# System Specifications

3

## Communication

3

## Data Link Layer General

2

Summary

This document defines the Data Link Layer interface and general definitions for Data Link Layer that are medium-independent.

Medium specific Physical Layer and Data Link Layer services for the various media are specified in dedicated Chapters.

Version 01.02.02 is a KNX Approved Standard.

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

# Document Updates

| Version | Date | Modifications |
|---|---|---|
| 1.0 | 2001.12.17 | Preparation of Approved Standard. |
| 1.1 | 2007.07.26 | • **Supplement 13 "Extended Frame Formats – Data Link Layer General Extension"** |
| | 2007.08.03 | • **Supplement 16 "System Broadcast services"** |
| | 2008.08.08 | • **AN111 "Busload definition and requirements on devices"** integrated. |
| | 2008.09.05 | • **AN106 "Phasing out TP0"** integrated.<br>• **AN109 "Phasing out PL132"** integrated. |
| | 2008.12.19 | Preparation of v1.1 of the Approved Standard. |
| 1.2 | 2009.06.29 | • **AN111 "Busload definition and requirements on devices"** integration extended<br>• Preparation for inclusion in the KNX Specifications v2.0. |
| 1.2.01 | 2009.10.13 | Editorial update. |
| 01.02.02 | 2013.10.28 | Editorial updates for the publication of KNX Specifications 2.1. |

# References

[01] Chapter 3/1/1 "Architecture"
[02] Part 3/2 "Communication Media"
[03] Chapter 3/3/3 "Network Layer"
[04] Part 3/5 "Management"
[05] Chapter 3/5/1 "Resources"
[06] Chapter 3/6/3 "External Message Interface"
[07] Chapter 3/7/1 "Interworking Model"

Filename: 03_03_02 Data Link Layer General v01.02.02 AS.docx
Version: 01.02.02
Status: Approved Standard
Savedate: 2013.10.28
Number of pages: 31

# Contents

**1 General (Overview) ..... 5**
1.1 Functions of the Data Link Layer ..... 5
1.2 Possible Media and their Impact on Layer-2 ..... 6
1.3 Objective ..... 6
1.4 Definitions ..... 6
1.4.1 Domain Address ..... 6
1.4.2 Individual Address ..... 6
1.4.3 Group Address ..... 7

**2 Data Link Layer services ..... 8**
2.1 Introduction and Data Link Layer modes ..... 8
2.2 L_Data service ..... 8
2.2.1 Service specification ..... 8
2.2.2 Usage of ack_request ..... 10
2.2.3 Usage of priority ..... 10
2.2.4 Usage of Frame Format ..... 11
2.2.5 Usage of octet_count ..... 12
2.3 L_SystemBroadcast ..... 13
2.4 L_Poll_Data-service and protocol ..... 15
2.5 L_Busmon Service ..... 16
2.6 L_Service_Information Service ..... 16
2.7 L_Management Service ..... 16

**3 Data Link Layer Protocols ..... 17**
3.1 Protocol ..... 17
3.2 Recommendations for duplication prevention ..... 17

**4 Parameters of Layer-2 ..... 18**

**5 Busload ..... 19**
5.1 Definition of Busload ..... 19
5.2 Definition of Character classes ..... 20
5.3 Measuring busload ..... 21
5.4 Busload generation ..... 22
5.5 Busload types ..... 22
5.5.1 Overview of the busload types ..... 22
5.5.2 Uniform busload ..... 23
5.5.3 Burst busload ..... 23
5.5.4 Mixed busload ..... 23
5.6 Busload composition ..... 23
5.7 Requirements on devices ..... 24
5.7.1 General ..... 24
5.7.2 Introduction ..... 24
5.7.3 Busload levels ..... 24
5.7.4 Busload evaluation Classes for devices ..... 25

**6 Specific devices ..... 29**
6.1 The Layer-2 of the TP1 Bridge and the TP1 Repeater ..... 29
6.2 The Layer-2 of a Router ..... 29
6.3 Coupler model overview (informative) ..... 30

**7 Externally accessible Data Link Layer Interface ..... 31**

# 1 General (Overview)

## 1.1 Functions of the Data Link Layer

The Data Link Layer (also called "Layer-2") shall be the layer between the Data Link Layer user and the Physical Layer. The Data Link Layer shall conform to the definitions of the ISO/OSI model (ISO 7498) Data Link Layer. It shall provide the medium access control and the logical link control.

The Data Link Layer shall be concerned with reliable transport of single Frames between two or more devices on the same Subnetwork.

When transmitting it shall be responsible for:

- building up a complete Frame from the information passed to it by the Network Layer,
- gaining access to the medium according to the particular medium access protocol in use and
- transmitting the Frame to the Data Link Layer in the peer entity or entities, using the services of the Physical Layer.

If the transmission fails, the transmitting Data Link Layer entity may decide to try again after a certain interval. In particular, if the remote device signals that its buffers are temporarily full, the Data Link Layer shall wait for a predetermined time and then attempt to retransmit the Frame (flow control).

When receiving, Data Link Layer shall be responsible for:

- determining whether the Frame is intact or corrupted,
- deciding after Destination Address check to pass the Frame to upper layers,
- issuing positive or negative acknowledgements back to the transmitting Data Link Layer entity.

It shall provide some means to prevent from service duplication (in case of repetitions because of corrupted acknowledgement Frames).

The services provided shall includ both unicast (this is, point-to-point connectionless and - connection-oriented), multicast, broadcast and system broadcast communication options.

The Data Link Layer shall use the services of the Physical Layer and shall provide services to the Data Link Layer user (see Figure 1).

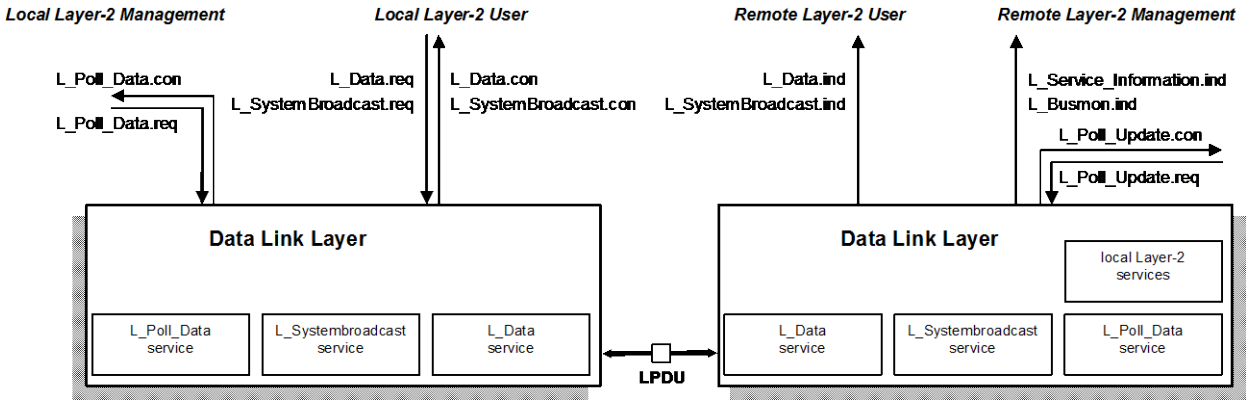



**Figure 1 - Interactivity of the Data Link Layer**

## 1.2 Possible Media and their Impact on Layer-2

Data Link Layer is defined for the following media:

- Twisted Pair 1
- Power Line 110
- Radio Frequency
- IP

The Data Link Layer is open for new media in the future.

