coupler Model 2.0**	
Status:		**Date:**
	Draft Proposal	2013.05.07
Transitional period:	Immediate effect after Final Voting.	
Date:	2013.05.07	

Subject: Legacy free extensible Coupler Model.

Documents

Modified

[01] Chapter 3/1/2 “Glossary”
[02] Chapter 3/3/3 “Network Layer”
[03] Chapter 3/5/1 “Resources”
[04] Chapter 3/5/2 “Management Procedures”
[05] Chapter 3/5/3 “Configuration Procedures”
[06] Chapter 3/7/3 “Standard Identifier Tables”
[07] Chapter 3/7/2 “Datapoint Types”
[08] Volume 6 “Profiles”

Referred

[09] Chapter 3/2/5 “Radio Frequency”
[10] Chapter 3/3/1 “Physical Layer General”
[11] Chapter 3/3/2 “Data Link Layer General”
[12] Chapter 3/3/4 “Transport Layer”
[13] Chapter 3/3/7 “Application Layer” v1.3.00 AS of 2010.10.22
[14] Chapter 3/4/1 “Application Interface Layer””
[15] Chapter 8/2/5 “RF Physical and Link Layer Tests”
[16] Chapter 8/3/3 “Network Layer Tests”
[17] Chapter 8/3/4 “Transport Layer Tests”
[18] Chapter 8/3/7 “Application (Interface) Layer Testing – Network Management Server/Client Testing”
[19] Part 8/7 “Interworking and Functionality Tests”
[20] AN124 “Interface Object Index Discovery”
[21] AN137 “Configuration Signature”
[22] KSG480-04 “Mask 0912h revised”
[23] “KNX Data Security” v0.15.04
“KNX Communication Security v0.15.04.pdf”
[24] “KSG495-17 New RF device Profile.docx”

Document updates

Version	Date	Modifications
KSG501-10	2013.01.18	• Inclusion of feedback of KSG meeting of 2013.01.08-09. - Parameter *Filter Table Use* redefined to be down-loadable Parameters: PID_FILTER_TABLE_USE. - Removed the possibility for multiple Secondary Sides. - Replaced “Coupler Model 2012” by “Coupler Model 2.0”. Introduced masks 0920h and 2920h. - PID_DEVICE_ADDR_ADD and PID_SUBNET_-ADDR_ADD removed from the Router Object. PID_DEVICE_ADDR and PID_SUBNET_ADDR removed from the Router Object. PID_ENABLE removed. - Redefinition of PID_LINE_STATUS; renamed to PID_MEDIUM_STATUS. - LTE Filter Table is mandatory
KSG501-11	2013.03.04	• Inclusion of feedback of the KSG meeting of 2013.01.28. - Indication of mandatory and optional Properties removed from Table 5 and maintained only in the Profiles. - PID_MEDIUM_STATUS interpretation generalised to “Communication Impossible”. - PID_FILTER_TABLE_USE becomes a real parameters to fully control the evaluation of the FT or not. - Clear indication in 2.4.3.1.2 of what negative and positive evaluation means. • Inclusion of Configuration Procedure received from Dr. Gütter: see 2.6.4
KSG501-12	2013.04.02	• Inclusion of TP1/RF Media Coupler. • Specification of PID_PL110_ENABLE_SBC and PID_RF_ENABLE_SBC reviewed and streamlined. Checked integration in routing algorithm. • Introduced PID_MEDIUM_INTERFACE_ID. • Introduced Load State Machine in the Device Object.
KSG501-13	2013.04.26	• Accepted all previously tracked changes. • Inclusion of feedback and conclusions of KSG online meetings of 2013.04.19 and 2013.04.26. - Inserted state machines for all communication modes. - Removed PID_FIRMWARE_REVISION, PID_ERROR_-FLAGS, PID_INTERFACE_ID (=PID_MEDIUM_INTER-FACE_ID), PID_SERVICE_CONTROL. - Added Interface Object Scan to the Configuration Procedures.
KSG501-14	2013.05.07	• Correction of 2.2.2.5.2: Coupler Mask is given by Secondary Side Medium, not by Primary Side (ref. 2920h). • Correction of 3.5.2: only one Secondary Side foreseen. • Added default hop count for Primary Side: PID_ROUTING_COUNT in the Router Object.
AN161 v01	2013.05.07	• Preparation of the Draft Proposal.

Contents

1 Purpose, motivation and scope (informative)

1.1 Introduction

This clause is not intended for integration in the KNX Specifications.

This document describes a generic Profile of a new KNX Coupler generation, called “Coupler Model 2.0”. This document does not specify the medium dependent parts, which are specified in separate papers.

The following features are not considered in this document.

- Application Program(s) in a Couplers (with additional Group Address Table).
- Couplers with multiple Secondary Sides.
- New services like for “Enhance Download Performance”.

1.2 General requirements

The below concepts are copied from the System B specification, as it is a good idea. However, these will not be integrated in the KNX Specifications

This will not be integrated because it is the case throughout the KNX Specifications.

The data representation in the memory is assumed to be big endian format.

This is also not to be integrated, as it is a general rule and because it is taken care of in the specification of the fields.

Reserved bits shall be set to zero. This means that when sending frames with reserved bits, senders shall set these bits to zero. Receivers shall verify these bits to be zero; if these bits are not zero, the frame shall be discarded.

1.3 Constraints (informative)

This clause is not intended for integration in the KNX Specifications.

This document shall specify the common model and – requirements for next KNX Coupler specifications. It specifies the second KNX *Profile Class* [1)] for Couplers and the minimal common requirements for the derived Coupler Profiles, but does not as such specify any specific Profile or mask. It is the intention that this paper will be used as the base for next Coupler Profile specifications. The first use will be the specification of the standard KNX TP1/RF Media Coupler ([24]). See also clause 2.4.1.

Table 1 gives an overview of the existing KNX Coupler Profiles Classes and masks and the planned new Profile Class and masks.

1) The concept of “Profile Classes” is used in [08] in clause 1.2.

Table 1 – Overview of existing and planned Coupler Profiles

<table>
<tr><th>Profile Class</th><th>S-Mode Configuration Profiles</th><th>Profile Collection Name</th></tr>
<tr><td>Coupler
Coupler Model 1.x</td><td>
<table>
<tr><th rowspan="2">Secondary Side</th><th colspan="4">Primary Side</th></tr>
<tr><th>TP1</th><th>PL110</th><th>RF</th><th>IP</th></tr>
<tr><td rowspan="2">TP1</td><td>0910h
0911h</td><td></td><td></td><td>091Ah</td></tr>
<tr><td>0912h</td><td></td><td></td><td></td></tr>
<tr><td>PL110</td><td></td><td>1900h</td><td></td><td></td></tr>
<tr><td>RF</td><td></td><td></td><td></td><td></td></tr>
<tr><td>IP</td><td></td><td></td><td></td><td></td></tr>
<tr><td>TP1</td><td></td><td></td><td></td><td></td></tr>
</table>
Empty cells denote Profiles that do not exist.
</td><td></td></tr>
<tr><td>Coupler Model 2.0</td><td>
<table>
<tr><th rowspan="2">Secondary Side</th><th colspan="4">Primary Side</th></tr>
<tr><th>TP1</th><th>PL110</th><th>RF</th><th>IP</th></tr>
<tr><td>TP1</td><td>0920h</td><td></td><td></td><td></td></tr>
<tr><td>PL110</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RF</td><td>2920h</td><td></td><td></td><td></td></tr>
<tr><td>IP</td><td></td><td></td><td></td><td></td></tr>
<tr><td>TP1</td><td></td><td></td><td></td><td></td></tr>
</table>
Empty cells denote Profiles that are not yet planned.
</td><td></td></tr>
</table>

1.3.1 Error reporting and diagnostics

The Coupler Model 2.0 does not foresee any standard means for error reporting or diagnostics. PID_ERROR_FLAGS is not mandatory.

1.3.2 Support of KNX Data Security

1.3.2.1 Access Control to the Coupler Resources

KNX Data Security operates at the Application Layer level and is transparent to the routing and filtering of the Coupler.

The Coupler Resources may optionally provide Access Control (Roles and Permissions), as specified in [23]. Indications are given in the Resource definitions.

1.3.2.2 Security Proxy

In KNX Data Security, it is foreseen that a Coupler may act as a Security Proxy between an assumed “safe” Subnetwork, like KNX TP1, and an assumed “unsafe” Subnetwork, like KNX RF. This will be an optional extension of the Coupler Model 2.0 and is an own standing specification.

1.3.2.3 Support for E-Mode

This version of the KNX Coupler Model 2.0 does not yet foresee any dedicated features for the support of E-Mode: assignment or distribution of IAs, announcement of use of GAs, etc.

1.3.2.4 Support of KNX RF Multi

This version of the KNX Coupler Model 2.0 does not foresee any dedicated features for the support of KNX RF Multi, like the handling of RF L2 Acknowledges, RF Repeater functionality or other.

2 Specification

2.1 Terms and definitions

These terms shall be integrated in [01].

DIV	Division with rounding results down to zero.
MOD	The remainder of division of one number by another. A MOD B = A – B * (A DIV B)
LSM	Load State Machine

2.2 Coupler Model 2.0

This document shall be integrated as a new document in Volume 6 “Profiles”. See also 3.1.

2.2.1 Motivation of the use terminology

This clause is meant as a motivation of the terminology and is not meant for inclusion in the KNX Specifications.

It has been concluded that the Coupler Model 2.0 should only have one Secondary Side. Yet, the idea is not abandoned totally and can be re-introduced later. So, the terminology that it introduced here must take care that remains usable in case of multiple Secondary Sides in the future.

- The Coupler Model 2.0 shall obviously in the Logical Topology be usable as Line Coupler or as Backbone Coupler. In this context only, the annotations Primary Side and Secondary Side make sense.
- The Coupler Model 2.0 shall however still be usable as KNX TP1 Bridge. In this context, the annotations Primary – and Secondary Side do not make sense.
- So, a general term is needed for the designation of the “connections”. However, the term “connection” mainly has an understanding in the wired world. This is less obvious in RF media (KNX RF, WLAN).
- Possibly, the Coupler Model 2.0 can sooner or later also be used for modelling KNX RF Repeater functionality.

For these reasons, the term “Medium Interface” is proposed and used.

2.2.2 Coupler Model 2.0

2.2.2.1 Basic Model

The Coupler Model 2.0 shall be composed of exactly two Medium Interfaces. For the use as a Router, one Medium Interface shall act as the Primary Side and the other Medium Interface shall act as the Secondary Side.

Each Medium Interface shall be represented by exactly one Router Object. See Figure 1. There are no requirements as to the absolute value of the Object Indexes of these Interface Objects, but the Object Index of the Router Object of the Primary Side shall be lower than the Object Index of the Router Object of the Secondary Side.

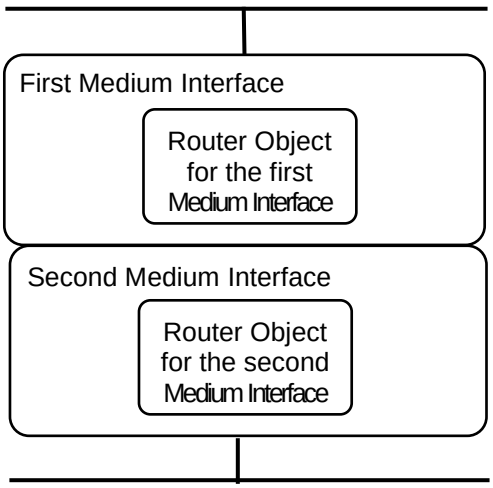


Figure 1 – Coupler Model 2.0 – basic model

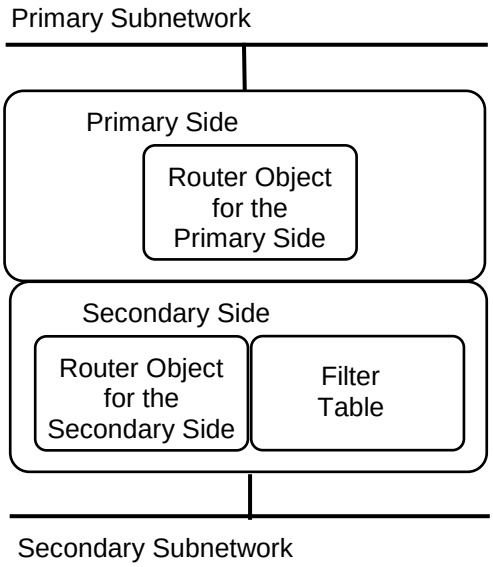


Figure 2 – Coupler Model 2.0 as Router

Implementations of the Coupler Model 2.0 shall always be usable as KNX Router. In this case, the first Medium Interface shall act as Primary Side and the second Medium Interface shall act as Secondary Side. See Figure 2. The classification of any Medium Interface as either Primary Side or Secondary Side is fixed at manufacturing time. It is not foreseen that these roles can be attributed through configuration.

For this, the Second Medium Interface shall exhibit a Filter Table.

2.2.2.2 Media for the Medium Interfaces

Secondary Side	Primary Side			
	TP1	PL110	RF	IP
TP1	yes	yes [a]	no	yes
PL110	yes	no	no	yes
RF	yes	yes [b]	no	yes
IP	no	no	no	no

[a] Implementations of such Coupler exist, but are not standardised. If this combination would cause problems, this model may not be supported in this paper.

[b] This paper does not exclude this Coupler realisation. However, it contains two DoAs: a two octet KNX PL110 DoA and a six octet KNX RF DoA. This has to be considered in the Network Configuration Procedures.

Legend

yes: A Coupler Model 2.0 with this Primary Medium and that Secondary Medium is assumed meaningful.

no: A Coupler Model 2.0 with this Primary Medium and that Secondary Medium is not assumed meaningful. Such device shall not be supported.

2.2.2.3 Router Object

The Interface Object Type of the Router Object shall be the same as in the Coupler Model 1.x. The same Properties with the same Property Identifiers shall be used for parameters with the same behaviour on both Medium Interfaces. Each Router Object shall contain the following parameters.

1. Common parameters required for each Router Object

   EXAMPLE 1 To enable the Medium Interface, to report the loss of communication on that Medium Interface, etc.

2. Parameters for the specific use of that Medium Interface as Primary – or as Secondary Side

   EXAMPLE 2 Filter Table, Subnetwork – and Device Address, etc.

3. Parameters specific to the KNX medium used for that Medium Interface

   EXAMPLE 3 Domain Addresses for open media

This is only an introduction. There is no need to be more precise. It will be clear from the Profile specification which Properties are mandatory for which Interface Object Instance.

The Properties in the Router Object for each connection shall be assigned according the following rule.

1. The first range shall be the range for the KNX communication medium independent Properties.
2. Then, range of 8 Properties are foreseen for the KNX communication medium specific Properties.

   If more Properties are needed for a KNX communication medium, then KNX Association may assign one or more additional ranges of 8 Properties to this communication medium, or may store the Properties for this medium in a medium specific Interface Object.

As a result of these rules, the current Properties defined in this document are assigned according the scheme given in Table 2.

Table 2 – PID assignment scheme in the Router Object for the Coupler Model 2.0

PID range		
Start	**End**	**Usage**
0	50	Interface Object Type independent Properties
51	95	KNX medium independent Coupler Properties
96	103	KNX TP1 specific Coupler Properties
104	111	KNX PL110 specific Coupler Properties
112	119	KNX RF specific Coupler Properties

This Table 2 does not put requirements on devices; it is only an administrative means for the KNX Association System Group, which has the responsibility for the assignment of the above and future standard Property Identifiers.

The Router Object is a Standard System Interface Object: according clause 3.1 “Assignment scheme” in [06], non-standardized (“implementation specific”) Properties shall have PID-values between 201 and 255.

KNX IP

There are no KNXnet/IP or KNX IP specific Properties in the Coupler Model 2.0. The KNXnet/IP Router shall base on the medium independent Coupler Properties and on the KNXnet/IP Parameter Object. Possibly, extension may be added later.

2.2.2.4 Control of routing and filtering

2.2.2.4.1 Network Layer, Filter Table and Coupler Parameters

As indicated in 2.3.1.1, the basic State Machine of the Network Layer for Routers can be extended and influence by additional, standard - and implementation specific parameters. For the Coupler Model 2.0 these can be the following.

1. The “State Machine for Couplers” as specified in [02] clause 2.4.2 “State Machine for Couplers”.
2. The Filter Table Realisation Type 3 as specified in 2.4.3.
3. The Parameter Filter Table Use (PID_FILTER_TABLE_USE) as specified in 2.4.4.2.13.
4. Implementation specific Parameters.

EXAMPLE 4 Parameters defining the handling of messages with hop count equal to 7.

The Coupler Model 2.0 does not have further standard Parameters influencing the routing behaviour. These are implementation specific, as is their mutual dependency and their impact on the state machine and Filter Tables. The Coupler Model 2.0 does not have a standard algorithm for concluding on the “*routing condition*” [2)].

NOTE 1 This requires that the Manufacturer Code is verified prior to accessing any Parameter.

ETS cannot interpret the implementation specific Parameters and can thus not conclude on whether or not the device will interpret the Filter Table. If the device does not interpret the Filter Table, then ETS may shorten its download time by not downloading the FT. This is described in 3.5.3 “Parameter Filter Table ”.

[2)] See “Routing in case of a Group Destination Address” in [02].

2.2.2.4.2 Implementation specific Coupler Parameters

The Coupler Model 2.0 allows for the implementation specific realisation of parameters that influence the routing and filtering behaviour. This may concern

- parameters that control the handling of messages in all communication modes [3)]
- the handling of messages with hop count equal to 7
- the configuration of a TP1 Coupler as TP1 Bridge or TP1 Repeater
- the configuration of the TP1 L2-acknowledge
- the routing of System Broadcast messages for PL110
- the routing of System Broadcast messages for RF
- …

Implementation specific Parameters and Master Reset

Yet, these parameters shall comply with the following requirements concerning Master Reset. These parameters shall reset to their implementation specific value on the following conditions.

- Master Reset with Erase Code “Factory Reset”
- Master Reset with Erase Code “Reset Param”
- Master Reset with Erase Code “Factory Reset without IA”

These Parameters shall not change on the following conditions.

- Master Reset with Erase Code “Reset IA”
- Master Reset with Erase Code “Reset Links”

2.2.2.5 Management

2.2.2.5.1 Individual Addresses

The Coupler Model 2.0 realisations shall have one single Individual Address. This IA shall be logically related to the Secondary Side.

This will be the IA that shall be used in the Network Configuration [4)].

The Primary Side shall not have an Individual Address.

