Application Note 117/08 v02

Title: **KNX IP Communication Medium**

Status: Draft for Voting **Date:** 2010.02.02

Transitional period: Immediate effect after Final Voting.

Date: 2013.10.29

Subject: Specification of KNX IP (IP as a KNX medium)

Documents

Modified

[1] Chapter 3/5/3 “Configuration Procedures”
[2] Chapter 3/6/3 “External Message Interface”
[3] Chapter 3/8/3 “KNXnet/IP Device Management”
[4] Chapter 3/8/5 “KNXnet/IP Routing”
[5] Volume 6 “Profiles”, v1.0 AS

Referred

[6] Chapter 3/1/1 “Architecture”
[7] Chapter 3/3/2 “Data Link Layer general”
[8] Chapter 3/3/7 “Application Layer”
[9] Chapter 3/8/1 “KNXnet/IP Overview”
[10] Chapter 3/8/2 “KNXnet/IP Core”
[11] Chapter 3/8/4 “KNXnet/IP Tunnelling”
[12] AN089 v03 “Mask 0705h Specification”
[13] AN118 v01 “cEMI Transport Layer”

Document updates

Version	Date	Modifications
AN117 v01	2008.02.27	Preparation of the Draft Proposal.
AN117 v02	2010.02.02	Preparation of the Draft for Voting.

The content of this specification is still under discussion within KNX Association at the time of publication of the KNX Specifications. The requirements in this paper may have changed. For the current state and version, please contact the KNX Association's System Department.

Contents

1 General

1.1 Scope

🗎 *This clause is informative and will not be integrated in the KNX Specifications.*

This Application Note defines how the Internet Protocol (IP) is used as a KNX medium by devices that are connected to an IP network only. This KNX medium is named KNX IP. This document contains:

- KNX IP medium description including topology and flow control,
  [to be placed in→ KNX Handbook Chapter 3/2/6 “KNX IP Communication Medium”]
- additions to KNXnet/IP Device Management regarding cEMI services for transport of APDUs,
  [to be placed in→ KNX Handbook Chapter 3/8/3 “KNXnet/IP Management”]
- additions to KNXnet/IP Routing regarding flow control (ROUTING_BUSY),
  [to be placed in→ KNX Handbook Chapter 3/8/5 “KNXnet/IP Routing”]
- Configuration Procedures for KNX IP devices, and
  [to be placed in→ KNX Handbook Chapter 3/5/3 “Configuration Procedures”]
- test procedures for KNX IP devices.
  [to be placed in→ KNX Handbook Chapter 8/8/3 and Chapter 8/8/5]

1.2 Motivation

🗎 *This clause is informative and will not be integrated in the KNX Specifications.*

The KNXnet/IP specification defines message services on IP using cEMI encapsulated KNX telegrams exchanged between KNXnet/IP Routers sent to a specified IP multicast address. Currently access to this IP multicast group and to the messages sent in this group is limited to KNXnet/IP Routers as a backbone to connect KNX Subnetworks using other media (TP, RF, PL).

Currently, IP is not defined as a KNX medium. As a consequence among others, a device that connects to IP only cannot be configured by ETS.

Interpreting this IP backbone as part of the KNX topology with a given address space and the possibility for non-router-devices to be located there and have access to the IP multicast group will define the KNX/IP backbone as a new KNX medium for IP-only devices without a connection to one of the original KNX media.

1.3 Objective

🗎 *This clause is informative and will not be integrated in the KNX Specifications.*

The main objective of this specification is to supply information on the operation, configuration and implementation requirements for an IP based command-control network as part of the KNX communication network. The document does not cover the methods used to implement this specification.

The same backward compatibility paths that are necessary for the Interworking of the to be defined enhanced IP communication with the existing KNXnet/IP Routers apply for KNX IP devices based on this specification.

1.4 Definitions, acronyms and abbreviations

- KNX IP is the term used when the Internet Protocol (IP) is utilized as a KNX medium.
- KNX IP device describes a KNX device using the Internet Protocol (IP) as the only KNX medium [1)].
- A KNXnet/IP Server is a KNX device that has physical access to a KNX network and implements the KNXnet/IP Server protocol to communicate with KNXnet/IP Client or other KNXnet/IP Servers (in case of routing) on an IP network channel. A KNXnet/IP Server is by design always also a KNX node.
- A KNXnet/IP Client is an application that implements the KNXnet/IP Client protocol to get access to a KNX Subnetwork over an IP network channel.
- A KNXnet/IP Router is a special type of KNXnet/IP device that routes KNX protocol packets between KNX Subnetworks.

Tables listing implementation requirements use the following abbreviations:

Symbol	Description
M	Mandatory
C^n	Conditions are specified under note “n”
O	Optional
X	not allowed
n/a	not applicable
?	not yet defined (editorial indication)

1) A KNXnet/IP Tunnelling device connected to a KNX Subnetwork is e.g. a KNX TP1 device and SHALL not use a KNX mask 5xxxh.

2 KNX IP medium

🗎 *The KNX IP medium description including topology and flow control shall be placed in → KNX Handbook Chapter 3/2/6 "KNX IP Communication Medium".*

2.1 Terms and definitions

No new terms or definitions are introduced in this paper.

2.2 Medium definition

2.2.1 Physical specification of the Medium

As the KNX IP multicast access for message transportation at Physical Layer does not physically access the client IP medium but is transporting the KNX telegram data as Application Layer payload for the client IP stack, no physical specification for this KNX medium type is necessary.

2.2.2 Mechanical requirements

The equipment used to connect to the IP network shall conform to the specific requirements for the IP medium access (e.g. 802.3 for Ethernet LANs or 802.11 for Wireless LANs etc.) and all national regulations for electric installations.

2.2.3 Supplemental IP protocols

The KNX IP medium requires the implementation of a minimal set of supplemental IP protocols for interworking. Required IP protocols for the KNX IP medium access are ARP, BootP/DHCP, UDP, ICMP and IGMP.

Additional IP protocols may be required for the implementation of optional, device specific KNX IP services.

2.2.4 Physical topology

The physical topology of the KNX IP medium depends on the installation of the underlying IP network (LAN, WLAN or WAN), which must ensure that IP multicast telegrams are transported correctly across different IP topology segments.

2.2.5 Topology constraints

The proposal assumes that IP defines the transmission mechanism as well as the physical specifications (e.g. Ethernet). KNXnet/IP Routing is defined as the standard protocol for communication.

KNX IP Routers assigned to the same Project-Installation-ID SHALL use the same ROUTING_MULTICAST_ADDRESS as KNXnet/IP Routers with the same Project-Installation-ID.

This adds some restrictions to where inside the topology KNX IP Subnetworks can be used and where not: If a KNX IP device is assigned to a Subnetwork, then that Subnetwork and any Subnetwork higher in the logical topology shall contain KNX IP devices only. This leads to additional rules for the logical topology (compare first three rules with [4]).