Each medium needs a specially dedicated Medium Access Control and a Logical Link Control that adapts to the Medium Access Control.

⇨ Present document will focus on medium independent features, i.e. mainly on the provided service interface to the Network Layer.

⇨ For medium dependent parts, please refer to dedicated Chapters for each medium in [02].

All media support bidirectional communication. In addition, the system allows unidirectional devices.

## 1.3 Objective

Dealing mainly with the service interface, this Chapter is mostly illustrative and giving definitions.

However it specifies the basics for a device to claim exhibiting a standard service interface.

A standard external interface will be completely defined in a dedicated chapter. See [06].

## 1.4 Definitions

### 1.4.1 Domain Address

Every device that connects to an open medium that supports a Domain Address, shall have exactly one Domain Address per connection to such an open medium.

### 1.4.2 Individual Address

Each device, i.e. a Router or an end device, shall have a unique Individual Address (IA) in a network. The Individual Address shall be a 2 octet value that shall consist of an 8 bit Subnetwork Address (SNA) and an 8 bit Device Address (DA).

The Device Address shall be unique within a Subnetwork. Routers shall always have the Device Address zero, i.e. other devices may have the Device Addresses with values 1 to 255. See also [01] for details.

Devices needing no Individual Address for operation may use a default Individual Address. This default Individual Address shall consist of the Device Address for unregistered devices and the medium dependent default Subnetwork Address. Please refer to the specification of the Individual Address space in [05].

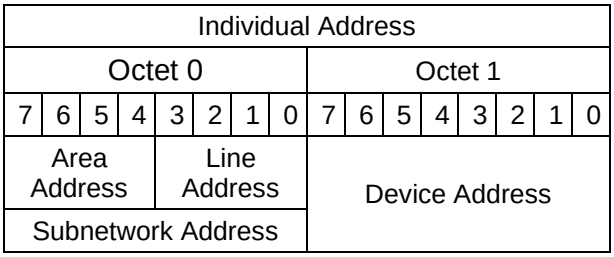



**Figure 2 - Individual Address**

If a hierarchical topology is used to enable an easy configuration of Routers, following recommendation shall apply:

- The Subnetwork Address shall consist of a 4 bit Line Address (LA) and a 4 bit Area Address (AA).
- The Line Address shall be unique within an Area (0 to 15). The devices in the Main Line of an Area shall always have the Line Address zero.
- The Area Address shall be unique within a network (0 to 15). The devices in the inner Zrea shall always have the Area Address zero.

### 1.4.3 Group Address

The Group Address shall be a 2 octet value that does not need to be unique. A Device may have more than one Group Address.

Every device shall belong to group zero, i.e. request Frames with destination Group Address zero shall be broadcasts.

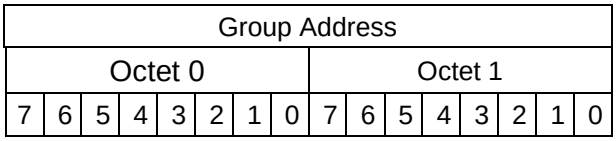



**Figure 3 - Group Address**

Group addresses are defined globally for the whole network. However the specification allows local and global Group Addresses by defining in each Frame the maximum number of Routers to be crossed (see [03]).

See also [04] for table of pre-assigned addresses.

# 2 Data Link Layer services

## 2.1 Introduction and Data Link Layer modes

The Data Link Layer can either be in Normal Mode or in Busmonitor Mode. In Normal Mode the remote L_Data-service, the remote L_SystemBroadcast service, the remote L_Poll_Data-service and the local L_Service_Information-service shall be available to the Data Link Layer user. In Busmonitor Mode only the local L_Busmon-service shall be available. The Data Link Layer mode shall be a parameter of Layer-2.

The Frame effectively sent on the medium (LPDU) is partly medium dependent. Therefore it is described in each Chapter dedicated to a physical medium.

## 2.2 L_Data service

### 2.2.1 Service specification

The L_Data service shall be a Frame transfer service. It shall transmit a single Data Link Layer Service Data Unit (LSDU) to Data Link Layer instance of one or several devices connected to the same Subnetwork. The Destination Address may be an Individual Address or a Group Address (multicast or broadcast). The service may be acknowledged or not, depending on quality of service requested.

There are 3 service primitives:

1. L_Data.req shall be used to transmit a Frame.
2. L_Data.ind shall be used to receive a Frame.
3. L_Data.con shall be a local primitive generated by the local Layer-2 for its own client to indicate that the it is satisfied with the transmission.

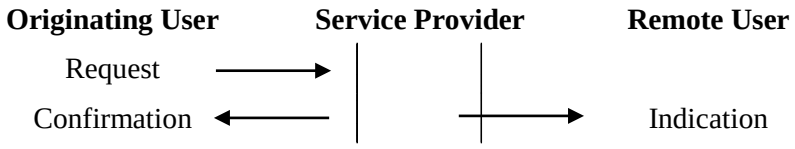



**Figure 4 - Exchange of primitives for the L_Data-service**

If the local user of Layer-2 prepares an LSDU for the remote user it shall apply the L_Data.req primitive to pass the LSDU to the local Layer-2. The local Layer-2 shall accept the service request and shall try to send the LSDU to the remote Layer-2 with the relevant Frame format.

The local Layer-2 shall pass an L_Data.con primitive to the local user that shall indicate either a correct or erroneous data transfer. Depending if an L2-acknowledgement is requested or not, this confirmation shall be related to the reception of the L2-acknowledgement, or only to the transmission of the Frame on the medium.

L_Data.req(ack_request, address_type, destination_address, frame_format, lsdu, octet_count, priority, source_address)

| | | |
|---|---|---|
| ack_request: | This parameter shall be used to indicate whether a Layer-2 acknowledge is mandatory or optional. | |
| address_type: | This parameter shall be used to indicate whether the Destination Address is an Individual Address or a Group Address. | |
| destination_address: | This parameter shall be used to indicate the Destination Address of the Frame to be transmitted; it shall be either an Individual Address or a Group Address. | |
| frame_format: | This parameter shall be used to indicate whether the Frame to be transmitted shall be a standard Frame or an extended Frame. | |
| lsdu: | This parameter shall be used to contain the user data to be transferred by Layer-2. | |
| octet_count: | This parameter shall be used to indicate the length information of the requested Frame. | |
| priority: | This parameter shall be used to indicate the priority that shall be used to the transmit the requested Frame; it shall be “system”, “urgent”, “normal” or “low”. | |
| source_address | Individual Address of the device that requests the L_Data-service. | |