2.2.2.5.2 Device Descriptor Type 0 (Mask Version)

The Mask Version of the Coupler shall indicate exclusively the Medium Type of the second Medium Interface. All Couplers according Coupler Model 2.0 with the same KNX Communication Medium used for the second Medium Interface shall have the same Mask Version. The Communication Medium used for the first Medium Interface shall be given through the Property PID_MEDIUM of the Router Object that configures the first Medium Interface. Also the Router Object of the second Medium Interface shall have the Property PID_MEDIUM. This approach shall avoid that for each new combination of Media for the second Medium Interface and for the first Medium Interface a new Mask Version has to be specified.

3) point-to-point connectionless and – connection-oriented, point-to-multipoint connectionless (multicast), point-to-all (broadcast).

4) Meant here are the Network Configuration Procedures in which the Coupler acts as a Management Server (device) in S-Mode. The Network Management Procedures in which the Coupler itself is a Client, mainly for SNA Management Procedures in E-Mode, are not considered here.

2.3 Stack and communication

2.3.1 Network Layer

2.3.1.1 State Machine of Network Layer for Routers

2.3.1.1.1 Common requirements

In [02] clause 2.4.2.4 “State Machine of Network Layer for Routers”, the following sentence shall be added immediately at the beginning of that clause, preceding “If an L_Data.ind with address_type = ´multicast´ and hop_count in [1...6] is received…”

This clause specifies the basic rules for the State Machine for the Network Layer for Routers. These rules can be influenced by additional, standard and implementation specific parameters.

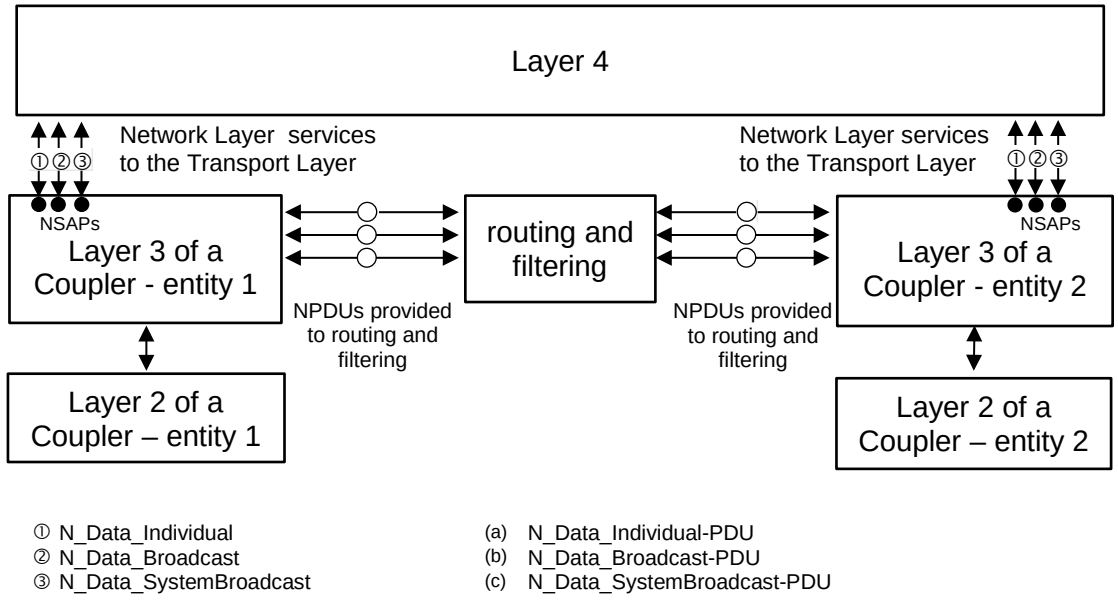


Figure 3 – Basic model of the Network Layer for a Coupler

NOTE 2 The Coupler has a different Layer-2 algorithm than an end device. This is specified in [11] clause 6.2 “The Layer-2 of a Router” and, amongst other, does not only accept point-to-point messages addressed to the own IA of the Coupler, but also to IAs of devices connected to Subnetworks of other Layer-2 instances in the Coupler.

The below considers the routing and internal handling of messages for management (N_Data_Individual, N_Data_Broadcast and N_Data_SystemBroadcast) . The internal handling of runtime multicast communication (N_Data_Group) is not specified. The use case where the Coupler Model 2.0 has itself an Application Program is not considered.

- The Coupler shall for each Interface have an entity of the “Layer 3 of a Coupler”. This shall make use of the Layer 2 services, and provide the services N_Data_Individual, N_Data_Broadcast and N_Data_SystemBroadcast to the Transport Layer, just like the Layer 3 of an end device.
- If the Transport Layer or the Transport Layer User confirms or responds to a Network Layer service primitive, then it shall use the same NSAP as on which the service is requested. The Coupler device shall thus respond to a service on the interface (Primary Side or Secondary Side) on which the request has arrived.

- Additionally each NPDU created by any Layer 3 entity shall be evaluated for routing in filtering for decision whether it shall be passed to the other Layer 3 entities. This shall be done both for any TPDU provided by the Transport Layer, as well as for any LPDU received from the Data Link Layer, this means, both for incoming messages as for outgoing messages.
- This means that NPDUs for system broadcast – and broadcast communication shall be transmitted on all interfaces.
- A layer 3 entity that handles an N-PDU provided by another Layer 3 entity shall not pass this message to the Transport Layer.

2.3.1.1.2 Implementation specific Parameters influencing Layer 3 of the Coupler

This clause concerns the general Network Layer specification, not the Coupler Model 2.0 specific possibilities. This clause is intended for integration in Chapter 3/3/3 “Network Layer” and is common to all Coupler Models. Hence, the Implementation Specific Parameters are only reminded here as a possibility, but not worked out.

The below specifications give the standard routing algorithms for the various communication modes. If the Profile of the Coupler allows for implementation specific Parameters, then these may influence the standard decisions of the flowchart and add additional or even alternative decisions.

There shall however be one combination of settings of the implementation specific Parameters that shall make the Coupler behave according the standard algorithms. It is not required that this be with the standard values of the Parameters.

2.3.1.2 Routing in case of a Group Destination Address

This clause shall **replace** the clause 2.4.2.4.1 in [02].

Telegrams on point-to-multipoint connectionless communication mode with standard Group Addresses, shall be routed or blocked from the Primary Side to the Secondary Side and vice versa, in function of the following.

1. The hop count value contained in the Telegram that is evaluated.
2. The Load State of the Device Object (see 2.4.2.2.1).
3. Parameters of the Network Layer
    1. Standard Parameters
    2. Optional implementation specific Parameters

       These parameters may allow the following:
        - generally block routing
        - generally route or
        - route according to the Filter Table
        - other routing directions.
4. The Filter Table
    1. The Load State machine of the Filter Table
    2. Parameters influencing the use of the Filter Table
    3. The evaluation of the Filter Table.

It is Profile dependent which of these conditions are available.

The standard routing shall be as follows.

if routing condition = TRUE and 0h < C < 7h **then** ROUTE_DECREMENTED
if routing condition = TRUE and C = 0h **then** IGNORE_ACKED [5)]
elsif C = 7h **then** ROUTE_UNMODIFIED
else IGNORE_TOTALLY

The above applies regardless whether the Coupler is used as a Backbone Coupler or a Line Coupler.

The routing condition for a Group Address x shall be calculated as depicted in Figure 4.

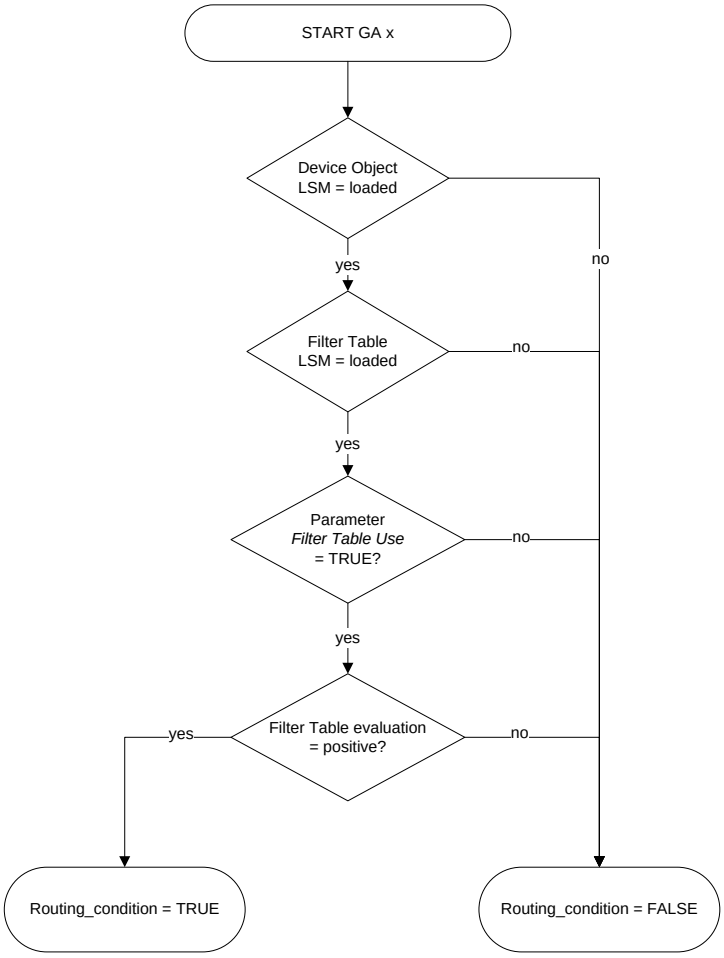


Figure 4 – Calculation of routing_condition (standard algorithm)

```
Evaluate the standard - and implementation specific Coupler parameters
IF routing condition = unknown then
        IF Load State of the Device Object != Loaded THEN
                Routing Condition = FALSE
        ELSEIF Load State of the Filter Table != Loaded THEN
                Routing Condition = FALSE
        ELSEIF Parameter Filter Tabe Use = False THEN
```

5) The ACK is sent by the Data Link Layer.

Routing Condition = FALSE
ELSEIF Filter Table evaluation = positive THEN
routing condition = TRUE
ELSE
routing condition = FALSE

2.3.1.3 Routing in case of an Individual Destination Address: Line Coupler

2.3.1.3.1 Main Line to Subline Routing

This clause shall be appended to clause 2.4.2.4.2.1 in [02].

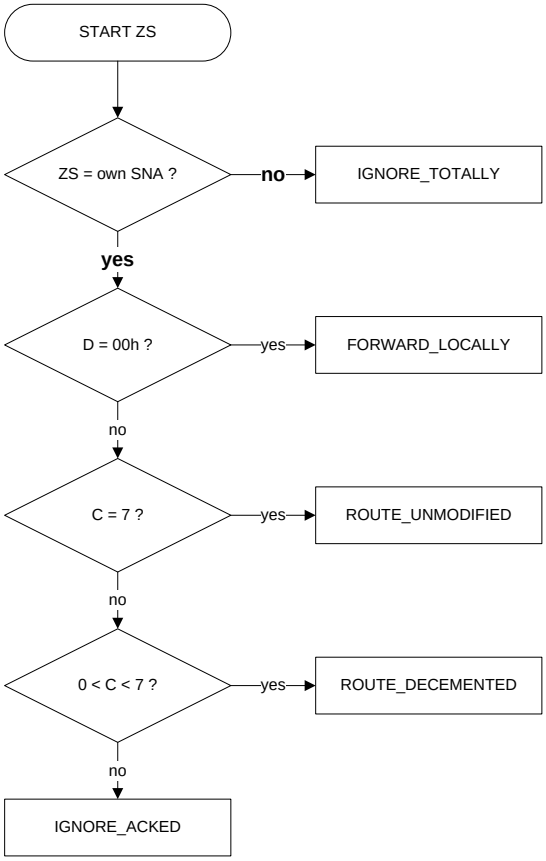


Figure 5 – Line Coupler – Routing of point-to-point messages from Main Line to Line (standard algorithm)

2.3.1.3.2 Subline to Main Line Routing

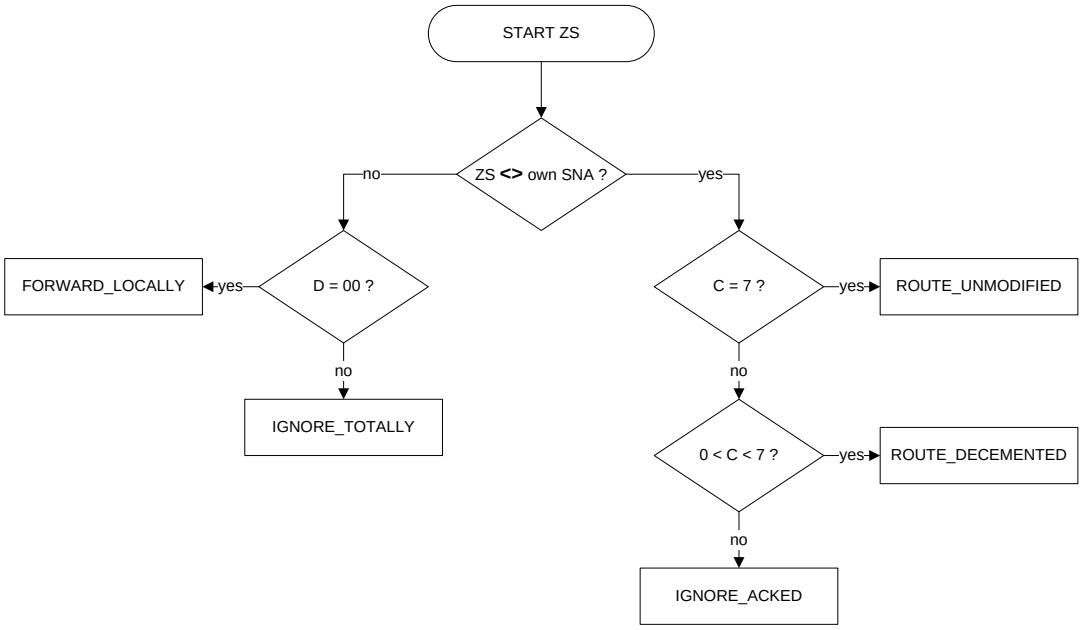


Figure 6 – Line Coupler – Routing of point-to-point messages from Line to Main Line (standard algorithm)

2.3.1.4 Routing in case of an Individual Destination Address: Backbone Coupler

This was name “Backbone Router” in [02]. this is an uncommon expression. Propose to rename this to “Backbone Coupler”.

2.3.1.4.1 Backbone Line to Main Line Routing

This clause shall be appended to clause 2.4.2.4.3.1 in [02].

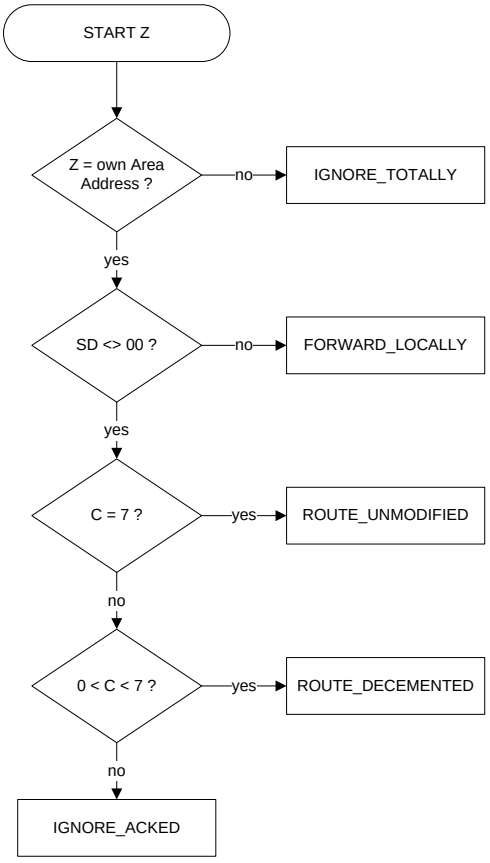


Figure 7 – Backbone Coupler – Routing of point-to-point messages from Backbone Line to Main Line (standard algorithm)

2.3.1.4.2 Main Line to Backbone Line Routing

This clause shall be appended to clause 2.4.2.4.3.2 in [02].

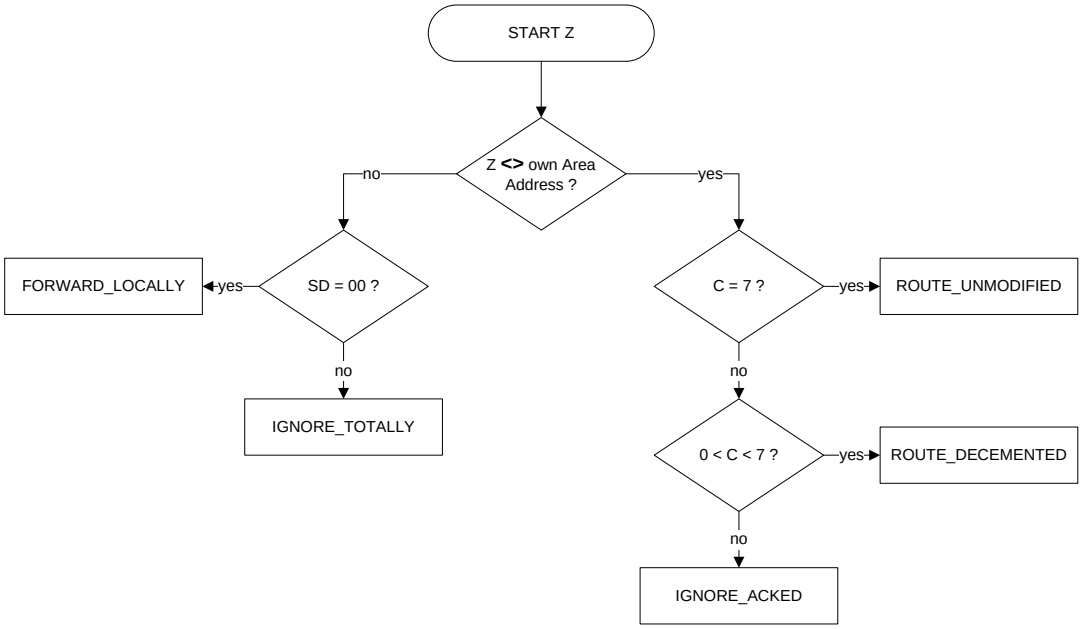


Figure 8 – Backbone Coupler – Routing of point-to-point messages from Main Line to backbone Line (standard algorithm)

2.3.1.5 Routing in case if a Broadcast – or a System Broadcast Destination Address

This clause shall REPLACE the clauses 2.4.2.4.4 “Routing in case of a Broadcast Destination Address” and the clause 2.4.2.4.5 “Routing in case of System Broadcast Destination Address – Media Coupler TP1-PL110” in [02]: the current specification handles the APCI handling only under the system broadcast handling, whereas is concerns as well the handling of a normal broadcast on a closed medium towards an open medium. The below is an attempt for a more structured and correct specification.