Only one KNXnet/IP Routing multicast address is used by KNXnet/IP Routers [2)].

KNX IP devices have to implement Core Services, Device Management and KNXnet/IP Routing (see [9] clause 3.5 KNXnet/IP device classes) except those parts that are specific to the function of a KNXnet/IP Router.

The following six rules apply to KNX IP devices. The first three rules already apply to KNXnet/IP Routers.

Rule 1

In general a KNXnet/IP Router may be used as a Line Coupler or a Backbone Coupler. The Individual Address has the format x.y.0, with x = 1 to 15 and y = 0 to 15.

Rule 2:

If a KNXnet/IP Router is applied as a Backbone Coupler with the Individual Address x.0.0 then no other KNXnet/IP Router with the Line Coupler Individual Address x.y.0 (y = 1 to 15) shall be placed topologically „below“ this KNXnet/IP Router.

Rule 3:

If a KNXnet/IP Router is applied as a Line Coupler (e.g. with Individual Address 1.2.0) then no other KNXnet/IP Router shall be used with a superior Backbone Coupler Individual Address (e.g. 1.0.0) in this installation.

Rule 4:

If a KNX IP device is assigned to a Subnetwork as a simple device (e.g. with Individual Address 1.0.1) then that Subnetwork and any Subnetwork higher in the system structure shall contain KNX IP devices only.

2.3 Datagram service

2.3.1 Transmission method

This clause defines the transportation levels of the KNX IP communication system complying with Chapter 3/8/5 “KNXnet/IP Routing” ([4]).

IP Datagram

The KNX IP medium uses IP UDP datagram transfer on the fixed port 3671. The multicast address may vary for separate distinct installations on the same IP infrastructure. Usually the IP multicast address used for the KNXnet/IP Routing traffic is the same as the registered “System Setup” address 224.0.23.12.

Transmission Speed

The physical transmission speed on the KNX IP medium depends on the underlying IP network. This speed may vary across different segments of the IP topology.

Usually the effective transmission speed of the IP network is a lot faster than on KNX Twisted Pair or KNX Powerline Subnetworks. Therefore a limitation of the “penetration” speed (telegrams per second) is necessary to prevent buffer overflows in Media Couplers. This is achieved by the flow control implemented with the ROUTING_BUSY frame (see clause 4 of this document).

2) - KNX IP devices have one Group Address Table only (no changes to ETS).
- Flow control affects all devices and all communication between KNX IP devices.

For KNXnet/IP Routers ROUTING_BUSY is a means of preventing the loss of ROUTING_INDICATION datagrams due to an overflow of the buffer to the KNX Subnetwork.

KNX IP devices receive datagrams via the network transceiver ("Ethernet chip"), which forwards them to the microprocessor. Depending on the hardware and software design of the interface between the network transceiver and the microprocessor the effective data transmission rate between these two parts inside a KNX IP device may be lower than the actual transmission rate on the communication network. This internal receiving transmission rate limitation may cause the loss of datagrams between network transceiver and microprocessor.

Two different approaches were discussed to prevent the loss datagrams due to limited processing resources in a receiving KNX IP device.

a) Limit the total ROUTING_INDICATION transmission rate on the network to a defined value (e.g. one datagram per one millisecond for all devices).

b) Limit the ROUTING_INDICATION transmission rate to the network and at the same time require KNX IP devices and KNXnet/IP Routers to receive and process incoming ROUTING_INDICATION datagrams at a minimum transmission rate.

With case a) the transmission rate can be limited by tying the transmission rate of a specific KNX IP device to a maximum overall transmission rate. Any KNX IP device would then have to listen to the current ROUTING_INDICATION datagram traffic and hold back on transmitting its own datagrams if the limit is reached. Listening into the ROUTING_INDICATION datagram traffic requires the device to process incoming datagrams immediately when received by the hardware. This requires full control over hardware and software, which typically is not possible with operating systems like Linux or Windows. To allow for implementations on different platforms case a) was not considered.

Case b) is described in the following.

Consequently, limit the ROUTING_INDICATION transmission rate to the network and at the same time require KNX IP devices and KNXnet/IP Routers to receive and process incoming ROUTING_INDICATION datagrams at a minimum transmission rate.

Any KNX IP device or KNXnet/IP Router SHALL limit the transmission of KNX IP ROUTING_INDICATION datagrams to a maximum of 50 datagrams per second within one second.

For simplicity reasons a KNX IP device MAY choose to always pause 20 ms after it transmitted a ROUTING_INDICATION datagram. A KNX IP device or KNXnet/IP Router SHALL always pause its transmission on an assigned multicast address for at least 5 ms after it transmitted a ROUTING_INDICATION datagram.

To ensure a minimum system performance any KNX IP device or KNXnet/IP Router SHALL be capable of receiving and processing a minimum number of ROUTING_INDICATION datagrams per second on an assigned multicast address.

Ideally, any KNX IP device or KNXnet/IP Router SHOULD be capable of receiving and processing at least 12 750 ROUTING_INDICATION datagrams [3)] per second on an assigned multicast address. This number enables KNX IP devices or KNXnet/IP Routers to receive and process datagrams sent by up to 255 KNX IP devices or KNXnet/IP Routers transmitting at a rate of 50 ROUTING_INDICATION datagrams per second.

A KNXnet/IP Router or KNX IP device SHALL be able to receive and process up to the KNX Network - respectively Application Layer at least 1 000 ROUTING_INDICATION frames per second.

3) IP datagram length: 64 octets. Testing SHOULD be done with evenly distanced datagrams.

2.3.2 Frame encapsulation

The KNX IP medium uses cEMI encoded Data Link Layer Indication (L_Data.ind) frames for the UDP transport on the Multicast address (see above). No changes or additions to the telegram payload will be performed on KNXnet/IP Routing layer.

The KNX IP medium uses only unconfirmed services and because of the asynchronous nature of IP networks no flow control mechanisms are defined on the IP medium.

2.3.3 Physical Layer service definitions

Not applicable as the Internet Protocol frame is used as transport "medium" and thus the Physical Layer is defined by whatever medium is used as LAN or WAN.

2.4 Power feeding service

As the KNX IP medium is only a logical Physical Layer without capabilities to provide power feeding service for KNX devices connected to the KNX IP medium, power feeding for such devices is not in the scope of this document.

2.5 Data Link Layer type KNX IP

2.5.1 Frame format

2.5.1.1 Frame type summary

Each frame shall be a sequence of octets.

Three frame formats shall be provided:

1. a variable length frame format (ROUTING_INDICATION) (Figure 1), and
2. a buffer overflow warning indication (ROUTING_BUSY) (Figure 3)
3. a buffer overflow indication (ROUTING_LOST_MESSAGE) (Figure 4).