L_Data.con(address_type, destination_address, frame_format, octet_count, priority, source_address, lsdu, l_status)

| | | |
|---|---|---|
| address_type: | This parameter shall indicate whether destination_address is an Individual Address or a Group Address. | |
| destination_address: | This parameter shall be used to indicate the Destination Address of the requested Frame; it shall either be an Individual Address or a Group Address. | |
| frame_format: | This parameter shall be used to indicate whether the transmitted Frame is a standard Frame or an extended Frame. | |
| octet_count: | This parameter shall be used to indicate the length information of the transmitted Frame. | |
| priority: | This parameter shall be used to indicate the priority that is used to transmit the requested Frame; it shall be “system”, “urgent”, “normal” or “low”. | |
| source_address | This parameter shall be used to indicate the Source Address of the requested Frame; it shall be the Individual Address of the device that requests the service primitive. | |
| lsdu: | This parameter shall be the user data transferred by Layer-2. | |
| l_status: | ok: | this value of this parameter shall be used to indicate that the transmission of the LSDU has been successful |
| | not_ok: | this value of this parameter shall be used to indicate that the transmission of the LSDU did not succeed |

L_Data.ind(ack_request, address_type, destination_address, frame_format, lsdu, octet_count, priority, source_address)

| | |
|---|---|
| ack_request: | This parameter shall be used to indicate whether a Layer-2 acknowledge is mandatory or optional. |
| address_type: | This parameter shall indicate whether destination_address is an Individual Address or a Group Address. |
| destination_address: | This parameter shall be used to indicate the Destination Address of this Frame. It shall be the Individual Address of this device or a Group Address of this device. |
| frame_format: | This parameter shall be used to indicate whether the received Frame is a standard Frame or an extended Frame. |
| lsdu: | This parameter shall be used to contain the user data that is received by Layer-2. |
| octet_count: | This parameter shall be used to indicate the length information of the received Frame. |
| priority: | This parameter shall be used to indicate the priority of the received Frame; it shall be “system”, “urgent”, “normal” or “low”. |
| source_address: | This parameter shall be used to indicate the Source Address of the received Frame; it shall be the Individual Address of the device that has transmitted the Frame. |

### 2.2.2 Usage of ack_request

0 :don't care

1 :ack requested

### 2.2.3 Usage of priority

| pr | Medium Access Priority: | |
|---|---|---|
| 11b | low | Mandatory for long Frames, burst traffic... |
| 01b | normal | Default for short Frames |
| 10b | urgent | Reserved for urgent Frames |
| 00b | system | Reserved for *high priority, system configuration and management procedures* |

In [07] the usage conditions for these priorities are listed.

In a network, the Frame traffic using urgent priority shall not exceed 5 % of the total traffic (integration period: 1 minute maximum).

### 2.2.4 Usage of Frame Format

This Frame Format shall be used to select the Standard - or Extended Frame Format for Data Link Layer and shall include information for the used Extended Frame type.

- If the Frame format parameter is 0, then the Standard Frame Format shall be used.
- If this parameter is different from 0 then it shall be used as the Frame Format in the Extended Control field and the Extended Frame Format shall be used. The Extended Frame Format shall be selected by the Frame Type Parameter bit FTP = 1.

For the definition of the Extended Control field see the medium dependent layer description.

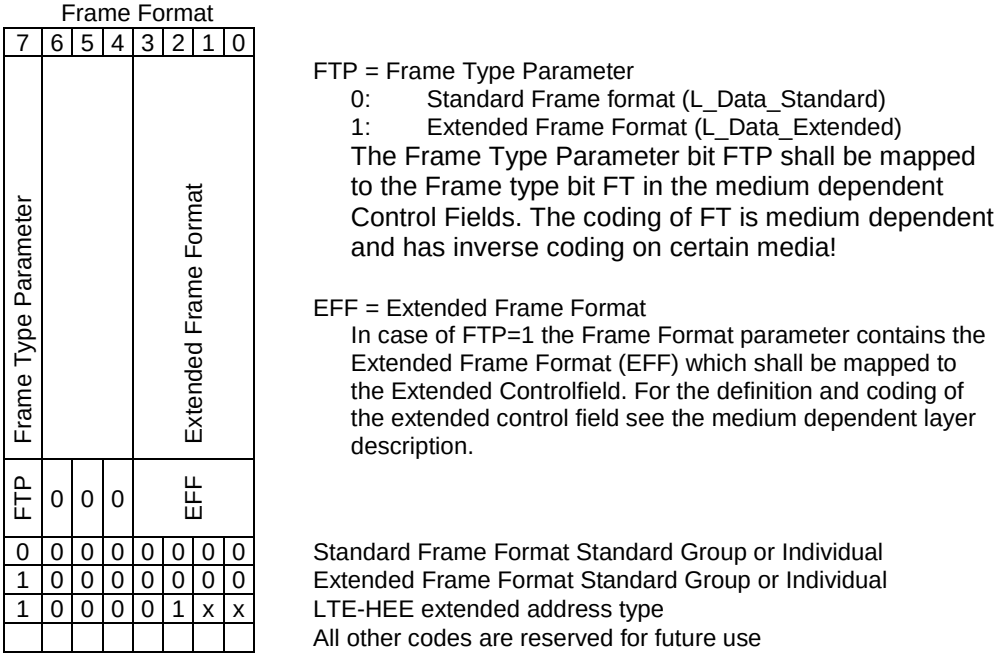



**Figure 5 - Frame format parameter**

The Extended Frame Format from the service parameter frame_format shall be placed in the Extended Control Field. The position of the extended Frame type is medium dependent.

Extended Frame Format (EFF)

| $b_3$ | $b_2$ | $b_1$ | $b_0$ | |
|---|---|---|---|---|
| $CtrlE_3$ | $CtrlE_2$ | $CtrlE_1$ | $CtrlE_0$ | Usage |
| 0 | 0 | 0 | 0 | Standard messages enabling long APDU >15 octets<br>Standard usage of DA for peer to peer or group communication |
| 0<br>0<br>0 | 0<br>0<br>0 | 0<br>1<br>1 | 1<br>0<br>1 | Reserved |
| 0 | 1 | X | X | LTE-HEE extended message format<br>$CtrlE_1$, $CtrlE_0$ shall contain the extension of DA group address |
| 1<br>1<br>1<br>1<br>1<br>1<br>1 | 0<br>0<br>0<br>0<br>1<br>1<br>1 | 0<br>0<br>1<br>1<br>0<br>0<br>1 | 0<br>1<br>0<br>1<br>0<br>1<br>0 | Reserved |
| 1 | 1 | 1 | 1 | Escape |

**Figure 6 - Coding of Extended Frame Format**

**Remark: Usage of Extended Frame Format**

The Extended Frame Format (L_Data_Extended) shall not be used instead of Standard Frame Format (L_Data_Standard) if the encoding capabilities of L_Data_Standard Frame are sufficient (e.g. for short Frames).

The decision whether to use Standard - or Extended Frame Format shall be made in the Application Layer and selected by the parameter frame_format in the T_Data_.... services. The remote Application Layer shall be tolerant towards usage of long Frames if short Frames would be sufficient

EXAMPLE A_PropertyValue_Read-PDU fits into the Standard (short) Frame Format. But if received using Extended (long) Frame Format it shall be accepted anyway by the remote Application Layer and the corresponding A_PropertyValue_Response-PDU shall be transported using the appropriate short or long format.

### 2.2.5 Usage of octet_count

This service parameter shall indicate the number of octets of the transported APDU.

The parameter octet_count shall be used on each medium to encode the LPDU LG field as follows.