2.3.1.5.1 Introduction

The routing shall depend on the communication direction, from Primary Side to Secondary Side or vice versa, as specified below.

In this, it is assumed that the closed medium is on the Primary Side and that the open medium is on the Secondary Side. The inverse case is not specified.

Table 3 – Overview

Secondary Side	Primary Side	
	Closed	Open
Closed	§2.3.1.5.2	not yet defined
Open	§2.3.1.5.3	not yet defined

2.3.1.5.2 Routing in case the Primary Side is a closed medium and the Secondary Side is a closed medium

The following requirements shall apply for a Line Coupler as well as for a backbone Coupler and for the direction from Primary Side to Secondary Side as well as from Secondary Side to Primary Side.

IF C = 7h **THEN** ROUTE_UNMODIFIED

ELSIF 0h < C < 7h then ROUTE_DECREMENTED

ELSE IGNORE_ACKED

FORWARD_LOCALLY

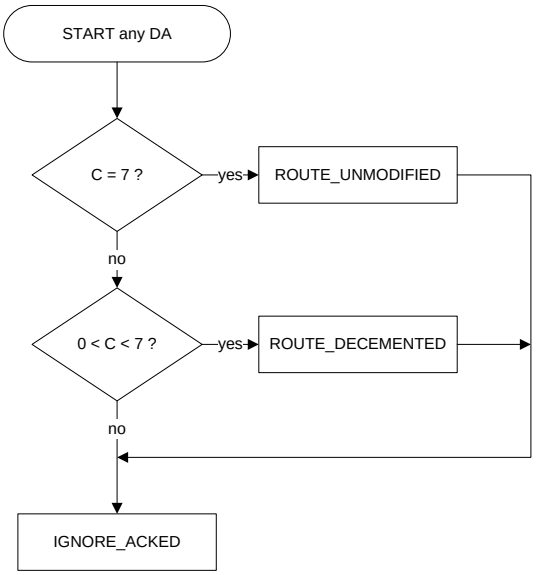


Figure 9 – Routing of broadcast messages from a closed medium to a closed medium (standard algorithm)

2.3.1.5.3 Routing in case the Primary Side is a closed medium and the Secondary Side is an open medium

2.3.1.5.3.1 Introduction

For KNX PL110 and KNX RF, there are parameters that shall control the handling of messages in system broadcast communication mode (PID_PL110_ENABLE_SBC respectively PID_RF_ENABLE_SBC).

Please refer to [08] and to clause 2.8.6.3.1 for the indication of the requirements of these Parameters for each Profile. If these parameters are not implemented, they shall be assumed to have the value “Enable”.

2.3.1.5.3.2 Primary Side (closed medium) to Secondary Side (open medium)

IF (APCI for System Broadcast AND parameter system broadcast = disable)

THEN IGNORE_ACKED;

ELSEIF C = 7h **THEN** ROUTE_UNMODIFIED

ELSEIF 0h < C < 7h **THEN** ROUTE_LAST

ENDIF

FORWARD_LOCALLY

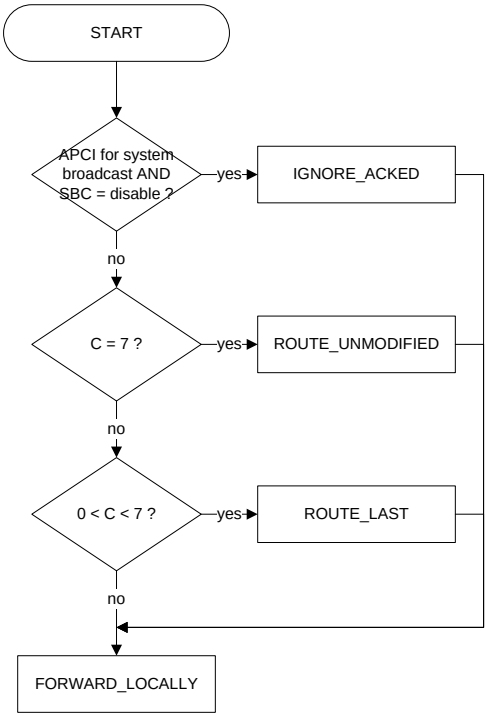


Figure 10 – Routing of system broadcast messages from a closed medium on the Secondary Side to an open medium on the Primary Side (standard algorithm)

- The KNX RF Communication Medium does not have a hop count field. ROUTE_MODIFIED and ROUTE_LAST shall mean that the message is routed without further conditions.
- If the message is routed from the closed medium to the open medium, then the communication mode that shall be used shall depend on the APCI contained in the message, according Table 4. In this, the Coupler shall additionally check whether the proper TL-service is used (T_Data_SystemBroadcast). If this is not the case, then the Coupler shall ignore the message totally and not forward it locally. (This is considered part of the condition “APCI for System Broadcast” and is not shown explicitly in the pseudo code or in Figure 10).

Table 4 – AL-service dependent use of broadcast – or system broadcast communication mode

APCI	Open Medium
APCI_DomainAddress_Write	System Broadcast
APCI_DomainAddress_Read	System Broadcast
APCI_DomainAddress_Response	System Broadcast
APCI_DomainAddressSelective_Read	System Broadcast
APCI_DomainAddressSerialNumber_Read	System Broadcast
APCI_DomainAddressSerialNumber_Response	System Broadcast
APCI_DomainAddressSerialNumber_Write	System Broadcast
APCI_IndividualAddress_Write	Broadcast
APCI_IndividualAddress_Read	Broadcast
APCI_IndividualAddress_Response	Broadcast
APCI_NetworkParameter_Write	Broadcast
APCI_NetworkParameter_Read	Broadcast
APCI_NetworkParameter_Response	Broadcast
APCI_IndividualAddressSerialNumber_Read	Broadcast
APCI_IndividualAddressSerialNumber_Response	Broadcast
APCI_IndividualAddressSerialNumber_Write	Broadcast
APCI_SystemNetworkParameter_Read	System Broadcast
APCI_SystemNetworkParameter_Response	System Broadcast
APCI_SystemNetworkParameter_Write [a]	System Broadcast
APCI_SystemNetworkParameter_InfoReport [a]	System Broadcast

[a] *These APCIs are not yet specified. The Media Coupler shall however handle them according this table.*

2.3.1.5.3.3 Secondary Side (open medium) to Primary Side (closed medium)

IF (APCI for System Broadcast AND parameter system broadcast = disable)

THEN IGNORE_ACKED;

ELSEIF C = 7h **THEN** ROUTE_UNMODIFIED

ELSEIF 0h < C < 7h **THEN** ROUTE_LAST

ENDIF

FORWARD_LOCALLY

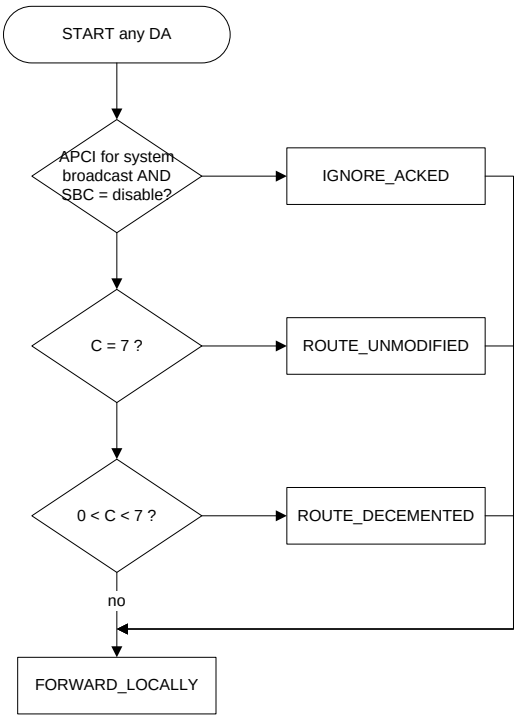


Figure 11 – Routing of system broadcast messages from a closed medium on the Primary Side to an open medium on the Secondary Side (standard algorithm)

Additionally, the following shall apply.

- If the message is routed from the open medium to the closed medium, then point-to-domain connectionless (broadcast) communication mode shall be used.
- The specific flavour of the KNX RF Communication Medium on the Secondary Side may not support Layer 2 acknowledge. In that case the IGNORE_ACKED shall be IGNORE_TOTALLY.
- The KNX RF Communication Medium does not have the field hop count. In this case, if the message is routed, it shall be routed with hop count 6.
- In the condition “APCI for System Broadcast” the Coupler shall additionally check whether the proper TL-service is used (T_Data_SystemBroadcast). If this is not the case, then the Coupler shall ignore the message totally and not forward it locally. (This is not shown explicitly in the pseudo code or in Figure 11).

2.4 Resource definition or used Resources

2.4.1 Device Descriptor Type 0 (Mask Version)

This clause is not intended for integration in the KNX Specifications.

This document specifies a common, generic Coupler model and no explicit Profile. Therefore, this document does not specify any Mask Version.

The Mask Versions based on this Coupler Model 2.0 shall the values m920h, in which m shall indicate the medium on the second Medium Interface. This has to be concluded in separate papers.

2.4.2 Device Object

Property Identifier (PID)	Property Data Type	Value
1 PID_OBJECT_TYPE	PDT_UNSIGNED_INT	DEVICE_OBJECT: 0000h
5 PID_LOAD_STATE_CONTROL	PDT_CONTROL	Load Control
11 PID_SERIAL_NUMBER	PDT_GENERIC_06	KNX Serial Number
12 PID_MANUFACTURER_ID	PDT_UNSIGNED_INT	manufacturer identifier
14 PID_DEVICE_CONTROL	PDT_GENERIC_01	temporary control field for the device
15 PID_ORDER_INFO	PDT_GENERIC_10	manufacturer specific order information
19 PID_MANUFACTURER_DATA	PDT_GENERIC_04	manufacturer specific device information
25 PID_VERSION	PDT_VERSION	revision number of the firmware
30 PID_DOWNLOAD_COUNTER	PDT_UNSIGNED_INT	Download Counter
51 PID_ROUTING_COUNT	PDT_UNSIGNED_CHAR	default hop count
54 PID_PROGMODE	PDT_BITSET8	Programming Mode
56 PID_MAX_APDULENGTH	PDT_UNSIGNED_INT	Maximal APDU-length for Device Management
57 PID_SUBNET_ADDR	PDT_UNSIGNED_CHAR	Subnetwork Address
58 PID_DEVICE_ADDR	PDT_UNSIGNED_CHAR	Device Address
71 PID_IO_LIST	PDT_UNSIGNED_INT[]	Interface Object List
78 PID_HARDWARE_TYPE	PDT_GENERIC_06	Hardware Type
83 PID_DEVICE_DESCRIPTOR	PDT_GENERIC_02	Device Descriptor Type 0

2.4.2.1 PID_OBJECT_TYPE (PID = 1)

- **Property name:** Interface Object Type
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_PropDataType (DPT_ID = 7.010)

For the common specification of PID_OBJECT_TYPE, please refer to [03].

2.4.2.2 PID_LOAD_STATE_CONTROL (PID = 5)

- **Property name:** Load Control
- **Property Datatype:** PDT_CONTROL
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

NOTE 3 PID_LOAD_STATE_CONTROL of the Security Interface Object is on the Black List, as obviously, it contains KNX Data Security information. For this implementation in the Router Object, there is no reason that it would be on the Black List, but the Access has to be respected.

This use of PID_LOAD_STATE_CONTROL shall comply with the common specification given in [03] in clause 4.16.2 “Load State Machine – Realisation Type 1 (Property based)”.

As concerning the Load Events to be supported and the Load Events that may be returned by the Coupler Model 2.0, please refer to clause 2.8.6.2.2 in this document.

The below specifies deviations or additional requirements for the Coupler Model 2.0.

2.4.2.2.1 Usage by the MaS

The MaS (Coupler) shall additionally comply with the following.

- If the Load State is “Loaded” then the following shall apply.
  - The Application Program, if implemented, shall run.
  - The (implementation specific) Parameters of the Network Layer may be interpreted.
- If the Load State is any other than “Loaded” then the following shall apply.
  - The Application Program, if implemented, shall not run.
  - The (implementation specific) Parameters of the Network Layer may or may not be interpreted. This Load State gives a clear indication that the MaS (Coupler) is under configuration by the MaC. It is implementation specific whether or not the (implementation specific) Parameters are interpreted or not.
  - The routing condition shall be FALSE.

2.4.2.2.2 Usage by the Mac

The MaC shall set the LSM to *Unloaded* prior to changing any standard – or implementation specific Parameter in the Coupler Model 2.0. When the download has completed, the MaC shall set the LSM back to *Loaded*.

If the MaC needs to restart the MaS during any Configuration Procedure, then it shall firstly set the LSM to *Loaded*, prior to restarting the MaS, and possibly set the LSM back to *Loading* after the restart, if needed to continue the Configuration Procedure.

2.4.2.3 PID_SERIAL_NUMBER (PID = 11)

- **Property name:** Serial Number
- **Property Datatype:** PDT_GENERIC_06
- **Datapoint Type:** DPT_SerNum (DPT_ID = 221.001)

For the common specification of PID_SERIAL_NUMBER, please refer to [03].

2.4.2.4 PID_MANUFACTURER_ID (PID = 12)

- **Property name:** Manufacturer Identifier
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** None.

For the common specification of PID_MANUFACTURER_ID, please refer to [03].

2.4.2.5 PID_DEVICE_CONTROL (PID = 14)

- **Property name:** Device Control
- **Property Datatype:** PDT_BITSET8 (alt.: PDT_GENERIC_01)
- **Datapoint Type:** DPT_Device_Control (DPT_ID = 21.002)

For the common specification of PID_DEVICE_CONTROL, please refer to [03].

Please refer to the Profile specification for the fields that have to be supported.

Bits that are not supported shall have the default value as specified in [03].

2.4.2.6 PID_ORDER_INFO (PID = 15)

- **Property name:** Order Info
- **Property Datatype:** PDT_GENERIC_10
- **Datapoint Type:** None.

For the common specification of PID_ORDER_INFO, please refer to [03].

2.4.2.7 PID_MANUFACTURER_DATA (PID = 19)

- **Property name:** Manufacturer Data
- **Property Datatype:** PDT_GENERIC_04
- **Datapoint Type:** not applicable: implementation specific encoding

For the common specification of PID_MANUFACTURER_DATA, please refer to [03].

2.4.2.8 PID_VERSION (PID = 25)

- **Property name:** Version
- **Property Datatype:** PDT_VERSION (alt.: PDT_GENERIC_02)
- **Datapoint Type:** DPT_Version (DPT_ID = 217.001)

For the common specification of PID_VERSION, please refer to [03].

2.4.2.9 PID_DOWNLOAD_COUNTER (PID = 30)

- **Property name:** Download Counter
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_Value_2_Ucount (DPT_ID = 7.010)

The Property PID_MCB_TABLE shall exclusively relate to the contents of the Filter Table and shall not cover any Parameter (memory mapped or Property based). In order to allow for a partial download, the Property PID_DOWNLOAD_COUNTER should be integrated.

Support by the MaS (device)

Please refer to the common specification of PID_DOWNLOAD_COUNTER in [21].

Support by the MaC (ETS)

Prior to executing a Partial Download, the MaC shall verify the value of PID_DOWNLOAD_-COUNTER of the part to be downloaded, as defined in [21].

If PID_DOWNLOAD_COUNTER is not available for the part to be downloaded, then the MaC shall not perform a Partial Download.

2.4.2.10 PID_ROUTING_COUNT (PID = 51)

- **Property name:** Routing Count
- **Property Datatype:** PDT_UNSIGNED_CHAR
- **Datapoint Type:** None.

For the common specification of PID_ROUTING_COUNT, please refer to [03].

2.4.2.11 PID_PROGMODE (PID = 54)

- **Property name:** Programming Mode
- **Property Datatype:** PDT_BITSET8
- **Datapoint Type:** None.

For the common specification of PID_PROGMODE, please refer to [03].

2.4.2.12 PID_MAX_APDU_LENGTH (PID = 56)

- **Property name:** MAX. APDU-Length
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** None.

For the common specification of PID_MAX_APDU_LENGTH, please refer to [03].

In the Device Object, this Property shall contain the maximal APDU-length of the messages that are supported by the Coupler for its own device management.

In the Router Object, this Property shall contain the maximal APDU-length of the messages that the Coupler can send to and receive from the Medium Interface that it represents.

In the Coupler Model 2.0, this Property is required in the Router Object in the Primary Side and in the Secondary Side. Both shall have the same value.

2.4.2.13 PID_SUBNET_ADDR (PID = 57)

- **Property name:** Subnetwork Address
- **Property Datatype:** PDT_UNSIGNED_CHAR
- **Datapoint Type:** None.

For the common specification of the Property *Subnetwork Address*, please refer to [03].

The Property Subnetwork Address in the Device Object of the Coupler Model 2.0 shall be the Subnetwork Address part of the own Individual Address of the Coupler which it shall respond to for Network – and Device Management.

This Property shall react to the “Configuration Procedures for Configuring the Subnetwork Address” (clause 1.3 in [05]) on the Primary Side and on the Secondary Side.

Default value

The default – and ex-factory value of the Subnetwork Address shall be FFh. The Coupler Model 2.0 shall assume this value on the following conditions.

- Master Reset with Erase Code “Factory Reset”
- Master Reset with Erase Code “Reset IA”

The value shall not change on the following conditions.

- Master Reset with Erase Code “Reset AP”
- Master Reset with Erase Code “Reset Param”
- Master Reset with Erase Code “Reset Links”
- Master Reset with Erase Code “Factory Reset without IA”

2.4.2.14 PID_DEVICE_ADDR (PID = 58)

- **Property name:** Device Address
- **Property Datatype:** PDT_UNSIGNED_CHAR
- **Datapoint Type:** None.

For the common specification of the Property *Device Address*, please refer to [03].

The Property Device Address in the Device Object of the Coupler Model 2.0 shall be the Device Address part of the own Individual Address of the Coupler which it shall respond to for Network – and Device Management.

Default value

The default – and value of the Device Address shall be 00h. The Coupler Model 2.0 shall assume this value on the following conditions.

- Master Reset with Erase Code “Factory Reset”
- Master Reset with Erase Code “Reset IA”

The value shall not change on the following conditions.