In the following representation of frames, the octet situated on the left hand side shall always be transmitted firstly.

2.5.1.2 Variable Length Frame Format

The structure of the variable length frame is shown in Figure 1.

```
KNXnet/IP header
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10              |   KNXNETIP_VERSION            |
|   (06h)                       |   (10h)                       |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ROUTING_INDICATION                                          |
|   (0530h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + sizeof(cEMI frame)                         |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   cEMI frame                                                  |
|   (variable length)                                           |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 1 - Complete frame encapsulation (datagram)

The encoding of the fields of the frame shall be done as specified in the clauses below.

KNXnet/IP header

For a detailed specification of the fields in the KNXnet/IP header see [10].

KNXnet/IP body

The KNXnet/IP body of the Routing Indication frame shall consist only of a cEMI encoded Data Link Layer Indication message.

Message Code	Additional Info Length	Additional Information	Control field 1	Control field 2	Src. High	Src. Low	Dest. High	Dest. Low	NPDU	
MC	**AddIL**	**…**	**Ctrl1**	**Ctrl2**	**SAH**	**SAL**	**DAH**	**DAL**	**L**	TPCI/APCI & data
29h	…	…	x0r0ppxx	…	…		…		…	…

Figure 2 - cEMI Data Link Layer Indication message format

The encoding of the fields of the cEMI L_Data.ind message in the context of KNXnet/IP shall be done as specified in the clauses below.

Additional Information

Medium specific "Additional Information" headers (see [2]) are allowed on the KNX IP medium if the originator of the frame is a KNXnet/IP Router or Media Coupler connected to a KNX medium that provides this medium specific additional information.

Control Fields

See [2].

Source Address

The Source Address shall be the Individual Address of the device that requests the transmission of the frame.

Destination Address

The Destination Address defines the device(s) that shall receive the frame. The Destination Address can either be an Individual Address or a Group Address, depending on the Address Type flag in the Control Fields (see above).

2.5.1.3 Buffer Overflow Warning Indication (ROUTING_BUSY)

The buffer overflow warning indication shall consist of a fixed length data field of six octets. It shall be used to indicate that the IP receive buffer has filled up to a point where the buffered incoming messages may take at least 100 ms to be sent to the KNX Subnetwork. The structure of the buffer overflow warning indication frame is shown in

Figure 3.

```
KNXnet/IP header
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10              |   KNXNETIP_VERSION            |
|   (06h)                       |   (10h)                       |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ROUTING_BUSY                                                |
|   (0532h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 4                                          |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

```
BusyInfo
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   DeviceState                 |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ROUTING_BUSY_WAIT_TIME (in milliseconds)                    |
|   (2 octets)                                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   ROUTING_BUSY_CONTROL_FIELD                                  |
|   (2 octets)                                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 3 – ROUTING_BUSY frame binary format

Details on handling of buffer overflow warnings and the Routing Busy Message frame can be found in [4]. The encoding of the fields of the frame shall be done as specified in the clauses below.

KNXnet/IP header

For a detailed specification of the fields in the KNXnet/IP header see [10].

KNXnet/IP body

For the encoding of the KNXnet/IP body of the Routing Lost Telegram frame see [4].

2.5.1.4 Buffer Overflow Indication (ROUTING_LOST_MESSAGE)

The buffer overflow indication shall consist of a fixed length data field of four octets. It shall be used to indicate that the IP receive buffer is full and incoming new messages got lost. The structure of the buffer overflow indication frame is shown in Figure 4.

```
KNXnet/IP header
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10              |   KNXNETIP_VERSION            |
|   (06h)                       |   (10h)                       |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ROUTING_LOST_MESSAGE                                        |
|   (0531h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 4                                          |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
LostMessageInfo
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   DeviceState                 |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   NumberOfLostMessages                                        |
|   (2 octets)                                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 4 – ROUTING_LOST_MESSAGE frame binary format

Details on handling of buffer overflows and the Routing Lost Message frame can be found in the [4]. The encoding of the fields of the frame shall be done as specified in the clauses below.

KNXnet/IP header

For a detailed description of the fields in the KNXnet/IP header see [10].

KNXnet/IP body

For the encoding of the KNXnet/IP body of the Routing Lost Telegram frame see [4].

2.5.2 Medium Access Control

KNX IP uses an asynchronous access to IP multicast medium. Medium access control on the IP network shall be handled by the underlying IP infrastructure.

Medium access is restricted by the ROUTING_BUSY scheme (see clause 4).

Further, a KNX IP device or KNXnet/IP Router shall limit the number of KNX IP ROUTING_INDICATION datagrams to a maximum of 50 datagrams per second.

2.5.3 Data Link Layer services and protocol

2.5.3.1 L_Data service

See [7] for the description of the service interface. The actions performed by Data Link Layer are listed in [6].

Telegrams shall be transported as cEMI (Data Link Layer, L_Data.ind = cEMI message code 29h) messages.

Frame Acceptance

A frame shall be considered valid if the number of characters received without error is consistent with the content of the "frame length" subfield.

Any reserved fields shall have the expected value.

Any invalid frame shall be ignored.

Frames exceeding the reception capabilities of the device shall be ignored.

Address Check

The frame is intended for the receiving device if the Destination Address (Individual Address or Group Address according to CTRL field) is recognised.

Data Link Layer acknowledgement sending and retransmissions

The KNX IP medium does not support Data Link Layer acknowledging of frames on the medium and hence does not define any retransmission rules on this layer.

Frame Fragmentation

Fragmentation on the physical medium is handled by the underlying IP infrastructure.

2.5.3.2 L_PollData service

This service is specific to KNX TP1 medium and cannot be supported on KNX IP.

NOTE 1 Periodic circular polling of devices may be achieved at application level using management supervision mechanisms (based on time slot assignment at applicative level).

2.5.3.3 L_Busmon service

See [7].

2.5.3.4 L_Service_Information service

See [7].

3 Additions to KNXnet/IP Device Management regarding cEMI services for transport of T_Data_Individual and T_Data_Connected messages

- *This clause explains the link from this document to additions to [2] (cEMI part) for transport of T_Data_Connected.req, T_Data_Connected.ind, T_Data_Individual.req, and T_Data_Individual.ind messages. [to be placed in → [2].*
- *This clause contains additions to KNXnet/IP Device Management regarding cEMI services for transport of T_Data_Connected.req, T_Data_Connected.ind, T_Data_Individual.req, and T_Data_Individual.ind messages.*
  [to be placed in → [3].

In a mixed environment with TP or PL Subnetworks, KNXnet/IP Routers and KNX IP devices in the KNXnet/IP backbone, these KNX IP devices can be accessed by the Management Client using serial or USB interfaces or even the KNXnet/IP Tunnelling adapter of one of the KNXnet/IP Routers. As the KNX IP devices are based on existing device models, these devices can be managed by the Management Client using the well-known Management- and Configuration Procedures for these device models.