- For Standard Frames, LG shall contains the number of octets in the APDU coded in 4 bit.
- For Extended Frames, LG shall contain the number of octets in the APDU coded in 8 bit except the value FFh. The value FFh (255) shall be used as an escape-code.

The escape-code (“ESC”) is reserved by KNX Association for future high speed media to enable larger lengths.

## 2.3 L_SystemBroadcast

The L_SystemBroadcast service shall be a Frame transfer service. It shall transmit a single Data Link Layer Service Data Unit (LSDU) to the Data Link Layer instances of all devices within the network. The Destination Address shall be the system broadcast address (Domain Address = 0000h and destination address = 0000h and address_type = “multicast”). The service may be acknowledged or not, depending on the transmission medium.

There shall be three service primitives:

1. L_SystemBroadcast.req shall be used to transmit a Frame;
2. L_SystemBroadcast.ind shall be used to receive a Frame;
3. L_SystemBroadcast.con shall be a local primitive generated by the local Layer-2 for its own client to indicate the success of the transmission.

If the local user of Layer-2 prepares an LSDU for the remote user it shall apply the L_System-Broadcast.req primitive to pass the LSDU to the local Layer-2. The local Layer-2 shall accept the service request and shall try to send the LSDU to the remote Layer-2 with the relevant Frame format.

The local Layer-2 shall pass an L_SystemBroadcast.con primitive to the local user that shall indicate either a correct or erroneous data transfer. Depending if an L2-acknowledgement is requested or not, this confirmation shall be related to the reception of the L2-acknowledgement, or only to the transmission of the Frame on the medium.

L_SystemBroadcast.req(ack_request, address_type, destination_address, frame_format, lsdu, octet_count, priority)

| | |
|---|---|
| ack_request: | This parameter shall be used to indicate whether a Layer-2 acknowledge is mandatory or optional. |
| address_type: | This parameter shall be set to “Group Address” |
| destination_address: | This parameter shall be used to indicate the Destination Address of the Frame to be transmitted; it shall be the system broadcast address 0000h |
| frame_format: | This parameter shall be used to indicate whether the Frame to be transmitted shall be a standard Frame or an extended Frame. |
| lsdu: | This parameter shall be used to contain the user data to be transferred by Layer-2. |
| octet_count: | This parameter shall be used to indicate the length information of the requested Frame. |
| priority: | This parameter shall be used to indicate the priority that shall be used to the transmit the requested Frame; it shall be “system”, “urgent”, “normal” or “low”. |

L_SystemBroadcast.con(address_type, destination_address, frame_format, octet_count, priority, source_address, lsdu, l_status)

| | | |
|---|---|---|
| address_type: | This parameter shall be set to “Group Address”. | |
| destination_address: | This parameter shall be used to indicate the Destination Address of the transmitted Frame; it shall be the system broadcast address 0000h | |
| frame_format: | This parameter shall be used to indicate whether the transmitted Frame is a standard Frame or an extended Frame. | |
| octet_count: | This parameter shall be used to indicate the length information of the transmitted Frame. | |
| priority: | This parameter shall be used to indicate the priority that is used to transmit the requested Frame; it shall be “system”, “urgent”, “normal” or “low”. | |
| source_address | This parameter shall be used to indicate the Source Address of the requested Frame; it shall be the Individual Address of the device that requests the service primitive. | |
| lsdu: | This parameter shall be the user data transferred by Layer-2 | |
| l_status: | ok: | this value of this parameter shall be used to indicate that the transmission of the LSDU has been successful |
| | not_ok: | this value of this parameter shall be used to indicate that the transmission of the LSDU did not succeed |

L_SystemBroadcast.ind(ack_request, address_type, destination_address, lsdu, octet_count, priority, source_address)

| | |
|---|---|
| ack_request: | This parameter shall be used to indicate whether a Layer-2 acknowledge is mandatory or optional. |
| address_type: | This parameter shall be set to “Group Address”. |
| destination_address: | This parameter shall be used to indicate the Destination Address of the received Frame; it shall be the system broadcast address 0000h |
| frame_format: | This parameter shall be used to indicate whether the received Frame is a standard Frame or an extended Frame. |
| lsdu: | This parameter shall be used to contain the user data that is received by Layer-2. |
| octet_count: | This parameter shall be used to indicate the length information of the received Frame. |
| priority: | This parameter shall be used to indicate the priority of the received Frame; it shall be “system”, “urgent”, “normal” or “low”. |
| source_address: | This parameter shall be used to indicate the Source Address of the received Frame; it shall be the Individual Address of the device that has transmitted the Frame. |

## 2.4 L_Poll_Data-service and protocol

The L_Poll_Data service shall be a confirmed multicast service. The local user of Layer-2 shall apply the L_Poll_Data.req primitive to request data from one or more remote users. The local Layer-2 shall accept the service request and shall try to send the L_Poll_Data-Frame to the remote Layer-2 with Frame format 3. The Destination Address shall always be a poll group address. The poll group address shall be a parameter of Layer-2.

L_Poll_Data request Frames that are not correctly received shall be discarded.

After receiving a correct L_Poll_Data request Frame with a poll_group_address equal to its own poll group addresses, the remote Layer-2 shall respond with a single Poll_Data character. The remote Layer-2 shall get the Poll_Data octet from its user with the L_Poll_Update.req primitive. The Poll_Data character shall be transmitted in the response slot associated with this device. The device's response slot shall be a defined time slot in which the Poll_Data slave device shall transmit the Poll_Data character. The duration of a response slot shall be an idle time of 5 times followed by a single UART character. If e.g. a device has the third response slot then the device shall wait for two Poll_Data characters transmitted by other devices, until the device transmits its Poll_Data character in the third response. The response slot number shall be a parameter of Layer-2.

A device shall not respond if its response slot number is larger than the number of expected poll data (no_of_expected_poll_data) in the L_Poll_Data request Frame.

The local Layer-2 shall expect a number of Poll_Data characters from the poll group. If an expected Poll_Data character has not started after five bit times the local Layer-2 shall send a FILL (FEh) after six bit times. The remote Layer-2 shall therefore still be able to count Poll_Data characters even if a member of the poll group does not respond.

The local Layer-2 shall pass a L_Poll_Data.con primitive to the local user that contains the received Poll_Data and FILL octets or an information that the service failed.

The L_Poll_Data Service can only be applied between devices on a single physical segment. The number of expected Poll_Data characters shall be limited to 16.

L_Poll_Data.req(destination, no_of_expected_poll_data)

| | |
|---|---|
| destination: | a poll group address |
| no_of_expected_poll_data: | number of expected poll data cycles |

L_Poll_Data.con(l_status, poll_data_sequence)

| | | |
|---|---|---|
| poll_data_sequence: | sequence of Poll_Data octets and FILL octets | |
| l_status: | ok: | valid poll_data_sequence |
| | not_ok: | invalid poll_data_sequence, i.e. collision occurred during transmission of a FILL, or at least one Poll_Data not correct |

L_Poll_Update.req(Poll_Data)

| | |
|---|---|
| Poll_Data: | The value of the Poll_Data octet to be transmitted in the L_Poll_Data_Response Frame. |

L_Poll_Update.con() Indicates that the L_Poll_Update.req has been accepted by the local Layer-2.