- Master Reset with Erase Code “Reset AP”
- Master Reset with Erase Code “Reset Param”
- Master Reset with Erase Code “Reset Links”
- Master Reset with Erase Code “Factory Reset without IA”

2.4.2.15 PID_IO_LIST (PID = 71)

- **Property name:** Interface Object List
- **Property Datatype:** PDT_UNSIGNED_INT[]
- **Datapoint Type:** None

For the common specification of the Property *Interface Object List*, please refer to [03].

2.4.2.16 PID_DEVICE_DESCRIPTOR (PID = 83)

- **Property name:** Device Descriptor
- **Property Datatype:** PDT_GENERIC_02
- **Datapoint Type:** None.

For the common specification of the Property *Device Descriptor*, please refer to [03].

2.4.3 Filter Table Realization Type 3

This clause shall be integrated in Chapter 3/5/3 “Resources” ([03]).

- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	none	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

The specification of this memory mapped Resource is given firstly. In the next clause, the Router Object, which shall refer to this Filter Table Realisation Type, will be specified.

2.4.3.1 Format and encoding

This specification is based on the specification in [22], in which the fixed memory location is removed.

2.4.3.1.1 Format

The Filter Table Realisation Type 3 shall be organised as a memory mapped bit-field of 65 536 bits and thus 8 192 octets. Each bit shall uniquely correspond to one Group Address. The full 16 bit KNX GA encoding range shall be supported.

The bit corresponding to a GA can be calculated as follows. “GA_value” shall be the numerical value of the GA.

- The address of the octet in which the bit resides shall be calculated as follows.

  octet_address = GA_value div 8

- The position of the bit within that octet shall be calculated as follows. The bits withing the octet shall be numbered from 0, least significant position, to 7, most significant position.

  bit_position = GA_value mod 8

EXAMPLE 5 For the GA 7/1879 in the Filter Table, the following is calculated.

The octet address will be 3F57h div 8 = 07EAh

The bit position will be 3F57h mod 8 = 7.

2.4.3.1.2 Encoding

Each bit in the Filter Table corresponding to a GA shall comply with the following encoding.

0: The Filter Table evaluation for this Group Address shall be negative: the message with the GA corresponding to this bit in the Filter Table shall not be passed.

1: The Filter Table evaluation for this Group Address shall be positive: the message with the GA corresponding to this bit in the Filter Table shall be passed.

NOTE 4 This solely concerns the check of the passing of a message in function of the Filter Table. Other conditions, like other Coupler Parameters and the hop_count of the Telegram, may exist that influence the passing of the Telegram.

Exceptions

- The Group Address 0000h is no valid KNX Group Address. The bit 0 in the Filter Table, the least significant bit of octet 0, shall be “don’t care”. It shall not be evaluated.
- For the routing of Frames in (system) broadcast address communication mode, the Filter Table shall not be evaluated.

2.4.3.2 Default value

The default contents of the Filter Table – Realisation Type 3 shall be implementation specific. The Coupler Model 2.0 shall assume this value on the following conditions.

- Master Reset with Erase Code “Factory Reset”
- Master Reset with Erase Code “Reset Param”
- Master Reset with Erase Code “Factory Reset without IA”

The value shall not change on the following conditions.

- Master Reset with Erase Code “Reset IA”
- Master Reset with Erase Code “Reset AP”
- Master Reset with Erase Code “Reset Links”

2.4.3.3 Location

The Filter Table realisation Type 3 shall be a memory mapped Resource. It shall start at the memory address indicated by the Property PID_TABLE_REFERENCE of the Router Object. This Property, and hence the memory location of the Filter Table, shall comply with the common specification of PID_TABLE_REFERENCE in [03].

There are no requirements concerning the value of the memory location and hence the value of PID_TABLE_REFERENCE.

2.4.3.4 Usage by the Management Server (device)

The below specifies the standard, minimal MaS side interpretation of the Filter Table. Additionally, the MaS may have further, implementation specific parameters that influence the evaluation of the Filter Table.

The Filter Table Realisation type 3 shall be controlled and managed by a Load State Machine – Realisation Type 1 (Property based) as given in [03] in clause 4.16.2..

If the Load State of the Filter Table is *Loaded*

If the MaS (Coupler) evaluates the Filter Table for concluding to pass a Frame with a given GA or not, then this evaluation shall be positive if the bit corresponding to the GA in the Filter Table is set. If the bit is cleared, then this evaluation shall be negative.

If the Load State of the Filter Table is any other than *Loaded*

The evaluation of the Filter Table shall always be negative.

NOTE 5 This solely concerns the evaluation of the Filter Table in concluding on passing a Frame. Other conditions exist as well: the Network Layer State Machine (see [02]) and the standard – and implementation specific Parameters.

2.4.3.5 Usage by the Management Client

The MaC shall respect the Load State Machine if modifying the Filter Table.

The MaC shall allocate the Filter Table memory using the Additional Load Control for memory allocation and then retrieve the pointer to the memory located from PID_TABLE_-REFERENCE.

When writing the Filter Table, the MaC shall access the Filter Table using DMP_MemWrite-_RCoV or DMP_UserMemWrite_RCoV in function of the memory addresses that are accessed: it shall use DMP_MemWrite_RCoV to access the memory below and equal to FFFFh and DMP_UserMemWrite_RCoVto access the memory above FFFFh..

2.4.4 Router Object

2.4.4.1 Introduction and overview

This overview only lists the Properties that are relevant for the new Coupler model. For the indications whether these Properties are mandatory or optional, please refer to clause 2.8 “Profile”.
This clause 2.4.4.1 is not intended for integration in the KNX Specifications.

This clause firstly gives the overview of the Properties in the Router Object.

Then, the possibly usage specific requirements of the Properties are specified, if any (clauses 2.4.4.2.1 and following).

Object Indexes of the Router Object

According to the Basic Model, each medium attachment shall have exactly one Router Object.

The Object Index of the Router Object of the Primary Side shall have a lower value than the Object Index of the Router Object of the Secondary Side. The MaC shall conclude on the relationship between Router Object instance and Primary – or Secondary Side based on this these Object Indexes.

The MaC can read the Object Indexes of the Router Objects through the Management Procedure NM_ObjectIndex_Read (using point-to-point communication mode), reading PID_OBJECT_INDEX and the Object Type of the Router Object. This Procedure is mandatory for the Coupler Model 2.0. (See [20])

Overview of the Properties in the Router Object

NOTE 6 Property implementation may be mandatory or optional in function of the Router instance (Primary – or Secondary Side). Please refer to 2.8.6 for the indication about the mandatory and optional Properties.

Table 5 - Properties in the Router Object

Property Identifier (PID)	Property Data Type	Value
Interface Object Type independent Properties (0 to 50)		
1 PID_OBJECT_TYPE	PDT_UNSIGNED_INT	ROUTER_OBJECT: 0006h
5 PID_LOAD_STATE_CONTROL	PDT_CONTROL	See clause 2.4.4.2.2..
7 PID_TABLE_REFERENCE	PDT_UNSIGNED_LONG	Pointer to the Filter Table
27 PID_MCB_TABLE	PDT_GENERIC_08	Memory Control Table
28 PID_ERROR_CODE	PDT_EUM8	Specifies the reason for Load State "Error".
29 PID_OBJECT_INDEX	PDT_UNSIGNED_CHAR	Object Index of the Router Object.
KNX medium independent Coupler Properties (51 to 95)		
51 PID_MEDIUM_STATUS	PDT_GENERIC_01	Status of the communication medium of this Medium Interface.
56 PID_ROUTETABLE_CONTROL	PDT_FUNCTION	Set of methods to modify the Filter Table.
57 PID_COUPL_SERV_CONTROL	PDT_BITSET8	Inconsistency and Subnetwork Address (SNA) mechanisms.
58 PID_MAX_APDU_LENGTH	PDT_UNSIGNED_INT	MAX. APDU-Length
61 PID_HOP_COUNT	PDT_UNSIGNED_INT	Default hop count for Frames routed to the Primary Side from a medium on the Secondary Side that has no hop count.
63 PID_MEDIUM	PDT_ENUM8	Medium of this Coupler Part.
67 PID_FILTER_TABLE_USE	PDT_BINARY_INFORMATION	Indicates whether the Filter Table is used or not.
KNX TP1 specific Coupler Properties (96 to 103)		
There are no KNX TP1 specific standard Coupler Properties specified.		
KNX PL110 specific Coupler Properties (104 to 111) 'C': The Property is mandatory in case the Router Object represent a medium connection to a PL110 medium. 'O': the Property is always optional.		
104 PID_PL110_ENABLE_SBC	PDT_FUNCTION	Flag to enable temporary the routing of system broadcast from Primary Side to Secondary Side
105 PID_PL110_DOA	PDT_UNSIGNED_INT	PL110 Domain Address of the connection
KNX RF specific Coupler Properties (112 to 119) 'C': The Property is mandatory in case the Router Object represent a medium connection to an RF medium. 'O': the Property is always optional.		
112 PID_RF_ENABLE_SBC	PDT_FUNCTION	Flag to enable temporary the routing of system broadcast from Primary Side to Secondary Side.
A dedicated paper exists with the TP1/RF Coupler based on this model. This is the document "KSG495-17 New RF device Profile.docx" ([24]). The Media Coupler specific specifications from that paper will be merged in this paper once both papers are stable.		
KNX IP specific Coupler Properties		
There are no KNXnet/IP or KNX IP specific Properties in the Coupler Model 2.0. The KNXnet/IP Router bases on the medium independent Coupler Properties and on the KNXnet/IP Parameter Object. Possibly, extension may be added later.		

2.4.4.2 Interface Object Type independent Properties

In the below, PID_TABLE_REFERENCE is specified as PDT_UNSIGNED_LONG. This is a correction of the specification in [03], which specifies PDT_UNSIGNED_INT.

NOTE 7 The PID assignment below reserves for each KNX communication medium a range of 8 possible PID-values.

NOTE 8 The Profile only specifies the Coupler Profile and makes no difference between Line Coupler, Backbone Coupler or Repeater. The above Properties have to be present in the three modes of the Coupler device.

2.4.4.2.1 PID_OBJECT_TYPE (PID = 1)

- **Property name:** Interface Object Type
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_PropDataType (DPT_ID = 7.010)
- **Access:**
  - If KNX Data Security is not supported: 3/1
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	none	n/a	n/a
M	**NwPar**	**R**	any	none	n/a

List: This Property shall be on the White List.

For the common specification of PID_OBJECT_TYPE, please refer to [03].

The Object Type of the Coupler Model 2.0 will be the same as for the preceding Coupler models, namely 6, even if certain Properties will no longer be allowed.

This Property is mandatory on the Primary Side and on the Secondary Side 1.

2.4.4.2.2 PID_LOAD_STATE_CONTROL (PID = 5)

- **Property name:** Load Control
- **Property Datatype:** PDT_CONTROL
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

NOTE 9 PID_LOAD_STATE_CONTROL of the Security Interface Object is on the Black List, as obviously, it contains KNX Data Security information. For this implementation in the Router Object, there is no reason that it would be on the Black List, but the Access has to be respected.

This use of PID_LOAD_STATE_CONTROL shall comply with the common specification given in [03] in clause 4.16.2 “Load State Machine – Realisation Type 1 (Property based)” and clause 4.4.2 specifically for the Router Object.

As concerning the Load Events to be supported and the Load Events that may be returned by the Coupler Model 2.0, please refer to clause 2.8.6.3.2 in this document.

This shall be the Load State Machine for the Filter Table that is used for the Medium Interface that is represented by this Router Object.

There are no further specific additions or deviations for the Load State Control in the Coupler Model 2.0.

2.4.4.2.3 PID_TABLE_REFERENCE (PID = 7)

- **Property name:** Table Reference
- **Property Datatype:** PDT_UNSIGNED_LONG
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/X
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	none	n/a	n/a

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

Please refer to the common specification of PID_TABLE_REFERENCE in [03].

2.4.4.2.4 PID_MCB_TABLE (PID = 27)

- **Property name:** Memory Control Table
- **Property Datatype:** PDT_GENERIC_08
- **Datapoint Type:** None
- **Access:**
  - If KNX Data Security is not supported: 3/X
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	none	n/a	n/a

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

Please refer to the common specification of PID_MCB_TABLE in [03].

PID_MCB_TABLE is mandatory in the Router Object of the Secondary Side.

For the implementation in the Coupler Model 2.0, PID_MCB_TABLE shall exclusively relate to the contents of the Filter Table (not to any other Parameter) and shall be read-only. Bit 0 in the contained CRC-control byte shall always be cleared: if the Load State is “Loaded”, then the CRC shall always be “valid”. It shall not be possible that the contents of the Filter Table modifies without resulting in a different CRC.

Usage by the MaC

The MaC shall not modify this Property. The MaC shall only read this Property to learn the CRC-value. The CRC shall always be valid.

2.4.4.2.5 PID_ERROR_CODE (PID = 28)

- **Property name:** Error code
- **Property Datatype:** PDT_ENUM8
- **Datapoint Type:** DPT_ErrorClass_System (20.011)
- **Access:**
  - If KNX Data Security is not supported: (3/X)
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
O	**Data:**	**Read:**	any	plain	3
		Write:	none	n/a	n/a

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

Please refer to the common specification of PID_ERROR_CODE in [03].

2.4.4.2.6 PID_OBJECT_INDEX (PID = 29)

- **Property name:** Object Index
- **Property Datatype:** PDT_UNSIGNED_CHAR
- **Datapoint Type:** DPT_Value_1_Ucount
- **Access:**
  - If KNX Data Security is not supported: 3/0
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	manuf	A	0
	NwPar	**R**	any	none	n/a

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

This implementation of PID_OBJECT_INDEX shall comply with the common specification of PID_OBJECT_INDEX in [03]. Additionally, it is required for this implementation that it be accessible using the A_NetworkParameter_Read-service in the procedure NM_Object-Index_Read, as specified in [20].

2.4.4.2.7 PID_MEDIUM_STATUS (PID = 51)

- **Property name:** Medium Status
- **Property Datatype:** PDT_GENERIC_01
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/X
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	none	n/a	n/a
M	**NwPar**	**R**	any	none	n/a
		T	n/a	n/a	n/a

List: This Property shall neither be on the White List, the Black List or the Intermediate List.

This use of PID_MEDIUM_STATUS shall comply with the common specification given in [03] in clause 4.4.3, including the use in NM_NetworkParameter_Write.

The following specification shall **replace** the specification of PID_MEDIUM_STATUS in [03]. This is because it can now be implemented in the Router Object on the Secondary Side and transmitted on the Primary Side, and it can as well be implemented in the Router Object of the Primary Side and be transmitted on the Secondary Side.

This Property shall include the status of the communication medium to which this medium interface communicates.

Bit 0 was named POWER_DOWN_SUBLINE before. This is now named COMMUNICATION_IMPOSSIBLE.

Bit	Name	Description	Coding
0	COMMUNICATION_IMPOSSIBLE	Indicates whether or not communication is possible using the medium connection represented by this Router Object.	0: FALSE: communication is possible 1: TRUE: communication is impossible
1 to 7		reserved	Shall be 0.

The Coupler shall distribute the value of PID_MEDIUM_STATUS of one Medium Interface to all other Medium Interfaces with the Management Procedure NM_NetworkParameter_-Write_R under the control of the field EN_SUBLINE_STATUS of PID_COUPL_SERV_-CONTROL, with Object Type= Router Object and PID = PID_MEDIUM_STATUS in broadcast communication mode; this single transmission shall be triggered by a change of value of this Property.

```
        /* Announce the status of the medium on one Medium Interface through the remaining Medium Interface. */
For each Medium Interface other than the initiating Medium Interface {
        if PID_COUPL_SERV_CONTROL.EN_SUBLINE_STATUS = enable then
                NM_NetworkParameter_Write_R(ASAP = void, comm_mode = point-to-all-points connectionless,
                hop_count_type_req = 6, object_type = Router Object, PID = PID_MEDIUM_STATUS,
                value = PID_MEDIUM_STATUS.Value)
}
```

It shall be noted that this message does not say what Medium Interface has a problem: the Primary Side or the Secondary Side. This would even become more imprecise if ever multiple Secondary Sides were re-introduced again. However, even the use of the Object Instances or Object Indexes would not help, as there are no requirements to these! In broadcast; this message thus only says that one Medium Interface of this Coupler with IA x.y.z has a problem. (In point-to-point, reading out by the MaC, knowing the relation between Medium Interface and Object Index or – Type, this does become clear.)

2.4.4.2.8 PID_ROUTETABLE_CONTROL (PID = 56)

- **Property name:** Routing Table Control
- **Property Datatype:** PDT_FUNCTION
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	Data:	Read:	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

Please refer to the common specification of PID_ROUTETABLE_CONTROL in the Router Object in [03].

2.4.4.2.9 PID_COUPL_SERV_CONTROL (PID = 57)

- **Property name:** Coupler Services Control
- **Property Datatype:** PDT_BITSET8
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	Data:	Read:	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

Please refer to the common specification of PID_COUPL_SERV_CONTROL in the Router Object in [03].

2.4.4.2.10 PID_MAX_APDU_LENGTH (PID = 58)

- **Property name:** MAX. APDU-Length
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/X
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	manuf	A	0

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

Please refer to the common specification of PID_MAX_APDU_LENGTH in the Router Object in [03].

In the Router Object, this Property shall contain the maximal APDU-length of the messages that it can route from and to the network connection that is controlled through this Router Object instance.

NOTE 10 In 2.8.6 it is required that PID_MAX_APDU_LENGTH is mandatory in the Router Object for the Coupler Model 2.0. It is also required that the Coupler Model 2.0 is able to support an APDU-length of 55 octets or more. That is as such not a requirement to PID_MAX_APDU_LENGTH, but a requirement to the Medium Dependent Layers of the Coupler Model 2.0. Therefore, this is added to the clauses of 2.8.2 for the various KNX Communication Media.

2.4.4.2.11 PID_HOP_COUNT (PID = 61)

- **Property name:** Hop Count
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** None.
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

This Property shall contain the default hop count that the Coupler shall use if it forwards a Frame to the Medium Interface that is represented by this Router Object.

This Property shall specifically be used if the Frame is forwarded from a KNX Medium that does not support any hop count, like KNX RF.

If the Coupler acts in the topology as a Line Coupler, the meaningful value for this Property on the Primary Side is five; if the Coupler in the topology acts as a Backbone Coupler, the meaningful value for this Property on the Primary Side is four. This way, Frames forwarded by this Coupler on the Primary Side have the same default hop count as if the Frame would originate from a TP1 Subnetwork.