KNX networks consisting solely of KNX IP devices have no bus interface for the Management Client. Task Force IP agreed that using the KNXnet/IP Routing communication for configuration purposes is unduly burdening the KNXnet/IP Routing communication. Furthermore with the KNXnet/IP Device Management protocol a point-to-point management mechanism for KNXnet/IP devices already exists for Property access. Therefore this KNXnet/IP Device Management protocol shall be extended to be used with the existing management and Configuration Procedures of the device models on which KNX IP devices are based.

KNXnet/IP uses cEMI to transfer KNX frames over IP. The KNXnet/IP Device Management protocol allows for a point-to-point connection between a Management Client over IP to a KNX IP device or KNXnet/IP Router. The cEMI Services for local device management (see [2] clause 4.1.7) allow for implementation independent management through cEMI Property services.

The cEMI Transport Layer defined in [13] allows for the download of an application program and parameters as well as the configuration of devices with a cEMI server without having to define new Management - and Configuration Procedures based on Property access only.

KNXnet/IP Device Management SHALL be used to establish the point-to-point communication connection transporting the APDUs required for configuration of the devices [4)].

KNXnet/IP devices implementing the cEMI services for local management SHALL announce Device Management service version 1.1 when responding to a SEARCH_REQUEST.

This approach covers configuration of devices using new mask versions for KNX IP as well as KNXnet/IP devices using existing mask versions.

4) To minimize the cost of protocol implementation the existing stack SHALL be used wherever possible.

4 Additions to KNXnet/IP Routing regarding flow control (ROUTING_BUSY)

🗎 *This clause contains additions to KNXnet/IP Routing regarding flow control (ROUTING_BUSY), [to be placed in→ KNX Handbook 3-8-5]*

Three new clauses shall be added to Chapter 3/8/5 "KNXnet/IP Routing":

- *clause 2.3.5, Flow control handling*
- *clause 5.4, ROUTING_BUSY*
- *clause 6.4, ROUTING_BUSY*

Chapter 3/8/3 "KNXnet/IP Device Management" shall be amended regarding

- *clause 2.4, KNXnet/IP Property definitions, Table 1*

new clause shall be added defining the a new Property added to Table 1

- *clause 2.4.27, PID_ROUTING_BUSY_WAIT_TIME*

Further, clause 2.5.1 shall be amended to include the new Property in Table 6, KNXnet/IP Parameter Object Properties.

These clauses are defined here.

4.1 Introduction

Flow control is introduced for KNXnet/IP Routers and KNX IP devices to avoid the loss of datagrams due to overflowing queues in KNXnet/IP Routers. The ROUTING_BUSY scheme SHALL be implemented by software packages too.

4.2 New clauses

4.2.1 Flow control handling

Depending on the configuration a KNXnet/IP Router could receive more datagrams from the LAN than it can send to the KNX Subnetwork. This could lead to an overflow of the LAN-to-KNX queue and subsequent loss of one or more KNXnet/IP telegrams because they could not be transferred from the network buffer to the queue to the underlying KNX subnetwork.

Flow control is introduced for KNXnet/IP Routers and KNX IP devices to avoid the loss of datagrams due to overflowing queues in KNXnet/IP Routers and KNX IP devices.

Limiting the data rate of sending devices is not a solution for flow control as it does not guarantee that the incoming queue on a specific device (e.g. a KNXnet/IP Router) does not overflow because it is receiving Datagrams to be sent onto the local Subnetwork from more than one sending device. The solution is for a receiving device to indicate to all other devices that its incoming queue is filling up and it may loose datagrams if they do not stop sending.

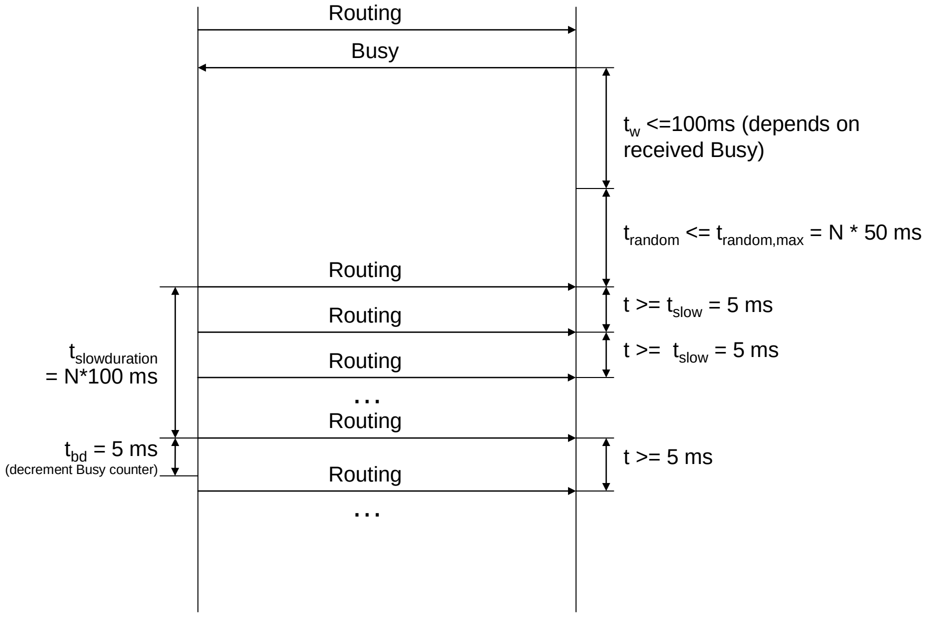


Figure 5 – Flow control with ROUTING_BUSY

Device sending ROUTING_BUSY

If the incoming queue (e.g. KNX IP to KNX TP1) of a KNXnet/IP Router or KNX IP device exceeds the number of datagrams that can be processed (e.g. sent to the local KNX Subnetwork by a KNXnet/IP Router) within a period of $T_{process}$ then this device SHALL send a ROUTING_BUSY frame with a wait time t_w. The default value for $T_{process}$ SHOULD be 100 ms and MAY be greater than 100 ms. t_w SHOULD resemble the time required to empty the incoming queue.

The value of t_w used by a device shall be stored in Property PID_ROUTING_BUSY_WAIT_-TIME. t_w SHALL be at least 20 ms and SHALL not exceed 100 ms.

The ROUTING_BUSY frame contains a routing_busy_control field. By default this routing_busy_control field SHALL be set to 0000h. This default value requires all KNX/IP devices and KNXnet/IP Routers to act upon receiving the ROUTING_BUSY frame.

The threshold for sending a ROUTING_BUSY frame with the individual address from the last ROUTING_INDICATION frame SHOULD be set at five messages in the incoming queue [5)].