## 2.5 L_Busmon Service

The L_Busmon service shall be a local Data Link service available only in Busmonitor Mode. It shall consist of the L_Busmon.ind primitive that shall transfer every received Frame from the local Layer-2 to the local Layer-2 user.

L_Busmon.ind(l_status, time_stamp, LPDU)

| | |
|---|---|
| l_status: | information whether a Frame error, bit error or a parity error is detected in the received Frame. Additional information about the number of already received Frames may also be contained. |
| LPDU: | all octets of the received Frame |
| time_stamp: | timing information, when the start bit of the Frame was received |

## 2.6 L_Service_Information Service

The L_Service_Information service shall be a local Data Link service available in Data Link Normal Mode. It shall consist of the L_Service_Information.ind primitive.

| | |
|---|---|
| L_Service_Information.ind() | a Frame is received that contains the Individual Address of the local Layer-2 as Source Address. |

## 2.7 L_Management Service

This version of the KNX System Specification does not include a complete specification of layer management.

# 3 Data Link Layer Protocols

## 3.1 Protocol

The Data Link Layers shall offer a reliable Frame transfer service between devices on the same Subnetwork. This means that corrupted Frames shall be detected and retransmitted (i.e. repeated) for a sensible number of times, that only information of correctly received Frames shall be presented to the Data Link Layer user.

Devices may support Frame length adapted to their needs.

Corrupted Frames and Frames that exceed the reception capacity of a Device shall be discarded.

Some means are also provided on some media to prevent information duplication (i.e. that this information is not presented several times to the Data Link user).

Most of these functions are done in a medium dependent way. Please refer to the Data Link Layer protocols specified in a dedicated Chapter of these KNX Specifications for each KNX medium.

## 3.2 Recommendations for duplication prevention

Some means to prevent duplication are available on certain media. However there is always a remaining risk of duplication.

Informal rules to take this into account:

- Reduce noise at the medium as much as necessary to avoid corrupted acknowledgments.
- Be aware during internal or external user application programming of the possibility that in some cases a duplicated L_Data service may occur. This will be rare on media with collision avoidance but far more common on media without collision prevention- or detection or media that are more noise sensitive.
- Excessive use of the priority ‘urgent’ is not recommended on certain media priority.
- The medium, transmission method and transmitter and receiver technology shall provide basic noise immunity. (See the media descriptions in [02]).

# 4 Parameters of Layer-2

The following parameters shall influence the behaviour of Layer-2 and are required inside Layer-2 in order to operate correctly:

| | |
|---|---|
| Domain Address | Domain Address of the network the Device belongs to (on open media) |
| Individual Address | Individual Address of this device |
| Group Address Table | address table with the Group Address(es) of this device |
| Zone Address Table | address table with zone address information. |

NOTE The Group Address Table may include information for each Group Address to enable/disable sending of Layer-2 acknowledgements. Standard formats of these tables can be found in [05].

| | |
|---|---|
| nak_retry | defines the number of retries in case of a NAK response or a acknowledgement time-out |
| busy_retry | defines the number of retries in case of a BUSY/FULL response |
| poll Group Address | the poll Group Address of this Device |
| response slot number | the response slot number of this Device (for polling mode) |
| Data Link Layer mode | either the Normal mode or the Busmonitor Mode of the Data Link Layer. |

# 5 Busload

## 5.1 Definition of Busload

The busload is defined as the ratio of the time the bus is occupied by a signal transmitted by a single or several bus participants (see Figure 7) divided by the measured time $t_{measured}$. The busload is a percentage value:

$$Busload\ [\%] = \frac{\sum_{i=1}^{n} t_i}{t_{measured}} \cdot 100$$

$t_i$ bus occupancy time of a character i

$n$ number of characters during measurement time

$t_{measured}$ total time busload is measured

$t_{ref}$ $= \sum_{i=1}^{n} t_i$ total time of characters received during measurement time with values of Table 1.

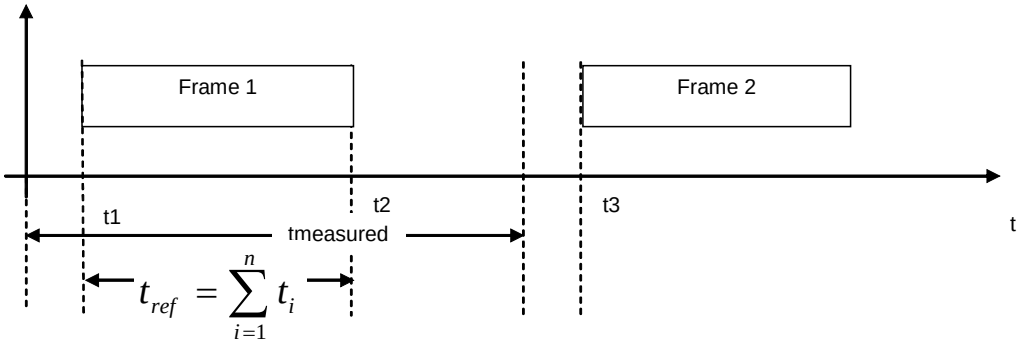



**Figure 7 - Example of times for measuring busload**

Consequently, 100 % busload defines a fully occupied bus with the nominal sender timings without any tolerance deviations (i.e. zero tolerance deviations).

## 5.2 Definition of Character classes

The character classes are defined in the KNX Specifications [02]. For each character class on the bus the occupation time $t_i$ is defined for each KNX medium (see Table 1).

**Table 1 - TP1 Character classes and the related bus occupation time**

| Character Class | TP1 character length | TP1 pause time before character starts | TP1 time bus is occupied ($t_i$) |
|---|---|---|---|
| START_OF_HIGH_PRIORITY_FRAME<br>START_OF_REPEATED_FRAME | 11 bit times | 50 bit times | 61 bit times = 6,354 ms |
| START_OF_LOW_PRIORITY_FRAME | 11 bit times | 53 bit times | 64 bit times = 6,667 ms |
| INNER_FRAME_CHAR | 11 bit times | 2 bit times | 13 bit times [1)] = 1,354 ms |
| ACK_CHAR | 11 bit times | 15 bit times | 26 bit times = 2,708 ms |
| POLL_DATA_CHAR | 11 bit times | 5 bit times | 16 bit times = 1,667 ms |
| POLL_DATA_FILL_CHAR | 11 bit times | 6 bit times | 17 bit times = 1,771 ms |

**EXAMPLE FOR TP1**

The sum of the bus occupancy times of the characters contained in a specific telegram is known. The value of the busload of a specific telegram (e.g. switching telegram with 1Bit payload) can be evaluated:

Switching telegram with 1Bit payload consists of 9 characters on bus.