Please refer to the Profile definitions in 2.8 for the indications for which Primary – or secondary Sides for which KNX Communication Media this Property is mandatory.

2.4.4.2.12 PID_MEDIUM (PID = 63)

- **Property name:** Medium
- **Property Datatype:** PDT_ENUM8
- **Datapoint Type:** DPT_Medium (20.1004) (see 3.3.1)
- **Access:**
  - If KNX Data Security is not supported: 3/X
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	Data:	Read:	any	plain	3
		Write:	manuf	A	0

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

The Property Medium shall indicate the KNX Communication Medium that is used for the Subnetwork Connected that is represented by this Router Object.

This Property is mandatory for each Router Object, both on the Primary Side as on the Secondary Side.

2.4.4.2.13 PID_FILTER_TABLE_USE (PID = 67)

- **Property name:** Filter Table Use
- **Property Datatype:** PDT_BINARY_INFORMATION
- **Datapoint Type:** DPT_Bool (1.002)
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	Data:	Read:	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

2.4.4.2.13.1 Abstract Resource definition

This parameter shall indicate whether the Filter Table shall be used or not.

2.4.4.2.13.2 Format and encoding

The parameter *Filter Table Use* shall be encoded as DPT_Bool (DPT_ID: 1.002).

Value	Specification
0	The Filter Table shall not be used.
1	The Filter Table shall be used.

2.4.4.2.13.3 Usage by the Management Server (Coupler)

Please refer to the Abstract Resource definition.

The MaS shall actively interpret this Parameter, even if the implementation specific Coupler Parameters would allow concluding on whether the Filter Table is used or not.

The MaS shall interpret this parameter *Filter Table Use* in the routing and filtering functionality as follows.

Value	Interpretation by the MaS
0	The Filter Table shall not be evaluated. If the standard - or implementation specific Coupler Parameter values conclude on the evaluation of Filter Table, then, this evaluation shall return negative for all Group Addresses.
1	The MaS shall evaluate the Filter Table as specified in 2.4.3.

This Parameter Filter Table Use may be part of the calculation of the routing condition, as indicated in 2.2.2.4.

2.4.4.2.13.4 Usage by the Management Client

Writing

The MaC shall always set this parameter, on any modification of the Filter Table contents or the (implementation specific) Coupler parameters.

Value	Interpretation by the MaC - writing
0	The FT contents shall be considered as invalid. The FT shall not be downloaded.
1	The FT contents shall be considered as valid. The FT shall be downloaded.

Reading and interpretation

The MaC shall write this Property parameter during any Configuration (Full download, Partial Download, Parameters only or any other).

The MaC shall read out this Parameter in status information, device information or in case of reverse engineering or other.

Value	Interpretation by the MaC – reading
0	The FT contents shall be considered as invalid. The FT may either be: - read out but marked invalid, or - not be read out.
1	The FT contents shall be considered as valid. The FT may be read out.

2.4.4.3 KNX PL110 specific Coupler Properties

2.4.4.3.1.1 PID_PL110_ENABLE_SBC (PID = 104)

- **Property name:** PL110 System Broadcast routing enable
- **Property Datatype:** PDT_FUNCTION
- **Datapoint Type:** None
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	Data:	Read:	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

This Property shall control whether the Routing of System Broadcast is enabled or disabled.

The default value shall be 0: disabled.

The Functionality of PID_PL110_ENABLE_SBC for PL110 shall be identical to the functionality of PID_RF_ENABLE_SBC (PID = 112). Please refer to clause 2.4.4.4.1 for the specification.

System broadcast routing Parameters for KNX RF and for KNX PL110

The parameter for System Broadcast handling for a KNX PL110 connection PID_PL110_-ENABLE_SBC (PID = 104) is not shared with the system broadcast handling parameters for a KNX RF connection (PID_RF_ENABLE_SBC;PID = 112).

This allows that there may be different parameter sets, - values and – handling between KNX PL110 and KNX RF.

This could also later on allow for an easy introduction of a PL110/RF Media Coupler.

2.4.4.3.2 PID_PL110_DOA (PID = 105)

- **Property name:** PL110 Domain Address
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_Value_2_Ucount
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	**Data:**	**Read:**	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

For the common specification of the Property PL110 Domain Address in the Coupler Model 2.0, please refer to the specification of PID_DOMAIN_ADDRESS [03].

This Property shall exclusively be available in the Router Object of a connection of the medium PL110.

2.4.4.3.2.1 Usage by the MaS (device)

2.4.4.3.2.1.1 Primary Side

The Coupler Model 2.0 shall use the value of this Property as PL110 Domain Address if the Primary Side is a PL110 segment.

NOTE 11 This is not yet foreseen.

2.4.4.3.2.1.2 Secondary Side

NOTE 12 Opposite to the Device Address and the Subnetwork Address, this Property is not mirrored with any Property in the Device Object.

In the *Router Object* of the Secondary Side, this Property shall be the Domain Address of the Coupler. The Coupler shall use this Property Value as Domain Address on its Secondary Side.

This instance of this Property shall react to the Management procedures for configuring the PL110 Domain Address (see [04]) on the Primary Side and on the Secondary Side. These are:

- NM_DomainAddress_Read
- NM_DomainAndIndividualAddress_Read
- NM_DomainAndIndividualAddress_Write
- NM_DomainAndIndividualAddress_Write2
- NM_DomainAddress_Scan

2.4.4.3.2.1.3 Default value and Master Reset

The default value of the PL110 Domain Address shall be 00FFh. The Coupler Model 2.0 shall assume this value on the following conditions.

- Master Reset with Erase Code “Factory Reset”
- Master Reset with Erase Code “Reset IA”

The value shall not change on the following conditions.

- Master Reset with Erase Code “Reset AP”
- Master Reset with Erase Code “Reset Param”
- Master Reset with Erase Code “Reset Links”
- Master Reset with Erase Code “Factory Reset without IA”

2.4.4.3.2.2 Usage by the MaC (ETS)

The MaC shall manage the DoA of the Secondary Side through the above listed Management Procedures. For the Primary Side, the MaC shall exclusively use Data Property services.

2.4.4.4 KNX RF specific Coupler Properties

2.4.4.4.1 PID_RF_ENABLE_SBC (PID = 112)

- **Property name:** RF System Broadcast routing enable
- **Property Datatype:** PDT_FUNCTION
- **Datapoint Type:** None
- **Access:**
  - If KNX Data Security is not supported: 3/2
  - If KNX Data Security is supported:

Support	Service:		Role:	Security:	Access level:
M	Data:	Read:	any	plain	3
		Write:	ETS	A	2

List: This Property shall neither be on the Black List, on the White List or on the Intermediate List.

This Property shall control whether the Routing of System Broadcast is enabled or disabled.

The default value shall be 0: disabled.

System broadcast routing Parameters for KNX RF and for KNX PL110

The parameters for System Broadcast handling for a KNX RF connection PID_RF_-ENABLE_SBC (PID = 112) are not shared with the system broadcast handling parameters for KNX PL110 (PID_PL110_ENABLE_SBC; PID = 104).

This allows that there may be different parameter sets, - values and – handling between KNX PL110 and KNX RF.

This could also later on allow for an easy introduction of a PL110/RF Media Coupler.

1) Write (A_FunctionPropertyCommand-PDU)

Octet 10
Mode

The command shall be encoded according DPT_Enable (DPT_ID: 1.003; 0: Disable; 1: Enable; see [07]).

- If the MaC sets PID_RF_ENABLE_SBC to “Enable”, then the MaS shall immediately activate the routing of RF System Broadcast Telegrams for a timeout of 10 minutes.
- If the MaC sets PID_RF_ENABLE_SBC to “Disable” then the MaS shall immediately deactivate the routing of RF System Broadcast Telegrams.

Response (A_FunctionPropertyState_Response-PDU)

Octet 10	Octet 11
Return Code	Mode

Return Code 00h: SUCCESS

Mode: This shall be the current mode of the routing of RF System Broadcast Telegrams, according DPT_Enable.

2) Read (A_FunctionPropertyState-Read-PDU)

No Data.

The MaC shall read the state of the Function Property PID_RF_ENABLE_SBC to obtain the current mode of the routing of RF System Broadcast Telegrams.

The format, contents and interpretation of the response (A_FunctionPropertyState_-Response-PDU) shall be identical as to the above response to the A_FunctionProperty-Command.

2.4.5 LTE Address Filter Object

Object Indexes of the LTE Address Filter Object

According to the Basic Model, the Secondary Side shall have exactly one *LTE Address Filter Object*.

There are no conditions concerning the Object Indexes of any *LTE Address Filter Object*.

Overview of the Properties in the LTE Address Filter Object

This overview only lists the Properties that are relevant for the Coupler Model 2.0. For the indications whether these Properties are mandatory or optional, please refer to clause 2.8 “Profile” clause 2.8.6.
This overview is not intended for integration in the KNX Specifications.

Property Identifier (PID)	Property Data Type	Value
1 = PID_OBJECT_TYPE	PDT_UNSIGNED_INT	LTE_ADDRESSFILTER_OBJECT: 0007h
5 = PID_LOAD_STATE_CONTROL	PDT_CONTROL	see: load state machines
52 = PID_LTE_ROUTETABLE	PDT_GENERIC_05[32]	LTE Filter Table

2.4.5.1 PID_LOAD_STATE_CONTROL (PID = 5)

This clause is not intended for integration in the KNX Specifications.

This use of PID_LOAD_STATE_CONTROL shall comply with the common specification given in [03] in clause 4.16.2 “Load State Machine – Realisation Type 1 (Property based)” and clause 4.5.2 specifically for the LTE Address Filter Object.

As concerning the Load Events to be supported and the Load Events that may be returned by the Coupler Model 2.0, please refer to clause 2.8.6.3.2 in this document.

2.4.5.2 PID_LTE_ROUTETABLE (PID = 52)

This clause is not intended for integration in the KNX Specifications.

This use of PID_LTE_ROUTETABLE shall comply with the common specification given in [03] in clause 4.5.4 “PID_LTE_ROUTETABLE (PID = 52)”.

2.5 Management procedures

This clause is not intended for integration in the KNX Specifications.

This document does not introduce neither modify any Management Procedures.

2.6 Configuration Procedures

This clause shall be integrated in [05].

2.6.1 General requirements

2.6.1.1 Introduction

For the Coupler Model 2.0, the general requirements of the Configuration Procedures of System B shall apply. These are specified in Chapter 3/5/3 “Configuration Procedures” clause 2.5.1.

The only difference is that there is only one mandatory memory mapped programmable part, this is the “Filter Table Realization Type 3” as specified in clause 2.4.3.

2.6.1.2 Load State Machine

For the Load State Machine in the Device Object, the MaS shall never restart the Coupler Model 2.0 while the Load State is “Loading”.

For the Load State Machine in the Router Object, the requirements of clause 2.5.1.1 “Load State Machine” of System B in [05] shall apply as well for the Coupler Model 2.0.

2.6.1.3 Load Controls

The same Load Controls for the Load State Machine and the same additional Load Events (0Bh “Data Relative Allocation”) as specified for System B in clause 2.5.1.2 “Load controls” in [05] shall apply as well for the Coupler Model 2.0.

2.6.2 Configuration Procedure to verify the installed device

This is not a full and own standing Configuration Procedure, but a series of Management Procedures that are used before starting the further real Configuration Procedures.

```
        /* Establish a Transport Layer connection to the Coupler. */
DMP_Connect_RCo(IA, connection-oriented);
```

NOTE 13 DMP_Connect_RCo also returns the Device Descriptor Type 0.

Compare the read out value of DD0 with the value of DD0 the device that will be downloaded.

```
        /* Check that the hardware type matches the given value in the product data */
        /* by reading and evaluating PID_MANUFACTURER_ID and PID_HARDWARE_TYPE. */
DMP_Identify_RCo2(Manufacturer Code, Hardware Type);

        /* Get the access rights. */
DM_Authorize2_Rco(client_key);
```

If the MaC (ETS) does not know the Object Indexes then it shall firstly scan the Interface Objects. The request and the response shall use point-to-point connectionless communication mode; if the MaC has a TL connection established with the Coupler, it shall use point-to-point connection-oriented communication mode.

```
DMP_InterfaceObjectScan_R(ASAP)
```

In the list of Interface Objects, there shall be two instances of the Router Object. The MaC shall relate the Router Object instance with the lower Object Index to the Primary Side and the instance with the higher Object Index to the Secondary Side.

In the below, RouterObjectPrim denotes the Router Object of the Primary Side and RouterObjectSec denotes the Router Object of the Secondary Side.

With the returned information, the MaC knows what Router Object relates to the Primary Side and what Router Object relates to the Secondary Side.

2.6.3 Network Configuration Procedure – Individual Address Assignment

The Device Configuration Procedures mainly use point-to-point communication. For this, it is primordial that the local interface and the Couplers in the installation have the correct Individual Addresses.

```
        /* Assign the main IA. */
NM_IndividualAddress_Write(/* [in] */ IA_new);
```

2.6.4 Configuration Procedure for a complete download

```
        /* Match the installed device with the description from the MaC's project information as specified in 2.6.2 */

        /* Connect to the Coupler. */
DMP_Connect_RCo(connection oriented communication, descriptor_type = 0);
```

MergeID 1

At this point, implementation specific additional Management Procedures may be executed.
This MergeID 1 is thought to allow the manufacturer to verify possible additional Properties before the download starts.

```
        /* Set the Load State Machine of the Device Object to "Unloaded". */
DMP_LoadStateMachineWrite_R_Co_IO(object_index = 0, data = {event = 04h});
```

This setting of the LSM of the Device Object makes amongst other that the routing of group messages is halted, so that the MaC may now modify any parameter without possible ambiguous situations for the MaS (Coupler).

MergeID 2

At this point, implementation specific additional Management Procedures may be executed.
At this point, the LSM of the Device Object is “Unloaded”, but the LSM of the Filter Table is still “Loaded”.

The ETS developers have been requested to foresee this MergeID, but it is not yet confirmed whether this will effectively be implemented. (May 2, 2013).

NOTE 14 PID_LOAD_STATE_CONTROL is of type PDT_CONTROL and not of type PDT_FUNCTION. It shall thus be accessed using A_PropertyValue_Write and not using A_FunctionProperty_Command.

```
        /* Clear the Filter Table. */
DM_FunctionProperty_Write_R(OI = RouterObjectSec.ObjectIndex, PID = PID_ROUTE_TABLE_CONTROL,
        command = SRVID_CLEAR_ROUTINGTABLE, error);
        /*      There is no need to check the state of the clearing of the Filter Table in a subsequent */
        /*      A_FunctionPropertyState_Read, as the Coupler should have reported any error */
        /*      in the “error” above. */

        /* Set the Load State to “Loading”. */
        /* There is no need to identify the PID: this procedure always accesses PID_LOAD_STATE_CONTROL. */
DMP_LoadStateMachineWrite_RCo_IO(OI = RouterObject.ObjectIndex, data = EV_START_LOAD);


        /* Memory allocation shall always be done, in line with other Configuration Procedures. */
        /* Always the maximal size for 65 535 GAs shall be allocated = MAX_FILTER_TABLE_SIZE. */
DMP_LoadStateMachineWrite_RCo_IO(OI = RouterObjectSec.ObjectIndex,
        data = EV_ADDITIONAL(requested memory_size = MAX_FILTER_TABLE_SIZE));
```

```
        /* Load the Filter Table. */
        /* The Filter Table shall only be loaded if it is used by the device. */
IF PID_FILTER_TABLE_USE = TRUE THEN
        /* Get the pointer to the memory mapped Filter Table. */
        DMP_InterfaceObjectRead_R(OI = RouterObject.ObjectIndex, PID = PID_TABLE_REFERENCE,
                start_index = 1, element_count = 1, data = FilterTable.BaseAddress)
        /* Write the Filter Table. */
        /* If the Filter Table is totally below FFFFh, then download it in full using standard Memory Services. */
        IF (FilterTable_BaseAddress + FilterTableSize < FFFFh) THEN DMP_MemWrite_RCoV(FilterTable)
                /* If the Filter Table is partially in standard memory and partially in user memory. */
        ELSEIF FilterTable_BaseAddress < FFFFh THEN
        {
                DMP_MemWrite_RCoV(FilterTable part below FFFFh);
                DMP_UserMemWrite_RCoV(FilterTable part above FFFFh)
        }
        ELSE DMP_UserMemWrite_RCoV(FilterTable);
ENDIF
        /* Finally, also set the Property PID_FILTER_TABLE_USE. */
DMP_InterfaceObject_Write_R(OI = RouterObject.ObjectIndex, PID = PID_FILTER_TABLE_USE,
        start_index = 1, element_count = 1, data = Filter Table Use)
```

MergeID 3

This MergeID 3 is in between the writing of the Filter Table and the setting of the FT LSM to “Loaded”, to allow the manufacturer to add additional parameters.

```
/* Set the Load State of the LSM of the Filter Table to “Loaded”. */
DMP_LoadStateMachineWrite_RCo_IO(OI = RouterObject.ObjectIndex, data = EV_LOAD_COMPLETE);
/* Read the CRC and store it in the ETS database. */

/* Set the Load State of the LSM of the Device Object to “Loaded”. */
DMP_LoadStateMachineWrite_RCo_IO(OI = 0, data = EV_LOAD_COMPLETE);
```

MergeID 4

```
/* Read the download counter. */
DMP_InterfaceObjectRead_R(OI = 0, PID = PID_DOWNLOAD_COUNTER, start_index = 1, element_count = 1,
        data = ReadDownLoadCounter);
```

The MaC shall store the read value of the Download Counter in its repository, so that it can use it later for possible acceptance of partial download.

```
/* Disconnect from the device. */
        DMP_Disconnect_RCo()
```

2.6.5 Configuration Procedure for partial download - parameters

```
/* Match the installed device with the description from the MaC’s project information as specified in 2.6.2 */

/* Read the download counter. This can be done connectionless or connection-oriented.  */
DMP_InterfaceObjectRead_R(OI = 0, PID = PID_DOWNLOAD_COUNTER, start_index = 1,
        element_count = 1, data = ReadDownLoadCounter)
```

If the read value of the Download Counter differs from the value that the MaC stored after the preceding configuration, then the MaC shall not continue with a partial download, but instead perform a complete download as specified in 2.6.4. Otherwise, the procedure shall continue as follows.

```
/* Establish a Transport Layer connection to the remote device. */
DMP_Connect_RCo(IA, connection-oriented)
```

MergeID 1

Here, there is the possibility for the first implementation specific actions.

This procedure will be completed in co-operation with the ETS developers in a next version of this document.