The threshold for sending a ROUTING_BUSY frame to all KNX/IP devices and KNXnet/IP Routers SHOULD be set at ten messages in the incoming queue.

The incoming queue SHOULD be able to hold at least 30 messages.

5) The recommended values are based on a system simulation assuming that up to 255 devices send 50 ROUTING_INDICATION datagrams per second.

Device receiving ROUTING_BUSY

Any KNX IP device and KNXnet/IP Router SHALL stop sending ROUTING_INDICATION frames as soon as it receives a ROUTING_BUSY frame for the time t_w with a routing busy control field set to 0000h. This is where the ROUTING_BUSY frame acts as general flow control.

If another ROUTING_BUSY frame is received before the time t_w has elapsed the resulting time t_w SHALL be determined by the higher value of the remaining time of a previous ROUTING_BUSY and the value t_w received with this last ROUTING_BUSY.

A KNX IP device or KNXnet/IP Router MAY resume sending after the wait time t_w has expired and an additional random wait time t_{random} has passed.
For an individual device the total time from receiving the ROUTING_BUSY to resuming sending shall be

(1) $t_{w,\,total} = t_w + t_{random}$

(2) $t_{random} = [0\ldots1]_{random} * N * 50\text{ ms};$ $\{ 0 <= t_{random,} <= N * 50\text{ ms} \}$

The additional random wait time t_{random} shall be derived from a random real number in the range [0…1] multiplied by 50 ms to transmit and process a datagram times N.
N is defined as the number of ROUTING_BUSY frames received in a moving period. N shall be incremented by one with each ROUTING_BUSY frame received after 10 ms [6)] have passed since the last ROUTING_BUSY and decremented by one every t_{bd} = 5 ms after $t_{slowduration}$ has elapsed.

(3) $t_{slowduration} = N * 100\text{ ms}$

The ROUTING_BUSY frame allows a central supervising entity to log the routing traffic and determine potential problems with the system network design.

If the ROUTING_BUSY frame contains a routing busy control field value not equal to 0000h then any device that does not interpret this routing busy control field SHALL stop sending for the time t_w. In this case the rules for the general flow control also apply.

6) This value assumes that incoming ROUTING_BUSY datagrams are not due to network delays past 10 ms after the first ROUTING_BUSY is received. This avoids incrementing the counter N because more than one device exceeds the buffer trigger for sending ROUTING_BUSY at the same time.

4.2.2 ROUTING_BUSY

```
KNXnet/IP header
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10              |   KNXNETIP_VERSION            |
|   (06h)                       |   (10h)                       |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ROUTING_BUSY                                                |
|   (0532h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 6                                          |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
BusyInfo
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   DeviceState                 |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ROUTING_BUSY_WAIT_TIME (in milliseconds)                    |
|   (2 octets)                                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   ROUTING_BUSY_CONTROL_FIELD                                  |
|   (2 octets)                                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 6 – ROUTING_BUSY frame binary format

4.2.3 ROUTING_BUSY