**Table 2 - Example for TP1 of occupied bus by Switching telegram**

| Character class | number | Time |
|---|---|---|
| START_OF_LOW_PRIORITY_FRAME | 1 | 64 bit times |
| INNER_FRAME_CHAR | 8 | 8*(13 bit times) |
| ACK_CHAR | 1 | 26 bit times |
| | SUM ($t_i$) | 194 bit times (app. 20,21 ms) |

SUM ($t_i$) = (194 * 1/(9 600 bits per second) =194 * 104,166 µs = 20,208333 ms

Assuming the measuring time is 2 s for 50 switch telegrams the resulting busload would be:

$$Busload\ [\%] = \frac{50 \cdot 20{,}21ms}{2s} \cdot 100 = 50{,}52\ \%$$

[1)] 1 start bit + 1 stop bit + 8 data bits + 1 parity bit + 2 pause bits = 13 bits

## 5.3 Measuring busload

For busload measurement it is important for the client to consider bus synchronization. Most communication interfaces automatically synchronize to the start of a telegram between ongoing data traffic. The client has to take this under consideration by waiting for synchronization for the longest possible bus synchronization time of the interface to the bus, which is worst case the longest occupation time of a telegram on the medium (For TP1: Extended Frame with 1 start of Frame + 7 octets header + 254 inner Frame character payload + 1 inner Frame character checksum + 1 ACK character = 3 496 bit = 364,2 ms) or the first start Frame received. Client measurement timer will therefore start automatically after a fixed time period has elapsed (for TP1 = 364,2 ms) or the first telegram has been received. For the end time of the measurement this must also be considered, because a nearly complete received telegram from the interface will first be visible for the measurement client after it is completely received from bus. Therefore the end time for the measurement will be after a period of the longest occupation time of a telegram on the medium after the measurement end event. If both for start and stop measurement time is taken into account the measurement can be started and stopped during ongoing traffic.

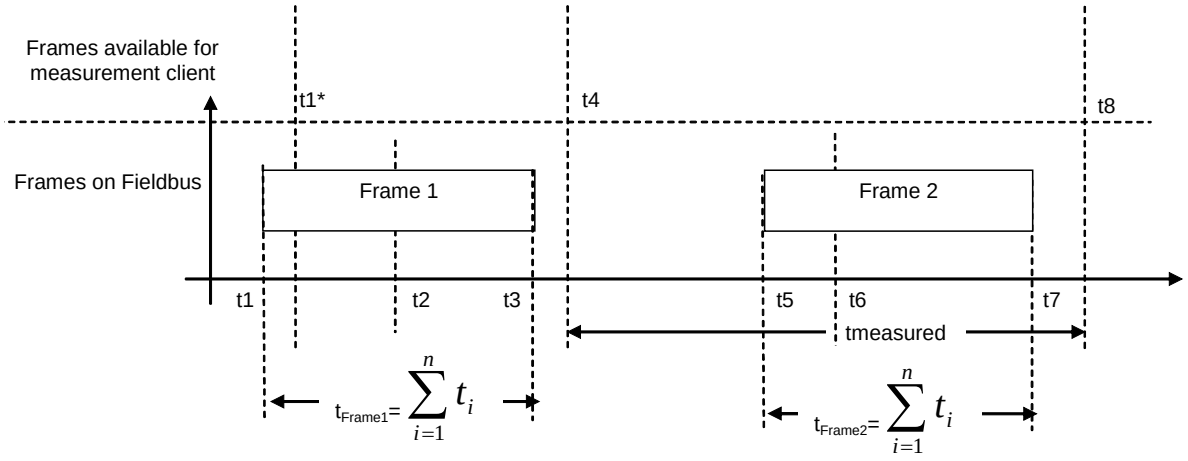



**Figure 8 - Example of measuring busload on TP1**

t1: Start time of Frame 1 on bus

t1*: Start time of Frame 1 plus evaluation and communication time between bus access device and measurement client (=t4 – $t_{Frame1}$)

t2: Start event for starting measurement of busload

t3: End time of Frame 1 on bus

t4: Frame 1 is available for measurement client, Measurement start time

t5: Start time of Frame 2 on bus

t6: Stop event for stopping measurement of busload

t7: End time of Frame 2 on bus

t8: Frame 2 is available for measurement client, Measurement stop time

**EXAMPLE**

Example values: $t_{Frame1} = t_{Frame2} = 20{,}21$ ms,
t1 = 0, t2 = 10 ms, t3 = 20,21 ms, t4 = 25 ms, t5 = 50 ms, t6 = 60 ms, t7 = 70,21 ms, t8 =75 ms
$t_{measured} = t8 - (t4 - t_{Frame1}) = 75\text{ ms} – (25\text{ ms} - 20{,}21\text{ ms}) = 70{,}21$ ms

Busload[%] = (2 * 20,21) / 70,21 * 100= 57,57 %

For an exact measurement the jitter of the time between the end of the Frames and the reception by the measurement client (t4-t3), (t8-t7) has to be known. This time is dependent on the internal structure of the interface and the communication path between device and measurement client. For the purpose of this document, the assumption that this jitter is constant is sufficient enough, especially for longer measurement times.

NOTE If the measurement time $t_{measure}$ for TP1 is bigger than 73 s (2*360 ms *100), even in worst case the error of the synchronisation to the TP1 field bus is smaller than 1 % and can then be ignored.

To build an average over some time intervals (e.g. for a visualisation) it is recommended to update this value every second. Same interval updates are recommended for a “history” datagram. For a shorter measurement time there must be considered the knowledge of the device internal delay times.

Due to the fact, that the measurement of busload is based on characters received on the medium, it is basically possible to include broken Frames and NACK characters into the calculation of busload. It is dependent of the used interface to the medium if broken Frames or NACK characters are available for the client and can therefore be included in the measurement.

## 5.4 Busload generation

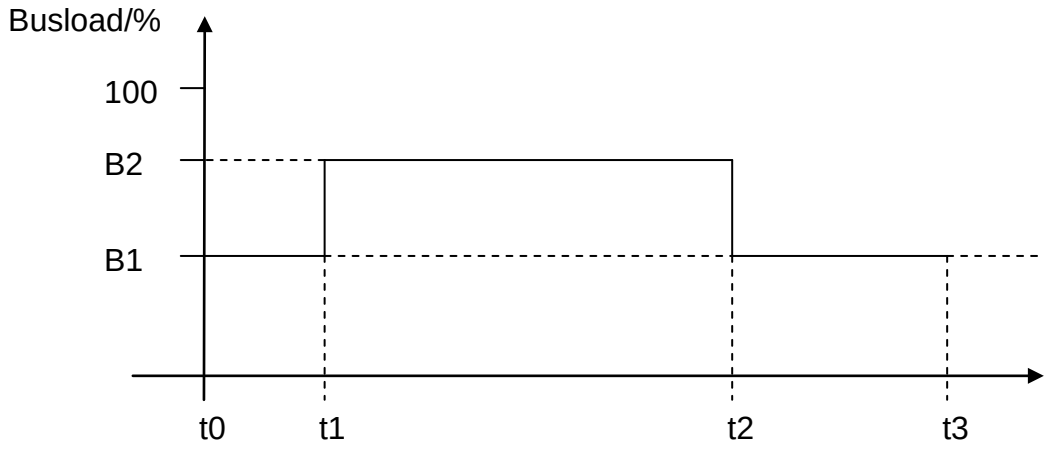
Busload may be generated in various ways; therefore a definition of busload types is required.

## 5.5 Busload types

### 5.5.1 Overview of the busload types

There are three different busload types defined:

1. uniform
2. burst
3. mixed




$$Busload(t) = B1 \quad \text{for } t_0 < t < t_1$$

$$Busload(t) = B2 \quad \text{for } t_1 < t < t_2$$

$$Busload(t) = B1 \quad \text{for } t_2 < t < t_3$$

$$\Delta t_1 := |t_1 - t_0|$$

$$\Delta t_2 := |t_2 - t_1|$$

$$\Delta t_3 := |t_3 - t_2|$$

$$\Delta t_{all} := |t_3 - t_0|$$

B1 stands for busload before t1 and after t2, B2 stands for busload for t1 < t <t2. The busload begins at t0 and ends at t3.