2.6.6 Configuration Procedure for partial download – Group Addresses

```
        /* Match the installed device with the description from the MaC's project information as specified in 2.6.2 */

        /* Establish a Transport Layer connection to the remote device. */
DMP_Connect_RCo(IA, connection-oriented)
```

MergeID 1

```
        /* Set the Load State to "Loading". */
        /* There is no need to identify the PID: this procedure always accesses PID_LOAD_STATE_CONTROL. */
DMP_LoadStateMachineWrite_RCo_IO(OI = RouterObject.ObjectIndex, data = EV_START_LOAD);
```

```
        /* Clear the Filter Table. */
DM_FunctionProperty_Write_R(OI = RouterObject.ObjectIndex, PID = PID_ROUTE_TABLE_CONTROL,
        command = SRVID_CLEAR_ROUTINGTABLE, error);
        /*      There is no need to check the state of the clearing of the Filter Table in a subsequent */
        /*      A_FunctionPropertyState_Read, as the Coupler should have reported any error */
        /*      in the "error" above. */
```

This procedure will be completed in co-operation with the ETS developers in a next version of this document.

2.6.7 Configuration Procedure for unload

This procedure will be completed in co-operation with the ETS developers in a next version of this document.

2.7 Usage and context

This clause is not intended for integration in the KNX Specifications.

The mask 2920h is introduced as the first standardised Media Coupler between KNX TP1 and KNX RF. This is necessary for the project of support of KNX RF S-Mode devices, this is, the support of KNX RF in ETS.

The Coupler Model 2.0 is an abstract Coupler Model definition, to finally evaluate to a further implementation – and legacy independent Coupler Model, allow for implementation specific extensions and make provisions for future Coupler generations.

2.8 Profiles definitions

It is the intention that in the future, not all Profiles are defined together in one single document in Volume 6. Instead, basically, every Profile will have its own document.
In [24] therefore, the Profiles for mask 2705h and for mask 27B0h are in different clauses.
In this paper however, the Profiles for the abstract Coupler Model 2.0, for mask 0920h and for mask 2920h are defined in the classic style in single tables, next to each other. With the integration of this document in the KNX Specifications, this will be split in different papers.

2.8.1 Introduction and common requirements

2.8.1.1 Introduction and common requirements for the Coupler Model 2.0

This clause is intended for integration in the Profile specific document for the Coupler Model 2.0 in Volume 6.

This Coupler Model 2.0 is specified so that the Resources are accessible both from the Primary – as well as from the Secondary Side, so that the Coupler can be configured from either side.

2.8.1.2 Introduction and common requirements for the mask 0920h

This clause is intended for integration in the Profile specific document for mask 0920h in Volume 6.

2.8.1.3 Introduction and common requirements for the mask 2920h

This clause is intended for integration in the Profile specific document for mask 2920h in Volume 6.

2.8.1.3.1 Device model

The mask 2920h Profile shall be a realisation of the abstract Coupler Model 2.0 extended with features to realise a KNX TP1/RF Media Coupler.

In this mask 2920h, the Primary Side shall be realised as a KNX TP1 medium interface and the Secondary Side shall be a KNX RF Ready interface.

The KNX RF Ready medium specific Coupler parameters are located in a KNX RF Medium Object instance that shall be related to the Secondary Side.

The mask 2920h device model is sketched in Figure 12.

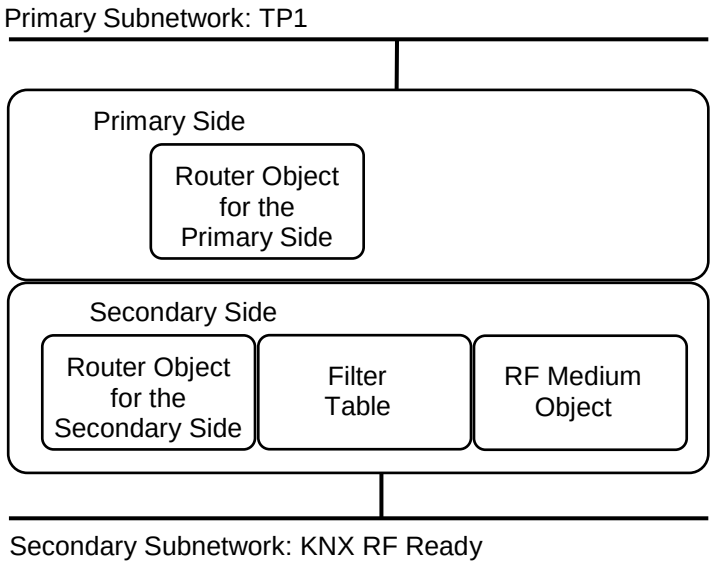


Figure 12 – Mask 2920h model

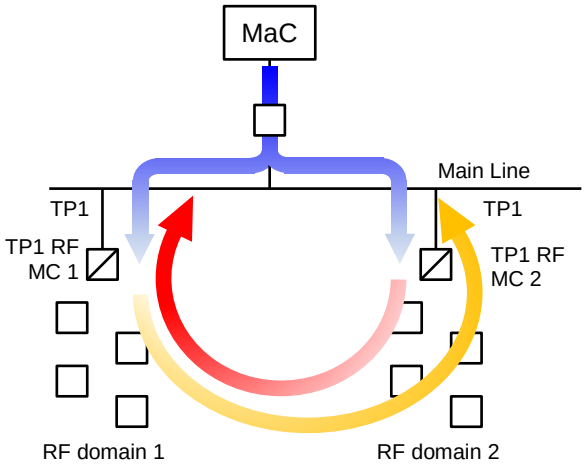
2.8.1.3.2 Loops and routing with open media

During configuration, for basic network discovery and configuration, the MaC will use messages in system broadcast communication mode.

EXAMPLE 6 A_DomainAddress_Write, A_DomainAddress_Read, A_SystemNetworkParameter_Read etc.

In an installation with two or more RF domains, with two or more KNX TP1/RF Media Couplers, these messages will route uncontrolled.

EXAMPLE 7 If the MaC transmits a message for system broadcast communication on RF, this will be received by both TP1/RF Media Couplers. Each will forward the message on system broadcast communication mode, so, this will be received by the other Media Coupler, which will route it on TP1 The loop is started and will never end.



System Broadcast communication uses the KNX Serial Number, but it is not possible for any Media Coupler to filter on this, because these system broadcast services are especially used to communicate with device not already known in the RF Domain.

Therefore, a KNX TP1/RF Media Coupler will per default not route messages that will use system broadcast communication model. Only when commissioning an unconfigured KNX RF S-Mode device (not already set on the RF domain), the MaC prepare only and exactly that TP1/RF Media Coupler that will be used in this communication: it will set the Property PID_RF_ENABLE_SBC to the value “enable”. When the RF DoA is assigned to the end device, the MaC can set the Property value back to “disable”.

The Media Coupler will also autonomously disable again its routing of system broadcast messages after a time out (default value = 1 min).

2.8.1.3.3 Routing broadcast - and system broadcast messages

From TP1 to RF

Additional to the standard Network Layer routing, the KNX TP1/RF MC has to evaluate the APCI to decide how to forward a broadcast message received from the TP1 connection on RF: on (normal) broadcast or on system broadcast.

- If the If the APCI shall according clause 2.3.1.5.3.2 be routed on **“point-to-domain, connectionless” (broadcast)** communication mode, then the message shall be routed on RF using broadcast communication mode.

  The field *SN/DoA* of the RF Frame shall be set with the RF DoA of the TP1/RF Media Coupler and the field *AET* shall be set to 1.

- If the APCI shall according clause 2.3.1.5.3.2 be routed on **"point-to-all points, connectionless" (system broadcast)** communication mode, then the handling shall depend further on the value of the Property PID_RF_ENABLE_SBC.
  - If this Property has the value "enabled" then the message shall be routed on RF using system broadcast communication mode.
    The field *SN/DoA* field of the RF Frame shall be set with the KNX Serial number of the TP1/RF Media Coupler and the field *AET* shall be set to 0.
  - If the Property has the value "disabled" then the TP1 broadcast message will in no way be forwarded to RF.

From RF to TP1

- If the Property PID_RF_ENABLE_SBC has the value "enabled" then RF System broadcast messages shall be forwarded on TP1 in (normal) broadcast communication mode.
- If the Property PID_RF_ENABLE_SBC has the value "disabled" then RF System broadcast messages will in no way be forwarded on TP1.

2.8.2 Medium dependent layers

2.8.2.1 TP1 medium dependent layers

This clause is intended for integration in the Profile specific document for mask 0920h in Volume 6.

Feature	Coupler Model 2.0		Mask 0920h		Mask 2920h	
	Primary Side	Secondary Side	Primary Side	Secondary Side	Primary Side	Secondary Side
1 Physical Layer	M	M	TP64 or TP256	TP64 or TP256	TP64 or TP256	n/a
2 Data Link Layer	M	M	M	M	M	
3 Extended Frames	M	M	M	M	M	
4 APDU-length	≥ 55 [6])	≥ 55 [6])	≥ 55 [6])	≥ 55 [6])	≥ 55 [6])	

2.8.2.2 RF medium dependent layers

This clause is intended for integration in the Profile specific document for mask 2920h in Volume 6.

6) TP1 Couplers shall support the Routing of L_Data_Extended frames and all Extended Frame Formats.

Feature	Coupler Model 2.0	Mask 2920h Secondary Side
1. Physical Layer Sender	X	X
2. Physical Layer Transceiver	M	M
3. Link Layer	M	M
4. Physical Layer - general	M	M
5. Data Link Layer – general	M	M
6. Data Link Layer – router	M	M
7. Data Link Layer - bridge	O	n/a
8. LTE frame format	M	M
9. Extended LTE Group Addresses and frame acceptance	M	M
10. Link Layer-Retransmitter	O	O
11. Link Layer Media Coupler	M	M
12. Local Services Metering	n/a	n/a
13. KNX Serial Number	M	M
14. APDU-length	≥ 55	≥ 55

2.8.3 Common stack

2.8.3.1 Overview

Feature	Coupler Model 2.0	Mask 0920h	Mask 2920h
1 Physical Layer – General	M	M	M
2 Data Link Layer – General	M	M	M
3 Data Link Layer – Router	M	M	M
4 Data Link Layer – Bridge	O	M	n/a
5 Network Layer general	M	M	M
6 Network Layer - Router	M	M	M
7 Network Layer - Bridge	O	M	n/a
8 Network Layer RF Retransmitter	O	n/a	O
9 Network Layer TP1/RF Media Coupler	O	n/a	O [7)]
10 Transport Layer – multicast	O	O	O
11 Transport Layer – connection oriented minimal	M	M	M

7) This is optional for mask 2920h because it contains requirements on the functionality of the RF Retransmitter that bases on the KNX Serial Number and the History List (and not on the DoA) and on the mapping of RF Extended Group Addresses (with KNX Serial Number) to standard Group Addresses.
For the mask 2920h, it is sufficient to require the support of the “Network Layer – Router”, which includes the mapping of the system broadcast services. Group Addresses need not to be translated with KNX RF S-Mode devices, because they are the same on TP1 as on RF with DoA.

Feature	Coupler Model 2.0	Mask 0920h	Mask 2920h
12 AL – Group Object services	O	O	O
13 AL – Property Value Services	M	M	M
14 AL – Function Property Services	M	M	M
15 AIL – Group Objects	O	O	O
16 AIL – Interface Objects	M	M	M
17 AIL – Function Properties	M	M	M
18 AIL – GO indirection	O	O	O
19 Application Interface Layer for unidirectional devices	n/a	n/a	n/a

2.8.3.2 Physical Layer - general

Specification	Test
• **General**	
[10] (contains no requirement)	none.

2.8.3.3 Data Link Layer - general

Specification	Test
• **General**	
[11] - §1.1 “Functions of the Data Link Layer” - §1.2 “Possible Media and their Impact on Layer-2” - §1.3 “Objective”	tested with medium specific tests
• **Individual address /Group Address**	
[11] - §1.4 “Definitions”	tested with medium specific tests
• **Data Link Layer Protocol**	
[11] - §3 “Data Link Layer Protocols”	tested with medium specific tests
• **Parameters**	
[11] - §4 “Parameters of Layer-2” (except TP1 Fast Polling)	tested with medium specific tests

2.8.3.4 Data Link Layer - Router

Specification	Test
[11] - §6.2 “The Layer-2 of a Router”	- tested with medium specific tests

2.8.3.5 Data Link Layer - Bridge

Specification	Test
[11] - §6.1 “The Layer-2 of the TP1-Bridge and the TP1 Repeater”	tested with medium specific tests

2.8.3.6 Network Layer - general

Specification	Test
• **General**	
[02] - §1 “Overview”	**[16]**
• **NPDU**	
[02] - §2.1 “NPDU”	**[16]** - §3 (Black Box Tests) 8) - Routers - Bridge
• **Parameters**	
[02] - §2.3 “Parameters of Network Layer” - hop_count: preferred value: 6.	- §3(Black Box Tests) 8) - Routers - Bridge
• **state machine**	
[02] - §2.4.1	**[16]** - §3 (Black Box Tests) 8) - Routers - Bridge

2.8.3.7 Network Layer – Router

Specification	Test
[02] - §2.4.3 “State Machine of Network Layer for Routers”	**[16]** - §5 “Test case 3: Testing of routing algorithm in routers”

Please note that this part of the Network layer specification will be extended and modified by clause 2.3.1 in this document. This also includes the filtering, routing and mapping of System Broadcast services.

2.8.3.8 Network Layer – Bridge

Specification	Test
[02] - §2.4.2 “State Machine of Network Layer for Bridges”	**[16]** - §5 “Test case 3: Testing of routing algorithm in routers”

2.8.3.9 Network Layer- RF Retransmitter

Specification	Test
[09] - §5.5.1 “History List” - §5.5.2 “RF Repeat Counter” - §5.5.3 “Filtering”	**[15]** - §3

8) System 1 or BCU 1 devices in test 3.4 (broadcast communication) may answer with routing count = 0...6.

2.8.3.10 Network Layer TP1/RF Media Coupler

Specification	Test
[09] - §5.5.4 “Retransmitter Flowchart” - §5.6 “The Layer-2 of an RF-TP Media Coupler”	

2.8.3.11 Transport Layer–multicast

Specification	Test
• **TPDU**	
[12] - §1.2 “Point-to-Multipoint, Connectionless (Multicast) Communication Mode” - §3.1 “T_Data_Group Service”	[17] - All end devices - Routers - Bridge [15] - RF bidirectional end device - RF unidirectional sender

2.8.3.12 Transport Layer–connection oriented minimal

In case the connection-oriented TL is not implemented (if it is optional in a Profile) a T_Disconnect-PDU shall be sent on reception of a T_Connect-PDU.

2.8.3.13 Application Layer – Group Object services

Specification	Test
• **APDU**	
[13] - §2 “APDU” o A_GroupValue_Read-PDU o A_GroupValue_Response-PDU o A_GroupValue_Write-PDU	[19] - All end devices - Routers - Bridge [15] - RF bidirectional end device
• **Data length**	
[13] data must be coded as indicated in §3.1 “Application Layer Services on Multicast Communication Mode”	[19] - All end devices - Routers - Bridge [15] - RF bidirectional end device

2.8.3.14 Application Layer – Property Value Services

Specification	Test
• **APDU**	
[13] - §2 “APDU” - §3.4.3.1 “A_PropertyValue_Read-service” - §3.4.3.2 “A_PropertyValue_Write-service”	

2.8.3.15 Application Layer – Function Property services

Specification	Test
• **APDU**	
[13] - §2 “APDU” - §3.4.5 “Function Property Services”	

2.8.3.16 Application Interface Layer – Group Objects

Specification	Test
• **Group Objects**	
[14] - §3 “Group Object Server”	[18] - §1.3 “Group Object Tests” - all end devices [15] - - RF bidirectional end device

2.8.3.17 Application Interface Layer – Interface Objects

Specification	Test
• **Interface Objects**	
[14] - §4 “Interface Object Server”	

2.8.3.18 Application Interface Layer – Function Properties

Specification	Test
• **Function Properties**	
[14] - §4.4.2 “Function Properties”	

2.8.3.19 Application Layer Interface Layer – Group Objects indirection

Specification	Test
• **Group Object Indirection**	
[14] - §3.4 “Group Object Indirection – Group Object Handles and PID-OBJECT_-VALUE (PID = 62) [03] - §4.3.13 “PID_OBJECT_VALUE (PID = 62)”	

2.8.3.20 Application Interface Layer for unidirectional devices

Specification			Test
		• **APDU**	
[13]	-	§2 “APDU”	[19]
	-	§3.1.3 “A_GroupValue_Write-service”: Server side only	
		• **Data length**	
[13]	-	§3.1 “Application Layer services on Multicast Communication Mode”: data shall be encoded as indicated in this referred clause.	[19]
		• **Connection Codes**	
	-	Datapoint Types shall comply with Connection Codes used.	[19]
		• **Application Interface Layer for unidirectional devices**	
[14]	-	§3.3.4 “Writing the Group Object Value”	

2.8.4 Specific parts

	Coupler Model 2.0		Mask 0920h		Mask 2920h	
Feature	Primary Side	Secondary Side	Primary Side	Secondary Side	Primary Side	Secondary Side
1 Link Layer – polling	O	O	O	O	O	n/a
2 Detection of Usage of own Individual Address	X	X	X	X	X	X
3 Extended Group Object Flags	O	O	O	O	O	O

2.8.5 Configuration and management

This clause solely focuses on the Coupler Profiles.
The contents of this AN solely concerns the “Coupler Model 2.0”, in blue text. The indications for the other Coupler Profiles are only for reference.

2.8.5.1 Communication

2.8.5.2 Device Management

2.8.5.2.1 Overview

Please note that the “Coupler services” are not allowed for the Coupler Model 2.0.

Feature	Coupler Mode 2.0	Mask 0920h	Mask 2920h
1 Direct memory Access	C [9)]	C [9)]	C [9)]
2 DMA on User Memory	C [10]	C [10)]	C [10)]
3 Coupler services	X	X	X
4 Verify Mode	M	M	M
5 Interface Object Handling [11)]	M	M	M
6 Reduced Interface Objects	X	X	X
7 Function Properties	M	M	M
8 Interface Object Index discovery	M	M	M
9 Load and Run State Machines			
a. Realisation Type 1	M	M	M
b. Realisation Type 2	X	X	X
10 Restart			
a. connectionless	M	M	M
b. connection-oriented	M	M	M
c. Master Reset	M	M	M [12)]
11 Authorization [13)]	M	M	M
nr of access levels	4	4	4

9) This is mandatory if the Filter Table is located (partially) in memory **below** FFFFh.

10) This is mandatory of the Filter Table is located (partially) in memory **above** FFFFh.