```
          +-------------------------------+
  1       |              06h              |    header size
          +-------------------------------+
  2       |              10h              |    protocol version
          +-------------------------------+
  3       |              05h              | \
          +- - - - - - - - - - - - - - - -+  > service type identifier 0532h
  4       |              32h              | /
          +-------------------------------+
  5       |              00h              | \
          +- - - - - - - - - - - - - - - -+  > total length, 0Ch octets
  6       |              0Ch              | /
          +-------------------------------+
  7       |              04h              |    structure length
          +-------------------------------+
  8       |              00h              |    device state
          +-------------------------------+
  9       |              00h              | \
          +- - - - - - - - - - - - - - - -+  > number of milliseconds to wait,
 10       |              64h              | /   e.g. 100 ms
          +-------------------------------+
 11       |              00h              | \
          +- - - - - - - - - - - - - - - -+  > routing busy control value
 12       |              00h              | /  (default value: 0000h)
          +-------------------------------+
```

Figure 7 – ROUTING_BUSY frame binary format: example

4.2.4 PID_ROUTING_BUSY_WAIT_TIME (PID = 78)

- Property name: Routing Busy Wait Time
- Property Datatype: PDT_UNSIGNED_INT
- Datapoint Type: None.

The Property PID_ROUTING_BUSY_WAIT_TIME SHALL be accessible via the KNXnet/IP Parameter Object of a KNXnet/IP Router or KNX IP device.

This Property shall hold the value for the wait time t_w sent with a ROUTING_BUSY frame. The default value shall be 100 ms. The permissible value range is any integer value between 20 ms and 100 ms.

This Property is mandatory for devices implementing KNXnet/IP or KNX IP.

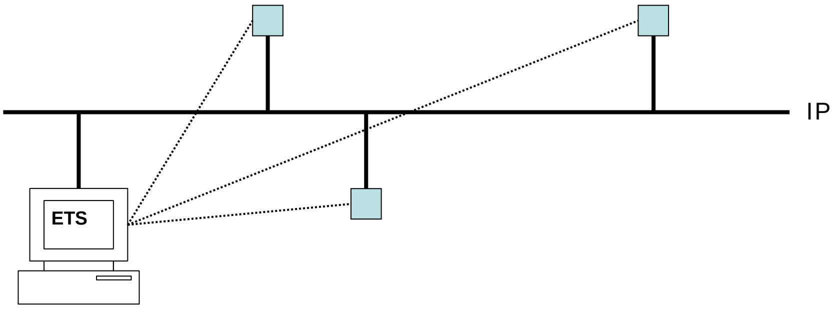
5 Configuration Procedures for KNX IP devices

This clause contains Configuration Procedures for KNX IP devices [to be placed in→ [1].

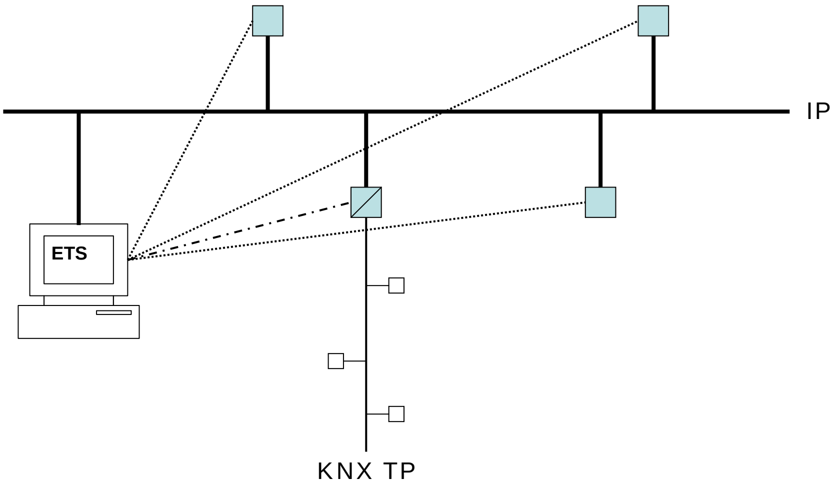
5.1 Assignment of Individual Address

There are three possible system constellations to be considered.

(A)



(B)



NOTE 2 User decision between direct connection via KNXnet/P Device Management or via KNXnet/IP Tunnelling. Default: direct connection via KNXnet/P Device Management.

(C)

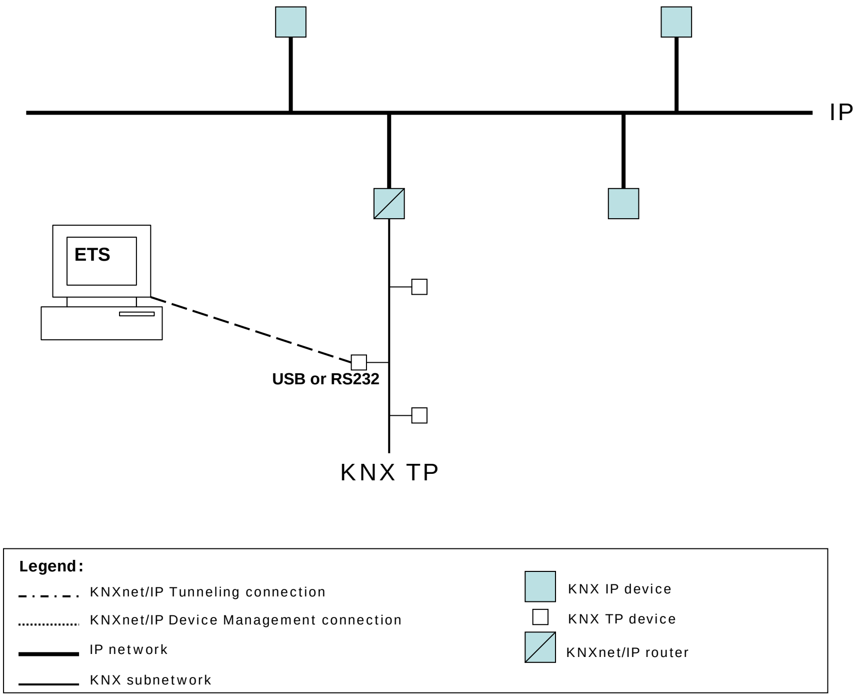


Figure 8 – System constellation A, B, and C

For the assignment of an Individual Address to a KNX IP device ETS SHALL implement KNXnet/IP Routing to ensure that the existing KNX Network Management Procedures can be used if other means of accessing the KNX network, e.g. USB, RS232, or KNXnet/IP Tunnelling server, are not available.

This approach covers

- that Programming Mode is active in only one device independent of the fact on which medium it is situated;
- that all system constellations (A), (B) or (C) are covered with the same procedure;
- that devices can be selected by activating their Programming Mode;
- that the Individual Address of a device can be changed (overwritten).

ETS SHALL offer an option for the assignment of an Individual Address using the MAC address of a device. ETS SHALL ask for the MAC address, then SEARCH for the corresponding device, and finally assign the Individual Address using KNXnet/IP Device Management. Alternatively, ETS may first SEARCH for KNX IP devices and then present a list with IP address, MAC address, and friendly name. If the list is empty the user may enter the IP address and a port number, which also covers NAT access situations.

5.2 Assignment of KNXnet/IP Routing Multicast Address

ETS SHALL set the KNXnet/IP Routing Multicast Address in all devices that are part of one installation.

This ensures that all devices of one installation use the same KNXnet/IP Routing Multicast Address. If an application entry provides a parameter for setting a KNXnet/IP Routing Multicast Address then this parameter value SHALL be overwritten by ETS with the correct value for the installation.

5.3 Download of Configuration:

ETS SHALL use KNXnet/IP Device Management with the cEMI Transport Layer services (cEMI T_Data_Connected.req, cEMI T_Data_Connected.ind, cEMI T_Data_Individual.req, and cEMI T_Data_Individual.ind) for point-to-point download of device configurations.

If a point-to-point connection is not possible ETS MAY configure a KNX IP device via the existing KNX Subnetworks, e.g. via KNX TP1 and access via USB or RS232 interface.

6 Test procedures for KNX IP devices

🗎 *This clause contains test procedures for KNX IP devices [to be placed in→ KNX Handbook 8-8-3 and 8-8-5]*

6.1 Overview

6.1.1 Tests

Test	KNXnet/IP Router	KNX IP device
6.2.1 Sending data to IP		
6.2.1.1 Wait time	M	M
6.2.1.2 Slow start	M	M
6.2.1.3 More ROUTING_BUSYs	M	M
6.2.1.4 ROUTING_BUSY received by a device that had sent ROUTING_BUSY	M	M
6.2.1.5 BUSY on KNX Subnetwork	M	n/a
6.2.1.6 No BUSY on KNX Subnetwork	M	n/a
6.2.1.7 Immediately stop sending on receiving ROUTING_BUSY	M	M
6.2.1.8 Limiting data rate to 50 frames per second	M	M
6.2.2 Receiving data from IP		
6.2.2.1 Test if different wait times	M	M
6.2.2.2 Don't stop sending after BUSY	M	M
6.2.2.3 Send packets not addressed	M	M
6.2.2.4 KNX Line down	M	n/a
6.2.2.5 Send large number of packets	M	n/a
6.2.3 KNX IP performance		
6.2.3.1 KNX IP device	n/a	M
6.2.3.2 KNX IP device (IGNORE)	n/a	M
6.2.3.3 KNX IP device (NETWORK)	n/a	M
6.2.3.4 KNXnet/IP Router	M	n/a
6.2.3.5 KNXnet/IP Router (IGNORE)	M	n/a
6.2.3.6 KNXnet/IP Router (NETWORK)	M	n/a

6.1.2 Test Set-up

6.1.2.1 Description

There are two different kinds of devices to test. Therefore two different Set-ups are needed.

This is required for both setups:

- KNXnet/IP Test device (KNXnet/IP Router, KNX IP device), BDUT
- PC with Testtool
- 10/100 MBit Switch/Managed Switch [7)]
- DHCP Server for automatic assignment of IP addresses
- Optional: PC with Ethereal/Wireshark
- Optional: A Routing v1.1 conforming Load Generator. It is only needed if the Testtool is not able to handle the Routing v1.1 protocol correctly. This could be the case because of timing issues.

Additionally, this equipment is needed for testing of KNXnet/IP Routers:

- KNX-Bus (Power supply)
- 2 (Busmon and Send/Receive) x Possibility to Access the KNX-Bus (via IP-Interface, USB-Interface, …)
- KNX load switch

7) Required for PC running a network traffic logger (e.g. Wireshark, Ethereal)

6.1.2.2 KNXnet/IP Router

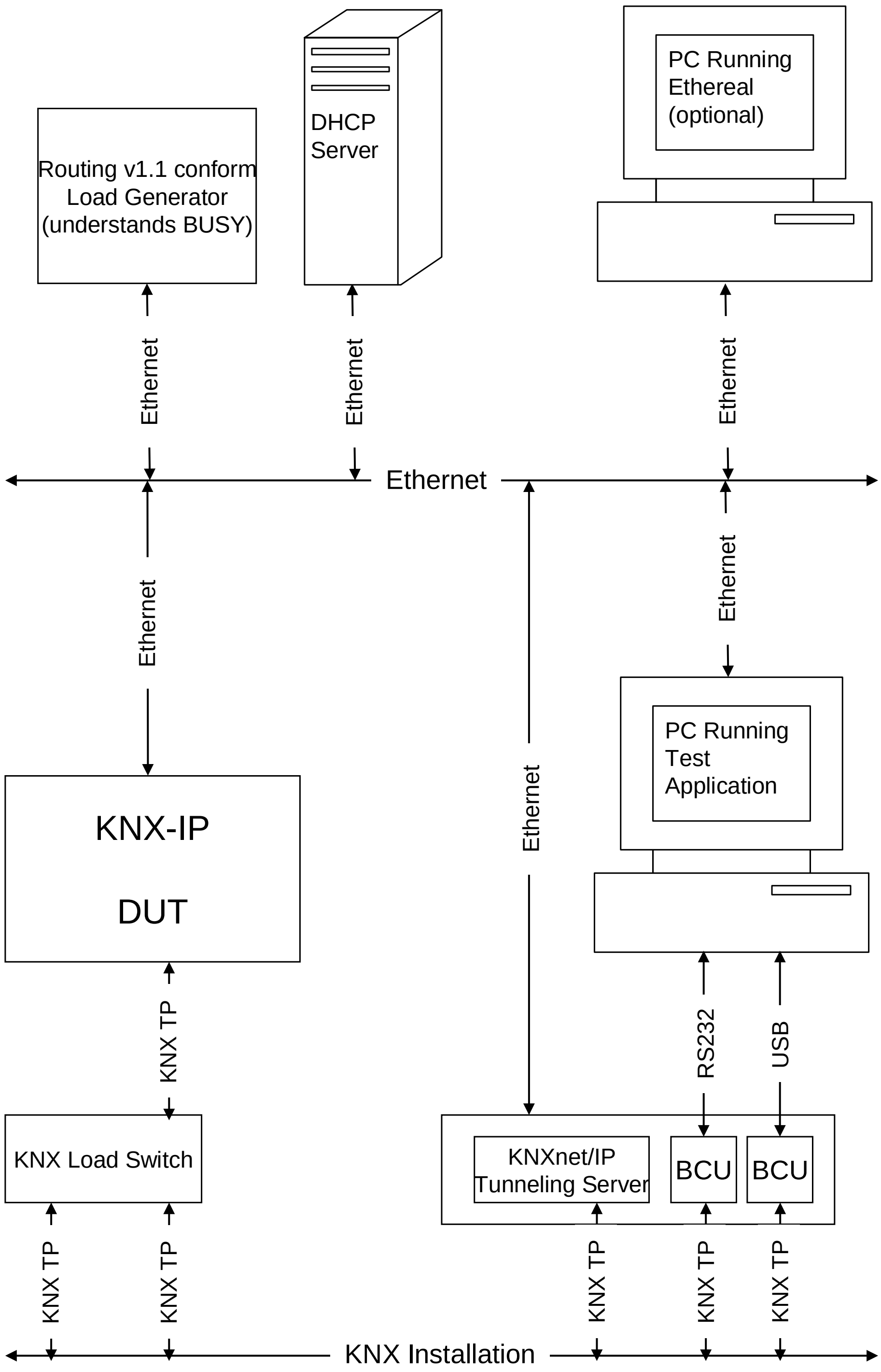


Figure 9 - Test setup for KNXnet/IP Routers

6.1.2.3 KNX IP device

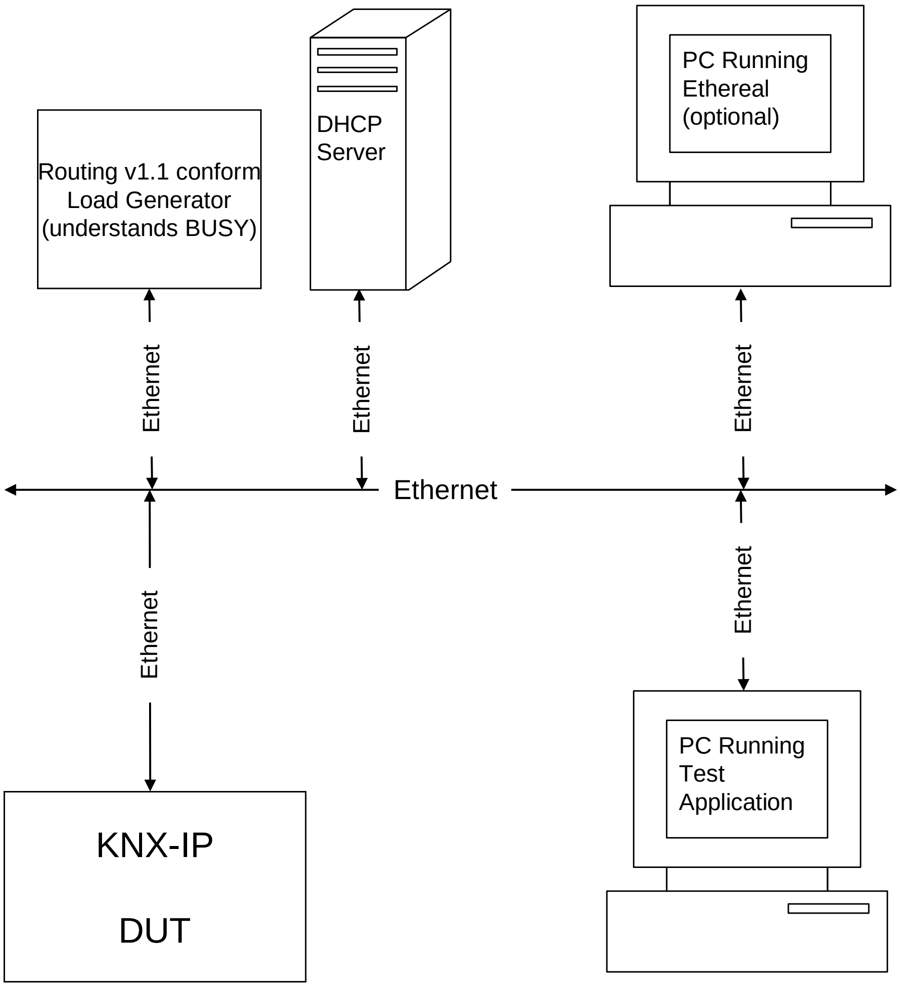


Figure 10 - Test setup for KNX IP devices

6.1.2.4 KNX IP load generator

The purpose of the KNX IP load generator is to send KNX IP multicast datagrams, which either have to be forwarded to a KNX Subnetwork (e.g. KNX TP1 line) or have to be processed by a KNX IP device.

If the KNX IP load generator is a separate hardware the load generation MAY be started manually.

The load generator SHALL send predefined frames.

Five frames are defined: FORWARD, PROCESS, BURDEN, IGNORE, and NETWORK.

The FORWARD [8)] frame SHALL be forwarded by a KNXnet/IP Router to the underlying KNX Subnetwork. The FORWARD frame is defined as