### 5.5.2 Uniform busload

A uniform busload has the same busload over a long time period ($\Delta t_{all} >> \Delta t_{Frame}$, B1 = B2 = const). The busload B1 is the same busload than B2. Measured busload will not differ if the measured time $t_{measured}$ will change as long as uniform busload lasts. Because the busload B1 = B2 there is no need to define the times t1 and t2.

$$\frac{t_{ref} \cdot telegrams_{counted}}{t_{measured}} = const = B2 = B1$$

**Table 3 - Example for uniform busload definitions**

| Uniform busload definitions (examples) | $\Delta t_{all}$ | B1 |
|---|---|---|
| Uniform_10min_100 | 10 min | 100 % |
| Uniform_1d_1 | 1 day | 1 % |

Be aware that uniform busload can only be measured in fixed time intervals ($t_{measured}$). It could be necessary to define this time and maybe times of repetition for measurement during tests.

### 5.5.3 Burst busload

Burst busload is defined as a busload during a limited amount of time (between times $t_1$ and $t_2$, B2 = const). There is no busload before and after that time (B1=0,):

**Table 4 - Example for burst busload definitions**

| Burst busload definitions (examples) | $\Delta t_1$ | $\Delta t_2$ | B2 | $\Delta t_3$ |
|---|---|---|---|---|
| Burst_10s_10min_20_1min | 10 s | 10 min | 20 % | 1 min |
| Burst_1min_2s_100_1min | 1 min | 2 s | 100 % | 1 min |

### 5.5.4 Mixed busload

Mixed busload will change at time $t_1$ from busload B1 to B2 and at $t_2$ from busload B2 to B1 (B1 = const1, B2 = const2). This is the most realistic case in the real world and is just a mixture from 5.5.2 and 5.5.3.

**Table 5 - Example for burst busload definitions**

| Mixed Busload definitions (examples) | $\Delta t_1$ | B1 | $\Delta t_2$ | B2 | $\Delta t_3$ |
|---|---|---|---|---|---|
| Mixed_5s_10_10s_100_1min | 5 s | 10 % | 10 s | 100 % | 1 min |
| Mixed_10min_30_20min_20_0 | 10 min | 30 % | 20 min | 20 % | 0 min |

## 5.6 Busload composition

A specific busload composition can be defined for a set of different Frames. To simulate real existing field bus environment it might make sense to define some typical busload compositions (e.g. several multicast Frames with one Broadcast Frame in between).

## 5.7 Requirements on devices

### 5.7.1 General

### 5.7.2 Introduction

#### 5.7.2.1 Use of correct frames

The focus of these definitions is to enable the testing of devices to correctly generated busload. The test suites therefore only generate correct frames to trigger ACK or BUSY L2-Acknowledge frames.

#### 5.7.2.2 Busload device class

Different device classes like system devices (e.g. a Coupler) need to fulfil higher requirements for busload evaluation, than normal end devices. The basic tests will be the same, but there are additional tests.

The following two devices classes are defined.

1. High power device (e.g. Coupler)
2. Normal End device

These classes are differentiated further in this paper.

### 5.7.3 Busload levels

#### 5.7.3.1 Definitions

The amount of busload can be classified in different busload levels.

| Level name | Expected behaviour | Busload value (approx.) |
|---|---|---|
| High | Chance to send in between (may send in between)<br>It is expected that the device can receive all telegrams, but may not be able to answer. | > 75 % |
| Medium | faire chance to send in between (must send in between) | 5 % to 75 % |
| Low | Any device shall be able to send, receive and respond with this busload<br>EXAMPLE 1 Within the time outs handled by ETS.<br>This shall be possible even under worst device conditions.<br>EXAMPLE 2 The device must send in-between even while writing EEPROM/ Flash write etc. | < 5 % |

#### 5.7.3.2 High busload

At high busload, devices must evaluate all received telegrams by checking for Data Link Layer conformity and generating L2-acknowledges.

There might be no possibility of a DUT to send telegrams during ongoing bus traffic because of

1. fairness of access (jitter in DUT), or
2. collision detection (if DUT is collision non dominant).

In case of TP1 the minimum pause time between two telegrams is 50 bit times + 3 bit times (low priority).

The DUT is only listener and could but might not send during ongoing bus traffic.

Expectation:

- DUT should generate ACK, NAK [2)] (if false characters/ telegrams are detected)
- DUT should store at least 1 read request and answer this at the latest when bus is free.

#### 5.7.3.3 Medium busload

At medium busload there is a fair chance to send telegrams during the ongoing bus traffic.

The time in between two telegrams must be bigger than 50 bit times + 3 bit low priority (if DUT sends low priority telegrams) + 3 bit jitter (fairness of access) + fair amounts of bits down to a total busload of 75 %.

The DUT must be able to send in between sending telegrams.

Expectation:

- DUT should generate ACK, NAK 2) (if false characters/ telegrams are detected)
- DUT should send during ongoing bus traffic (e.g. answer to read requests).

#### 5.7.3.4 Low busload

At low busload every device must answer, even if it has to write to EEPROM or do other time critical operations. This busload value is for clients like e.g. ETS of interest for programming or managing devices.

### 5.7.4 Busload evaluation Classes for devices

The busload evaluation Classes classify a busload from the perspective of a single device. There are the following evaluation classes of busload definition:

1. busload only receiving
   The device receives only a generated busload.
2. busload receiving and transmitting
   The device receives a generated busload and generates busload itself.
3. busload only transmitting
   The device only generates busload itself.

For each evaluation class (transmitting and/or receiving) and busload level (from low to high) and busload types (uniform, burst, mixed) exists expected device behaviour.

2) For further definitions NAK will not be mentioned. The busload topic is focused on evaluating and generating busload, not to define the reaction on faulty telegrams. It is expected, that on non faulty telegrams no NAK will be generated, independent of busload.

#### 5.7.4.1 Busload only receiving

A device shall be able to receive high busload. There shall be no different behaviour for different busload types.

**Table 6 - Expected behaviour for receiving at high busload**

| Busload High | Point to point | Multicast | Broadcast (not applicable [3]) |
|---|---|---|---|
| Device is not addressed | IGNORE [4] | ACK [5]/ IGNORE | Not applicable [6] |
| Device is addressed once during not addressed busload | IGNORE(only for not addressed)/ ACK (only for addressed frames) | ACK / IGNORE(only for not addressed | Not applicable |
| Device addressed [7] | ACK/ BUSY | ACK/ BUSY | ACK/ BUSY |

High busload with broadcast frames is not a realistic case. It would make sense to define some busload composition with one Broadcast frame in between and no BUSY allowed.

At medium busload no BUSY is allowed for multicast and point to point traffic.