11) Please refer to Annex A for the specification of mandatory and optional Interface Objects, Properties and Property fields.

12) Seen the open nature of the KNX RF Communication Medium on the Secondary Side, it is a security risk if Master Reset would be supported unauthorized. Therefore, it is recommended to support Master Reset only in combination with KNX Data Security (Roles and Permissions).

13) The support of the A_Authorize- and the A_Keywrite-service does not imply that the device itself has access protected areas. If this is not the case, a device shall always allow – regardless of the attributed keys – access to the highest level (0), including when receiving an illegal key (‘illegal’ in this sense meaning another key than any of the keys entered in the key table).

2.8.5.2.2 Direct Memory Access

Specification	Test
[04] - §3.2.1 “DMP_Connect_RCo” - §3.3.2 “DMP_Disconnect_RCo” - §3.18.2 “DMP_MemRead_RCo” - §3.16.2 “DMP_MemWrite_RCo”	- - §2 “Network Management Server Tests” corresponding tests

2.8.5.2.3 DMA on user memory

Specification	Test
[04] - §3.19 “DM_UserMemWrite”	- - §2 “Network Management Server Tests” corresponding tests

2.8.5.2.4 Coupler services

Coupler services are not allowed for the Coupler Model 2.0 and the derived masks.

Specification	Test
[04] - §3.34 “DM_LCSlaveMemWrite” - §3.35 "DM_LCSlaveMemVerify” - §3.36 “DM_LCSlaveMemRead” - §3.37 “DM_LCExtMemWrite” - §3.38 “DM_LCExtMemVerify” - §3.39 “DM_LCExtMemRead” - §3.40 “DM_LCExtMemOpen” - §3.41 “DM_LCRouteTableStateWrite” - §3.42 “DM_LCRouteTableStateVerify” - §3.43 “DM_LCRouteTableStateRead”	

2.8.5.2.5 Verify Mode (for A_Memory_Write)

Specification	Test
[03] - §4.2.14.7 "Verify Mode Control" management server part.	[18] - §2 ”Network Management Server Tests” Tests 7.1 to 7.7

2.8.5.2.6 Interface Object Handling

Please refer to 2.8.6 for the specification of mandatory and optional Interface Objects, Properties and Property fields.

Specification	Test
[04] - §3.22.2 “DMP_InterfaceObjectWrite_R” - §3.24.2 “DM_InterfaceObjectRead_R” - §3.23.2 [20061102ac] “DMP_InterfaceObjectVerify_R” - §3.25.2 “DM_InterfaceObjectScan_R” [20061102ac]	[18] - §2 “Network Management Server Tests” corresponding tests

Specification	Test
[14] - §4 “Interface Object Server”	[18] - §2 “Network Management Server Tests” corresponding tests

2.8.5.2.7 Reduced Interface Objects

Reduced Interface Objects are not allowed for the Coupler Model 2.0 and the derived masks.

Specification	Test
[04] - §3.22.2 “DMP_ReducedInterface-ObjectWrite_R” - §3.24.3 “DMP_ReducedInterfaceObject-Read_R” - §3.25.3 “DMP_ReducedInterfaceObjectScan_R”	[18] - §2 “Network Management Server Tests” corresponding tests
[14] - §4.1 “Common structure” - §4.3.2 “Reduced Interface Object”	[18] - §2 “Network Management Server Tests” corresponding tests

2.8.5.2.8 Function Properties

Specification	Test
[13] - §2 “APDU” - §3.4.1 “Introduction” - §3.4.5 “Function Property Services”	
[14] - §4.1 “Common structure” - §4.2 “Minimal requirements of Interface Objects” - §4.4.2 “Function Properties”	
[04] - §3.26 “DM_FunctionProperty_Write_R”	

2.8.5.2.9 Interface Object Index discovery

Specification	Test
[20] - §2.4.1 “NM_ObjectIndex_Read”	To be defined.

2.8.5.2.10 Load - and Run State Machines

a) Realisation Type 1 - Property based	
Specification	**Test**
[03] - records	
[04] - §3.27.3 “DMP_LoadStateMachineWrite_-Rco_IO” - §3.28.3 “DM_LoadStateMachineVerify_-R_IO” - §3.29.3 “DMP_LoadStateMachineRead_-R_IO” - §3.30.3 “DMP_RunStateMachineWrite_-R_IO” - §3.31.3 “DMP_RunStateMachineVerify_-R_IO” - §3.32.3 “DMP_RunStateMachineRead_-R_IO”	[18] - §2 “Network Management Tests” corresponding tests

The Load – and Run State Machines – Realisation Type 2 are not allowed for the Coupler Model 2.0 and the derived masks.

b) Realisation Type 2 – Memory mapped	
Specification	**Test**
[03] records	
[04] - §3.27.2 “DMP_LoadStateMachineWrite_Rco_Mem” - §3.28.2 “DMP_LoadStateMachineVerify_Rco_Mem” - §3.29.2 “DMP_LoadStateMachineRead_Rco_Mem” - §3.30.2 “DMP_RunStateMachineWrite_Rco_Mem” - §3.31.2 “DMP_RunStateMachineVerify_Rco_Mem” - §3.32.2 “DMP_RunStateMachineRead_Rco_Mem”	[18] - §2 “Network Management Tests” corresponding tests

2.8.5.2.11 Restart

2.8.5.2.11.1 Restart connectionless

Specification	Test
[04] - §3.7.2 DM_Restart_RCl	

2.8.5.2.11.2 Restart connection-oriented

Specification	Test
[04] - §3.2.1 DMP_Connect_RCo - §3.7.3 DM_Restart_RCo	- §2.9.1 “Send Restart”

2.8.5.2.11.3 Master Reset

Specification	Test
[04] - §3.7.2 “DM_Restart_RCl” – master reset - §3.7.3 “DM_Restart_RCo” – master reset	

2.8.5.2.11.3.1 Erase Codes

Erase Code	Coupler Model 2.0	mask 0920h	mask 2920h
Confirmed Restart	M	M	M
Factory Reset	O	O	O
ResetIA	O	O	O
ResetAP	O	O	O
ResetParam	O	O	O
ResetLinks	O	O	O
Factory Reset without IA	O	O	O

2.8.5.2.12 Authorization

Specification	Test
• **Management Procedures**	
[04] - §3.5.1 “DM_Authorize_RCo” - §3.5.2 “DM_Authorize2_RCo” - §3.6 “DM_SetKey”	

2.8.5.3 Device Identification

Feature	Coupler Model 2.0	mask 0920h	mask 2920h
1 Device Descriptor Service - connection oriented	M	M	M
2 Device Descriptor Service - connectionless	O	O	O
3 Device Descriptor Type 0	M	M	M

2.8.5.3.1 Device Descriptor Service - connection oriented

Specification	Test
[13] - §3.4.2.1 “A_DeviceDescriptor_Read-service”	[18] - §2.5.1 “Read Mask-version” (Network Management Tests)

2.8.5.3.2 Device Descriptor Service - connectionless

Specification	Test
[13] - §3.4.2.1 “A_DeviceDescriptor_Read-service”	[18] - §2.5.1 “Read Mask-version” (Network Management Tests)
[04] - §3.2.2 “DMP_Connect_RCI”	

2.8.5.3.3 Device Descriptor Type 0

Specification	Test
[03] - §4.1.2 “Device Descriptor Type 0”	[18] - §2.5.1 “Read Mask-version” (Network Management Tests)

2.8.5.4 Device Individualisation

Feature	Coupler	Coupler Model 2.0	mask 0920h	mask 2920h
1 Programming Mode				
1.a Connection oriented	M	M	M	M
1.b Connectionless	O	O	O	O
2 KNX Serial Number				
a client initiated	O	M	M	M
b server initiated	O	O	O	O
3 Domain Address Assignment	C*	C*	n/a	M
4 Distributed Address Assignment	X	X	X	X
5 Default Individual Address	O	M	M	M
6 SNA Server	O	M	M	M

* mandatory on open media

2.8.5.4.1 Programming Mode

2.8.5.4.1.1 connection oriented

Specification	Test
[04] - §2.2 “NM_IndividualAddress_Read” - §2.3 “NM_IndividualAddress_Write” [14]	[18] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”
Programming Mode Control • via HMI: device selection and indication of Programming Mode • via bus: [03] - §4.19.3 “Programming Mode – Realisation Type 2” [04] - §3.13.2 “DMP_ProgModeSwitch_RCo”	[18] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”

2.8.5.4.1.2 Programming Mode – connectionless

Specification	Test
[04] - §2.10 “NM_DomainAnd-IndividualAddress_Write2” - §2.2 “NM_IndividualAddress_Read”	[18] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”
Programming Mode Control • via HMI: device selection and indication of Programming Mode [04] - §3.13.2 “DMP_ProgModeSwitch_RCo” [03] - §4.19.3 “Programming Mode – Realisation Type 2”	[18] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”

2.8.5.4.2 KNX Serial Number

2.8.5.4.2.1 Client initiated

Specification	Test
[04] - §2.4 “NM_IndividualAddress_Serial-Number_Read” - §2.5 “NM_IndividualAddress_Serial-Number_Write”	[18] - §2.16 “Testing of A_IndividualAddress-SerialNumber_Write-Service : Server Test” - §2.17 “Testing of A_IndividualAddress-SerialNumber_Read-Service : Server Test”

2.8.5.4.2.2 Server initiated

Specification	Test
[04] - §2.6 “NM_IndividualAddress_Serial-Number_Write2	[18] - §2.16 “Testing of A_IndividualAddress-SerialNumber_Write-Service : Server Test”

[14] Implies connection-oriented TL and Application Layer services for accessing the Device Descriptor.

2.8.5.4.3 Domain Address Assignment

	Specification	Test
[04]	- §2.7 “NM_DomainAddress_Read” - §2.12 “NM_DomainAddress_Scan”	[18]

2.8.5.4.4 Distributed Address Assignment

This feature is not allowed for any Coupler Profile.

2.8.5.4.5 Default Individual Address

	Specification	Test
[03]	- §3.3 “Individual Address” - Subnetwork Address, value according to the medium. - Device Address, fixed value FFh	

2.8.5.4.6 SNA Server

	Specification	Test
[03]	• **Device Object** - PID_SUBNET_ADDR • **Router Object** - PID_COUPL_SERV_CONTROL	
[04]	- §2.18 General Procedure “NM_NetworkParameter_Read_R” - §2.17.1 “NM_NetworkParameter_Write_R”	
[05]	- §1.3.3 “SNA read” – Management Server side support - §1.3.4 “SNA update on IA change” - §1.3.5 “SNA update on power-up” (optional) - §1.3.6 “SNA heartbeat”	

2.8.5.5 Device Linking

	Feature	Coupler Model 2.0	mask 0920h	mask 2920h
1	Filter Table			
	1.1 Realisation Type 3	M	M	M

2.8.5.5.1 Filter Table – Realisation Type 3

Specification	Test
§2.4.3 “Filter Table Realisation Type 3”	

2.8.5.6 Application Handling

This feature is no standard feature of any of the approved Coupler Profiles.

2.8.6 Interface Objects and Properties

2.8.6.1 Interface Objects

S-Mode Profiles → Interface and Couplers → Interface Objects

NOTE 15 The Coupler Model 2.0, mask 0920h and mask 2920h do not require fix Object Indexes. Only the general rule applies that the Device Object shall have Object Index 0. The MaC may discover the Object Indexes using NM_ObjectIndex_Read.

	Coupler Model 2.0		mask 0920h		mask 2920h	
Object Type / Interface Object	Primary Side	Secondary Side	Primary Side	Secondary Side	Primary Side	Secondary Side
0 Device Object	M		M		M	
1 Addresstable Object	O		O		O	
2 Associationtable Object	O		O		O	
3 Applicationprogram Object	O		O		O	
4 Interfaceprogram Object	O		O		O	
6 Router Object	M	M	M	M	M	M
7 LTE Address Filter Table Object	X	M	X	M	X	M
8 cEMI Server Object	O		O		O	
9 Group Object Table Object	O		O		O	
10 Polling Master	O	O	O	O	O	n/a
11 KNXnet/IP Parameter Object [15)]	O		O		O	
17 Security Interface Object	O		O		O	
19 RF Medium Object	M		M		M	

15) The KNXnet/IP Parameter Object is mandatory if there is a KNXnet/IP interface.

2.8.6.2 Device Object (Object Type = 0)

2.8.6.2.1 Overview

S-Mode Profiles → Interface and Couplers → Device Object

	Property Identifier	Property Service type		Read/Write	M/O	Access Level	Role	Security
1	PID_OBJECT_TYPE	Data	Plain	Read	M	3	any	n/a
				Write	X	n/a	n/a	n/a
	White List: M		Secure	Read	O	3	any	plain
	Black List: X			Write	O	0	manuf	A+C
	Intermediate List: X	NwPar	Plain	Read	M	3	any	plain
5	PID_LOAD_STATE_CONTROL	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	M	2	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	2	ETS	A
9	PID_FIRMWARE_REVISION [16]	Data	Plain	Read	X	3	n/a	n/a
	White List: X			Write	X	X	n/a	n/a
	Black List: X		Secure	Read	X	3	n/a	n/a
	Intermediate List: X			Write	X	1	n/a	n/a
11	PID_SERIAL_NUMBER	Data	Plain	Read	M	3	any	n/a
	White List: X			Write	O	1	OEM	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	1	OEM	A
12	PID_MANUFACTURER_ID	Data	Plain	Read	M	3	n/a	n/a
	White List: M			Write	O	1	OEM	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	1	OEM	A
14	PID_DEVICE_CONTROL	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	M	2	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	2	ETS	A

16) PID_FIRMWARE_REVISION is not allowed. Instead, PID_VERSION shall be used.

	Property Identifier	Property Service type		Read/Write	M/O	Access Level	Role	Security
15	PID_ORDER_INFO	Data	Plain	Read	O	3	n/a	n/a
	White List: X			Write	O	1	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	1	OEM	A
19	PID_MANUFACTURER_DATA	Data	Plain	Read	O	3	n/a	n/a
	White List: X			Write	O	1	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	1	OEM	A
25	PID_VERSION	Data	Plain	Read	O	3	n/a	n/a
	White List: X			Write	O	1	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	1	OEM	A
30	PID_DOWNLOAD_COUNTER	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	X	n/a	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	1	OEM	A
51	PID_ROUTING_COUNT	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	M	2	n/a	n/a
	Black List: X		Secure	Read	M	3	any	plain
	Intermediate List: X			Write	M	2	ETS	A
54	PID_PROGMODE	Data	Plain	Read	M	3	n/a	n/a
	White List: M			Write	M	2	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	2	ETS	A
56	PID_MAX_APDU_LENGTH	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	M	0	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	0	manuf	A
57	PID_SUBNET_ADDR	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	X	n/a	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	X	n/a	n/a	n/a
		NwPar		Read	M	n/a	n/a	n/a
				Write	X	n/a	n/a	n/a

	Property Identifier	Property Service type		Read/Write	M/O	Access Level	Role	Security
58	PID_DEVICE_ADDR	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	X	n/a	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	X	n/a	n/a	n/a
		NwPar		Read	M	n/a	n/a	n/a
				Write	X	n/a	n/a	n/a
71	PID_IO_LIST	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	O	0	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	0	manuf	A
		NwPar		Read	M	n/a	n/a	n/a
				Write	X	n/a	n/a	n/a
71	PID_HARDWARE_TYPE	Data	Plain	Read	M	3	n/a	n/a
	White List: X			Write	O	0	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	O	0	manuf	A
83	PID_DEVICE_DESCRIPTOR	Data	Plain	Read	M	3	n/a	n/a
	White List: M			Write	X	n/a	n/a	n/a
	Black List: X		Secure	Read	O	3	any	plain
	Intermediate List: X			Write	X	n/a	n/a	n/a

2.8.6.2.2 PID_LOAD_STATE_CONTROL (PID = 5)

*S-Mode Profiles → Interface and Couplers → **Device Object** → PID_LOAD_STATE_CONTROL*

The *Additional Load Controls* are not allowed for the Coupler Model 2.0.

Load Control	Sub-type	Description	Coupler Model 2.0	mask 0920h	mask 2920h
00h		No operation	O	O	O
01h		Start Loading	M	M	M
02h		Load Completed	M	M	M
03h		Additional Load Controls	X	X	X
	00h	Absolute Code/Data Allocation	X	X	X
	01h	Absolute Stack Allocation	X	X	X
	02h	**Segment Control Record**	X	X	X
	03h	**Task Pointer Record**	X	X	X
	04h	**Task Control Record-1**	X	X	X
	05h	**Task Control Record-2**	X	X	X
	0Ah	Relative Allocation	X	X	X
	0Bh	Data Relative Allocation	X	X	X
04h		Unload	M	M	M

The following Load States may be returned.

Value	Load State	Coupler Model 2.0	mask 0920h	**mask 2920h**
00h	Unloaded	M	M	M
01h	Loaded	M	M	M
02h	Loading	M	M	M
03h	Error	M	M	M
04h	Unloading	O	O	O
05h	LoadCompleting	O	O	O

NOTE 16 The Load States Unloading and LoadCompleting are optional and not forbidden. This means that the MaC shall take into account that the Coupler Model 2.0 implementation may return these Load States.