```
        +-------------------------------+
  1     |              00h              |
        +-------------------------------+
  2     |              00h              |
        +-------------------------------+
  3     |              01h              | \
        +- - - - - - - - - - - - - - - -+  \
  4     |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
  5     |              5Eh              |   |
        +- - - - - - - - - - - - - - - -+    > destination MAC
  6     |              00h              |   |  (multicast for address 224.0.23.12)
        +- - - - - - - - - - - - - - - -+   |
  7     |              17h              |   |
        +- - - - - - - - - - - - - - - -+  /
  8     |              0Dh              | /
        +-------------------------------+
  9     |              08h              | \
        +- - - - - - - - - - - - - - - -+  \
 10     |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
 11     |              00h              |   |
        +- - - - - - - - - - - - - - - -+    > source MAC 9
 12     |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
 13     |              00h              |   |
        +- - - - - - - - - - - - - - - -+  /
 14     |              77h              | /
        +-------------------------------+
 15     |              08h              | \
        +- - - - - - - - - - - - - - - -+  > Type IP
 16     |              00h              | /
        +-------------------------------+
 17     |              45h              | \
        +- - - - - - - - - - - - - - - -+  > Beginn of IP header
 18     |              00h              | /
        +-------------------------------+
 19     |              00h              | \
        +- - - - - - - - - - - - - - - -+  > Total length IP
 20     |              31h              | /
        +-------------------------------+
 21     |              00h              | \
        +- - - - - - - - - - - - - - - -+  > IP identification
 22     |              00h              | /
        +-------------------------------+
 23     |              00h              | \
        +- - - - - - - - - - - - - - - -+  \
 24     |              00h              |   |
        +-------------------------------+   |
 25     |              10h              |   |
        +- - - - - - - - - - - - - - - -+    > Header checksum 10
 26     |              11h              |   |
        +- - - - - - - - - - - - - - - -+   |
 27     |              B3h              |   |
        +- - - - - - - - - - - - - - - -+  /
 28     |              AFh              | /
        +-------------------------------+
 29     |              00h              | \
        +- - - - - - - - - - - - - - - -+  \
```

8) The FORWARD frame is associated with a response, which adds to the number of datagrams on the bus and/or network.

9) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

10) This Header Checksum must be changed with a different source MAC address.