**Table 7 - Expected behaviour for receiving at medium busload**

| Busload medium | Point to point | Multicast | Broadcast (not applicable) |
|---|---|---|---|
| Device is not addressed | IGNORE | ACK/ IGNORE | Not applicable |
| Device is addressed once during not addressed busload | IGNORE(only for not addressed)/ ACK (only for addressed frames) | ACK / IGNORE(only for not addressed | Not applicable |
| Device addressed | ACK | ACK | ACK/ BUSY |

At low busload no BUSY is allowed for Broadcast traffic.

**Table 8 - Expected behaviour for receiving at low busload**

| Busload low | Point to point | Multicast | Broadcast |
|---|---|---|---|
| Device is not addressed | IGNORE | ACK/ IGNORE | Not applicable |
| Device is addressed once during not addressed busload | IGNORE(only for not addressed)/ ACK (only for addressed frames) | ACK / IGNORE(only for not addressed | Not applicable |
| Device addressed | ACK | ACK | ACK |

3) High busload for broadcast frames is not realistic and should therefore not be tested.
4) IGNORE: DUT sends no L2-Acknowledge frame.
5) Devices are allowed to send Data Link Layer acknowledge frames on multicast traffic in a non-selective way (see [3]).
6) Every device has to evaluate Broadcast frames. There exists no broadcast traffic where the device is not addressed.
7) Only realistic for short bursts (time/ number of telegrams is not specified yet).

### 5.7.4.2 Busload receiving and transmitting

At high busload there might be no possibility of DUT to send telegrams during ongoing bus traffic because of:

1. fairness of access (jitter in DUT)
2. Collision Detection (If DUT is collision non dominant)

For burst or mixed busload types the device must be able to send if busload is medium or smaller.

**Table 9 – Expected behaviour for receiving and transmitting at high busload**

| Busload High | Point to point | Multicast | Broadcast (not applicable) |
|---|---|---|---|
| Device is not addressed | IGNORE | ACK/ IGNORE | Not applicable |
| Device is addressed once during not addressed busload | IGNORE(only for not addressed)/ ACK (only for addressed frames) and answer | ACK / IGNORE(only for not addressed)/ ACK (only for addressed frames) and answer | Not applicable |
| Device addressed (min 1 answer [8)], all other read request can be dropped) | ACK and answer/ BUSY (only after 1 read request) | ACK and answer/ BUSY (only after 1 read requests) | ACK and answer/ BUSY |

At medium busload level the device must be able to send during ongoing bus traffic.

**Table 10 – Expected behaviour for receiving and transmitting at medium busload**

| Busload medium | Point to point | Multicast | Broadcast (not applicable) |
|---|---|---|---|
| Device is not addressed | IGNORE | ACK/ IGNORE | Not applicable |
| Device is addressed once during not addressed busload | IGNORE(only for not addressed)/ ACK (only for addressed frames) and answer | ACK / IGNORE(only for not addressed)/ ACK (only for addressed frames) and answer | Not applicable |
| Device addressed | ACK and answer | ACK and answer | ACK and answer/ BUSY |

At low busload no BUSY is allowed for Broadcast traffic.

[8)] Either group value read or property value read or memory read

**Table 11 – Expected behaviour for receiving and transmitting at low busload**

| Busload low | Point to point | Multicast | Broadcast |
|---|---|---|---|
| Device is not addressed | IGNORE | ACK/ IGNORE | Not applicable |
| Device is addressed once during not addressed busload | IGNORE(only for not addressed)/ ACK (only for addressed frames) and answer | ACK / IGNORE(only for not addressed)/ ACK (only for addressed frames) and answer | Not applicable |
| Device addressed | ACK and answer | ACK and answer | ACK and answer |

### 5.7.4.3 Busload transmitting

On Data Link Layer level there are no requirements concerning the “busload only sending”. The Data Link Layer shall process the L_Data service requests from the Data Link Layer user according the Data Link Layer specifications of the relevant medium (i.e. respect the medium access control and the L_Data protocol).

NOTE For the application and the Application Interface Layer, the busload requirements as laid down in Chapter 3/7/1 “Interworking Model” clause 2.3 apply.

For multicast reading (value read/ value read responses) exist no sending restrictions because busload will sum up if there is more than one device asking for values or answering with values.

Clients have to respect restrictions in point to point communication, e.g. wait for an answer, or a specific time before sending next request to devices.

In future a send rate limitation for the medium to an absolute value, if devices exist with a lower computation power for the maximum KNX medium bandwidth (i.e. Ethernet 10/100/1000 MBit) could be made. This would lead to a change of the behaviour of Data Link Layer functionality.

# 6 Specific devices

## 6.1 The Layer-2 of the TP1 Bridge and the TP1 Repeater

A TP1 Bridge and a TP1 Repeater shall connect two segments of one single Subnetwork.

The notion of TP1 Bridge however provides only limited interest: it is not able to correctly handle the L2-acknowledgement, without detailed knowledge of the network.

Frames shall be forwarded and L2 acknowledgements shall be sent on each side of the TP1 Bridge and the TP1 Repeater depending on the Destination Address: the TP1 Bridge and a TP1 Repeater need to know which Device Addresses are used on each side of the TP1 Bridge or the TP1 Repeater [9)]. All other Layer-2 services shall be ignored.

A TP1 Bridge and a TP1 Repeater do not need an Individual Address. An Individual Address of a TP1 Bridge or a TP1 Repeater may be used to set manufacturer specific parameters in the TP1 Bridge or TP1 Repeater.

## 6.2 The Layer-2 of a Router

A Router shall a Layer-2 that shall respond to a correct L_Data Frame on either one of the 3 following conditions:

1. The value of the Destination Address is listed in the routing_table (see [03]).
2. The Destination Address is an Individual Address that indicates that the destination is on the other side of the Router.
3. The Destination Address is equal to the Individual Address of the Router.

In these cases, the L_Data.ind is indicated to the layer-3. All other Layer-2 services are ignored.

9) Sending back of L2-Acknowledgement cannot be done by the devices through the TP1 Bridge or TP1 Repeater: delays will be incompatible with the Layer-2 protocol.
Non-selective sending of L2-Acknowledgement and forwarding of Frames through the Bridge (i.e. as a repeater with delay) would inhibit the L2-acknowledge functionality. Mechanisms relying on this function to check addresses available would be fooled. So such Bridges are forbidden.

## 6.3 Coupler model overview (informative)

Table 12 gives the overview of these Coupler models.

**Table 12 – Overview of KNX Coupler models**

| Feature | Coupler type | | | |
|---|---|---|---|---|
| | **KNX Router** | **KNX TP1 Bridge** | **repeater** [a] | **KNX TP1 Repeater** |
| Positioned between | | | | |
| • Subnetworks | Yes | No | No | No |
| • physical segments of the same Subnetwork | No | Yes | Yes | Yes |
| Filtering | Yes. In function of DA. | No. | No. | No [b]. |
| hop count | | | | |
| • Frames with hop_count 0 | blocked | forwarded | forward | blocked |
| • hop count in forwarded Frames | Decremented (except hc 7) | unchanged | unchanged | Decremented (except hc 7) |

[a] This device does not exist on KNX.

[b] A device that can be configured as Router and as TP1 Repeater or TP1 Bridge holds a Filter Table. Filtering shall however be disabled to comply with this requirement.

# 7 Externally accessible Data Link Layer Interface

The Data Link Layer services can be made available to an external user application.

See [06].