2.8.6.2.3 PID_DEVICE_CONTROL (PID = 14)

Bit# Bit function	Coupler Model 2.0	mask 0920h	mask 2920h
0User stopped	n/a	n/a	n/a
1Individual Address duplication	n/a	n/a	n/a
2Verify Mode On	M	M	M
3Safe State On	n/a	n/a	n/a
4reserved	n/a	n/a	n/a
5reserved	n/a	n/a	n/a
6reserved	n/a	n/a	n/a
7reserved	n/a	n/a	n/a

2.8.6.3 Router Object (Object Type = 6)

2.8.6.3.1 Overview

S-Mode Profiles → Interface and Couplers → Router Object

PID	Property		Coupler Model 2.0		mask 0920h		mask 2920h	
			Primary Side	Secondary Side	Primary Side	Secondary Side	Primary Side	Secondary Side
1	PID_OBJECT_TYPE	Data	3/1	3/1	3/1	3/1	3/1	3/1
		NwPar	R	R	R	R	R	R
5	PID_LOAD_STATE_CONTROL	Data	X	3/2	X	3/2	X	3/2
7	PID_TABLE_REFERENCE	Data	X	3/x	X	3/x	X	3/x
27	PID_MCB_TABLE	Data	X	3/x	X	3/x	X	3/x
28	PID_ERROR_CODE	Data	X	(3/X)	X	(3/X)	X	(3/X)
29	PID_OBJECT_INDEX	Data	(3/X)	(3/X)	3/0	3/0	3/0	3/0
		NwPara	R	R	R	R	R	R
51	PID_MEDIUM_STATUS	Data	3/x	3/x	3/x	3/x	3/x	3/x
		NwPar	RT	RT	RT	RT	RT	RT
54	PID_MAIN_LCGRPCONFIG	Data	X	X	X	X	X	X
55	PID_SUB_LCGRPCONFIG	Data	X	X	X	X	X	X
56	PID_ROUTETABLE_CONTROL	Data	X	3/2	X	3/2	X	3/2
57	PID_COUPL_SERV_CONTROL	Data	3/2	3/2	3/2	3/2	3/2	3/2
58	PID_MAX_APDU_LENGTH	Data	3/0	3/0	3/0	3/0	3/0	3/0
61	PID_HOP_COUNT	Data	(3/2)	(3/2)	n/a	n/a	3/2	n/a
63	PID_MEDIUM	Data	3/0	3/0	3/0	3/0	3/0	3/0
67	PID_FILTER_TABLE_USE	Data	X	3/2	X	3/2	X	3/2
104	PID_PL110_ENABLE_SBC [17)]	Data	3/2 [17)]	3/2 [17)]	n/a	3/2	n/a	n/a
105	PID_PL110_DOA	Data	n/a	3/2	n/a	3/2	n/a	n/a
112	PID_RF_ENABLE_SBC [18)]	Data	3/2 [18)]	3/2 [18)]	n/a	n/a	n/a	3/2

[17)] PID_PL110_ENABLE_SBC and PID_PL110_DOA are only mandatory in Medium Interfaces to the PL110 medium.

[18)] PID_RF_ENABLE_SBC is only mandatory in Medium Interfaces to the RF medium.

2.8.6.3.2 PID_LOAD_STATE_CONTROL (PID = 5)

*S-Mode Profiles → Interface and Couplers → **Router Object** → PID_LOAD_STATE_CONTROL*

Load Control	Sub-type	Description	Coupler Model 2.0	mask 0920h	mask 2920h
00h		No operation	O	O	O
01h		Start Loading	M	M	M
02h		Load Completed	M	M	M
03h		Additional Load Controls	M	M	M
	00h	Absolute Code/Data Allocation	O	O	O
	01h	Absolute Stack Allocation	O	O	O
	02h	Segment Control Record	O	O	O
	03h	Task Pointer Record	O	O	O
	04h	Task Control Record-1	O	O	O
	05h	Task Control Record-2	O	O	O
	0Ah	Relative Allocation	O	O	O
	0Bh	Data Relative Allocation	M	M	M
04h		Unload	M	M	M

The following Load States may be returned.

Value	Load State	Coupler Model 2.0	mask 0920h	mask 2920h
00h	LS_UNLOADED	M	M	M
01h	LS_LOADED	M	M	M
02h	LS_LOADING	M	M	M
03h	LS_ERROR	M	M	M

2.8.6.4 LTE Address Filter Table Object (Object Type = 0007h)

S-Mode Profiles → Couplers and Interfaces → LTE Address Filter Table Object

	Property		Coupler Model 2.0	mask 0920h	**mask 2920h**
1	PID_OBJECT_TYPE	Data	3/x	3/x	3/x
5	PID_LOAD_STATE_CONTROL	Data	3/0	3/0	3/0
51	PID_LTE_ROUTESELECT	Data	3/0	3/0	3/0
52	PID_LTE_ROUTETABLE	Data	3/0	3/0	3/0

2.8.6.4.1 PID_LOAD_STATE_CONTROL (PID = 5)

S-Mode Profiles → Interface and Couplers → LTE Address Filter Table Object → PID_LOAD_STATE_CONTROL

Value	Load Control	Coupler Model 2.0	mask 0920h	**mask 2920h**
00h	EV_NOP	M	M	M
01h	EV_START_LOAD	M	M	M
02h	EV_LOAD_COMPLETE	M	M	M
03h	EV_ADDITIONAL	X	X	X
04h	EV_UNLOAD	M	M	M

The following Load States may be returned.

Value	Load State	Coupler Model 2.0	mask 0920h	**mask 2920h**
00h	LS_UNLOADED	M	M	M
01h	LS_LOADED	M	M	M
02h	LS_LOADING	M	M	M
03h	LS_ERROR	M	M	M

2.8.6.5 RF Medium Object (Object Type = 19)

S-Mode Profiles → Couplers → RF Medium Object

PID	Property		**Mask 2920h**
1	PID_OBJECT_TYPE	Data	3/1
51	PID_RF_MULTI_TYPE	Data	3/2
56	PID_RF_DOMAIN_ADDRESS	Data	3/2
57	PID_RF_RETRANSMITTER	Data	(3/2)
60	PID_RF_BIDIR_TIMEOUT	Data	X
61	PID_RF_DIAG_SA_FILTER_TABLE	Data	3/2
62	PID_RF_DIAG_BUDGET_TABLE	Data	3/2

2.8.7 User Interface

2.8.7.1 Reset to factory default

The Coupler Model 2.0, mask 0920h and mask 2920h shall have a way in the HMI to reset to the factory default state.

2.9 Identifiers and discovery

This clause is not intended for integration in the KNX Specifications.

The new masks introduced in this document are identified by the mask versions 0920h and 2920h.

Prior to downloading, the Manufacturer Code (PID_MANUFACTURER_ID) and the Hardware Type (PID_HARDWARE_TYPE) will be used for further identification of the download.

The Primary Side and the Secondary Side are represented by Interface Objects of the same Object Type “Router Object”. The MaC shall assume that the instance with the lowest Object Index represents the Router Object of the Primary Side.

3 Impact and dependencies

3.1 System specification (“Handbook”) dependencies

This clause is not intended for integration in the KNX Specifications.

In the current structure of the KNX Specifications, the specifications for a given Profile are spread over multiple documents and the existing Profiles specifications ([08]) only refers to the features in these papers by name. There is no single paper that gives for one Profile a description and an introduction to its key features.

EXAMPLE 8 System B is introduced in AN057. However, this is integrated and spread over the KNX Specifications and there is no longer a single paper that gives an introduction and overview to System B.

Additionally, it is increasingly complex to add further Profiles and features to the current Profile specifications. For this reason, already in the current version, “End Devices” and “Couplers and Interfaces” have been separated and already “Special Profiles”, like the cEMI Server have been introduced.

Therefore, it is the intention to re-organise Volume 6 “Profiles” and to foresee a normative paper for each device Profile. The contents of this paper is intended to establish a first such Profile. The exact Part - and Chapter number have to be concluded by KNX Association.

3.2 Configuration interworking

[Expected contents of this clause: Derived requirements (media couplers, gateways, ;..)

Constraints (media, existing media couplers & gateways, …)

Conflicts (existing implementations, …)]

[Applicability of this clause: For this clause, the mention of “Not Applicable on the following grounds” is allowed only with substantial and adequate motivation.]

3.3 Run-time Interworking

3.3.1 DPT_Medium

This clause shall be integrated in Chapter 3/7/2 "Datapoint Types" ([07]).

Format:	1 octet: N_8
octet nr.	1
field names	*field1*
encoding	N N N N N N N N
Encoding:	Encoding absolute value N = [0 … 255]
Unit:	none
Resol.:	none
PDT:	PDT_ENUM8 (alt: PDT_UNSIGNED_CHAR)

Datapoint Types				
ID:	Name:	Encoding:	Range:	Use:
20.1004	DPT_Medium	*field1* = KNX Medium 0 : KNX TP1 1 : KNX PL110 2 : KNX RF 3 : reserved. Shall not be used. 4 : reserved. Shall not be used. 5 : KNX IP 63 to 255 : not used; reserved	{0, 1, 2, 5}	FB

3.4 Registration and certification

[Expected contents of this clause: This clause mainly focuses on the timing and the conditions for registration and certification of new, changed or abandoned KNX features. This should be given here for each feature. (Features may be grouped). This should or may make a difference between

- *registration and certification*
- *stack versus application*
- *not allowed (anymore), mandatory (since)*

This should also indicate transition periods.

EXAMPLE

- *Feature 1*

Not allowed for new registrations or certifications after the transitional period.

- *Feature 2 and 3*

Mandatory for renewed registrations of existing applications

- *Feature 3*

Forbidden for new registrations of stacks after a transitional period of …

3.5 Integration and common tool impact

3.5.1 Inheritance from legacy Coupler implementations

In the Coupler Object, this Coupler Model 2.0 does not re-use all of the Properties of the legacy Coupler Type. ETS shall not automatically assume the presence of any Property in the Coupler Object, neither for the Coupler Model 2.0 nor for any legacy Coupler Profile. Instead, ETS shall relate this to the value of the Device Descriptor Type 0 (mask version) and based on this assume the different Properties.

3.5.2 Topology view

The Coupler Model 2.0 is no longer considered to support more than one Secondary Side and hence has no special topology constraints, different from the existing Coupler Profiles.

3.5.3 Parameter Filter Table Use and Filter Table LSM

ETS needs to know whether the installed Coupler Model 2.0 implementation evaluates the FT or not in the following use cases.

- Download: if the device does not evaluate the FT then ETS can shorten the download time by omitting the FT download.
- *DeviceInfo*: suppress read and display of the FT if it is not evaluated or if the LSM of the FT does not have the state *Loaded*.
- *DeviceCompare*: suppress compare of FT if the FT is not evaluated or if the LSM of the FT does not have the state *Loaded*.
- The diagnostic wizard
  - Suppress compare of FT if the FT is not evaluated or if the LSM of the FT does not have the state *Loaded*.
  - The wizard has a function to temporarily add a GA to all FTs between ETS and a target device. This can also be suppressed if the FT is not evaluated.
- Reconstruction
  - The FT does not need to be read out if it is not evaluated or if the LSM of the FT does not have the state *Loaded*.

So ETS needs some way to discover the filtering disposition. As the FT evaluation is influenced by implementation specific parameters, ETS cannot interpret these parameters to conclude on this. Therefore, a special parameter “Filter Table use” in the Coupler Model 2.0 product data shall be foreseen. This shall be the downloadable Property based parameter “Filter Table Evaluation”.

This parameter shall have the following possible values.

0	=	FALSE	The Filter Table is not used.
1	=	TRUE	The Filter Table is used.

ETS shall always download this Parameter.

The parameter can either be presented directly in the ETS parameter dialog, or be calculated from other parameters, using ETS’ “calculated parameters” feature.

3.6 Risks and compatibility issues

Annex A
(informative)

Comparison mask 0912h and Coupler Model 2.0

A.1 Device Object

A.1.1 Comparison Properties

	Mask 0912h		Coupler Model 2.0		REMARKS
	Property Identifier (PID)	**Property Datatype**	**Property Identifier (PID)**	**Property Datatype**	
1	PID_OBJECT_TYPE	PDT_UNSIGNED_INT	PID_OBJECT_TYPE	PDT_UNSIGNED_INT	DEVICE_OBJECT: 0000h
8	PID_SERVICE_CONTROL	PDT_UNSIGNED_INT	PID_SERVICE_CONTROL	PDT_UNSIGNED_INT	Permanent control field of the device
9	PID_FIRMWARE_REVISION	PDT_UNSIGNED_-CHAR	not used	not used	revision number of the firmware
25	not used	not used	PID_VERSION	PDT_VERSION	
11	PID_SERIAL_NUMBER	PDT_GENERIC_06	PID_SERIAL_NUMBER	PDT_GENERIC_06	KNX Serial Number
12	PID_MANUFACTURER_ID	PDT_UNSIGNED_INT	PID_MANUFACTURER_ID	PDT_UNSIGNED_INT	manufacturer identifier
14	PID_DEVICE_CONTROL	PDT_GENERIC_01	PID_DEVICE_CONTROL	PDT_GENERIC_01	temporary control field for the device
15	PID_ORDER_INFO	PDT_GENERIC_10	PID_ORDER_INFO	PDT_GENERIC_10	Same
19	PID_MANUFACTURER_DATA	PDT_GENERIC_04	PID_MANUFACTURER_DATA	PDT_GENERIC_04	Same
51	PID_ROUTING_COUNT	PDT_UNSIGNED_-CHAR	PID_ROUTING_COUNT	PDT_UNSIGNED_CHAR	default hop count
53	PID_ERROR_FLAGS	PDT_UNSIGNED_-CHAR	not used	not used	Not used in the Coupler Model 2.0.
54	PID_PROGMODE	PDT_BITSET8	PID_PROGMODE	PDT_BITSET8	O in mask 0912h M in Coupler Model 2012

	Mask 0912h		Coupler Model 2.0		REMARKS
	Property Identifier (PID)	Property Datatype	Property Identifier (PID)	Property Datatype	
56	PID_MAX_APDULENGTH	PDT_UNSIGNED_INT	PID_MAX_APDULENGTH	PDT_UNSIGNED_INT	O in mask 0912h M in Coupler Model 2012
57	PID_SUBNET_ADDR	PDT_UNSIGNED_-CHAR	PID_SUBNET_ADDR	PDT_UNSIGNED_CHAR	SubNetAddress

A.2 Router Object

	Mask 0912h		Coupler Model 2.0		REMARKS
	Property Identifier (PID)	Property Datatype	Property Identifier (PID)	Property Datatype	
1	PID_OBJECT_TYPE	PDT_UNSIGNED_INT	PID_OBJECT_TYPE	PDT_UNSIGNED_INT	DEVICE_OBJECT: 0000h
5	PID_LOAD_STATE_CONTROL	PDT_CONTROL	PID_LOAD_STATE_CONTROL	PDT_CONTROL	
7	Does not exist	Does not exist	PID_TABLE_REFERENCE	PDT_UNSIGNED_LONG	Pointer to the filter table
29	Does not exist	Does not exist	PID_OBJECT_INDEX	PDT_UNSIGNED_CHAR	Object Index of the Router Object.
??	Does not exist	Does not exist	PID_PHYS_ROUTE_MASK	PDT_UNSIGNED_INT	Routing mask for Individual Address.
51	PID_LINE_STATUS	PDT_GENERIC_01	PID_MEDIUM_STATUS	PDT_GENERIC_01	Report a power down in the Subnetwork at the Secondary Side.
52	PID_MAIN_LCCONFIG	PDT_GENERIC_01	Does not exist	Does not exist	Defines the handling of individually addressed and broadcast frames from main line.
53	PID_SUB_LCCONFIG	PDT_GENERIC_01	Does not exist	Does not exist	Defines the handling of individually addressed and broadcast frames from sub line.
54	PID_MAIN_LCGRPCONFIG	PDT_GENERIC_01	Does not exist	Does not exist	Defines the handling of group addressed frames from main line.
55	PID_SUB_LCGRPCONFIG	PDT_GENERIC_01	Does not exist	Does not exist	Defines the handling of group addressed frames from sub line

	Mask 0912h		Coupler Model 2.0		REMARKS
	Property Identifier (PID)	**Property Datatype**	**Property Identifier (PID)**	**Property Datatype**	
??	Does not exist	Does not exist	~~PID_ROUTE_CONFIG_IND~~	~~PDT_GENERIC_02[2]~~	~~Defines the handling of individually addressed frames~~
??	Does not exist	Does not exist	~~PID_ROUTE_CONFIG_GRP~~	~~PDT_GENERIC_02[2]~~	~~Defines the handling of group addressed frames~~
??	Does not exist	Does not exist	~~PID_ROUTE_CONFIG_BC~~	~~PDT_GENERIC_02[2]~~	~~Defines the handling of broadcast frames~~
??	Does not exist	Does not exist	~~PID_ROUTE_CONFIG_SBC~~	~~PDT_GENERIC_02[2]~~	~~Defines the handling of system broadcast frames~~
56	PID_ROUTETABLE_CONTROL	PDT_FUNCTION	PID_ROUTETABLE_CONTROL	PDT_FUNCTION	Set of methods to modify the routing table. Changed from Optional to Mandatory
	ServiceID: 1 (SRVID_CLEAR_ROUTINGTABLE)				Same
	ServiceID: 2 (SRVID_SET_ROUTINGTABLE)				Same
	ServiceID: 3 (SRVID_CLEAR_GROUPADDRESS)				Same
	ServiceID: 4 (SRVID_SET_GROUPADDRESS)				Same
57	PID_COUPL_SERV_CONTROL	PDT_GENERIC_01	PID_COUPL_SERV_CONTROL	PDT_GENERIC_01	Inconsistency and Subnetwork Address (SNA) mechanisms.
58	PID_MAX_APDU_LENGTH	PDT_UNSIGNED_INT	PID_MAX_APDU_LENGTH	PDT_UNSIGNED_INT	
59	PID_L2_COUPLER_TYPE	PDT_BITSET8	~~PID_L2_COUPLER_TYPE~~	~~PDT_BITSET8~~	Coupler Model 2.0: M in TP1/TP1 Couplers. X in all other types.
??	Does not exist	Does not exist	PID_ENABLE_SBC	PDT_UNSIGNED_CHAR	Flag to enable temporary the routing of system broadcast from main to subline

A.3 LTE Address FilterObject

	Mask 0912h		Coupler Model 2.0		REMARKS
	Property Identifier (PID)	**Property Datatype**	**Property Identifier (PID)**	**Property Datatype**	
1	PID_OBJECT_TYPE	PDT_UNSIGNED_INT	PID_OBJECT_TYPE	PDT_UNSIGNED_INT	LTE_ADDRESSFILTER_OBJECT: 0007h
5	PID_LOAD_STATE_CONTROL	PDT_CONTROL	PID_LOAD_STATE_CONTROL	PDT_CONTROL	see: load state machines
51	PID_LTE_ROUTESELECT	PDT_GENERIC_01	PID_LTE_ROUTESELECT	PDT_GENERIC_01	defines the general handling of LTE frames
52	PID_LTE_ROUTETABLE	PDT_GENERIC_05[32]	PID_LTE_ROUTETABLE	PDT_GENERIC_05[32]	LTE routing table
??	Does not exist	Does not exist	~~PID_ROUTE_CONFIG_LTE~~	~~PDT_GENERIC_02[2]~~	~~defines the general handling of LTE frames~~

Annex B
(informative)

Overview Coupler Mask Versions

Table 6 – Overview Coupler Mask Versions

Secondary Side		Primary Side			
		0 TP1	**1 PL110**	**2 RF**	**5 IP**
0	**TP1**	0910h 0911h 0912h 0920h [a]		2920h	091Ah
1	**PL110**	1900h			
2	**RF**	(2910h)			
5	**IP**				

[a] Mask 0920h is not yet approved. This is proposed in this document version.

A scheme for the assignment of mask versions for Couplers appears necessary.