```
30      |              00h              |    |
        +- - - - - - - - - - - - - - - -+    |
31      |              00h              |    |
        +- - - - - - - - - - - - - - - -+    |
32      |              00h              |    |
        +-------------------------------+    |
33      |              E0h              |    |
        +- - - - - - - - - - - - - - - -+    |
34      |              00h              |    |
        +- - - - - - - - - - - - - - - -+     > Begin of UDP
35      |              17h              |    |
        +- - - - - - - - - - - - - - - -+    |
36      |              0Dh              |    |
        +- - - - - - - - - - - - - - - -+    |
37      |              0Eh              |    |
        +- - - - - - - - - - - - - - - -+    |
38      |              57h              |    |
        +- - - - - - - - - - - - - - - -+    |
39      |              0Eh              |    |
        +- - - - - - - - - - - - - - - -+  /
40      |              57h              | /
        +-------------------------------+
41      |              00h              | \
        +- - - - - - - - - - - - - - - -+  > Length
42      |              1Dh              | /
        +-------------------------------+
43      |              79h              | \
        +- - - - - - - - - - - - - - - -+  >
44      |              61h              | /
        +-------------------------------+ - - - - KNXnet/IP header - - - -
45      |              06h              |    header size
        +-------------------------------+
46      |              10h              |    protocol version /
        +-------------------------------+
47      |              05h              | \
        +- - - - - - - - - - - - - - - -+  > service type identifier 0530h
48      |              30h              | /
        +-------------------------------+
49      |              00h              | \
        +- - - - - - - - - - - - - - - -+  > total length, 21 octets
50      |              15h              | /
        +-------------------------------+ - - - - cEMI frame - - - -
51      |              29h              |
        +-------------------------------+
52      |              00h              |
        +-------------------------------+
53      |              A0h              |
        +-------------------------------+
54      |              60h              |
        +-------------------------------+
55      |              AFh              | \
        +- - - - - - - - - - - - - - - -+  > Source address 10.15.254
56      |              FEh              | /
        +-------------------------------+
57      |              11h              | \
        +- - - - - - - - - - - - - - - -+  > Target address 1.1.207
58      |              CFh              | /
        +-------------------------------+
59      |              05h              | \
        +- - - - - - - - - - - - - - - -+  \
60      |              03h              |    |
        +- - - - - - - - - - - - - - - -+    |
61      |              D5h              |    |
        +- - - - - - - - - - - - - - - -+    |
62      |              00h              |     > Property Value Read
        +- - - - - - - - - - - - - - - -+    |  Object = 0
63      |              01h              |    |  Property = ID 1
        +- - - - - - - - - - - - - - - -+    |  number_of_elements = 1
64      |              10h              |    |  start_index = 1
        +- - - - - - - - - - - - - - - -+  /
65      |              01h              | /
        +-------------------------------+
```

The datagram carries this Frame:

Source Address 10.15.254

Target address 1.1.207

A_PropertyValue_Read-PDU

Object = 0

Property = ID 1

number_of_elements = 1

start_index = 1

The PROCESS frame SHALL be processed by a KNX IP device or KNXnet/IP Router. The PROCESS frame is defined as

```
        +--------------------------------+
 1      |              00h               |
        +--------------------------------+
 2      |              00h               |
        +--------------------------------+
 3      |              01h               | \
        +- - - - - - - - - - - - - - - --+  \
 4      |              00h               |   |
        +- - - - - - - - - - - - - - - --+   |
 5      |              5Eh               |   |
        +- - - - - - - - - - - - - - - --+    > destination MAC
 6      |              00h               |   |  (multicast for address 224.0.23.12)
        +- - - - - - - - - - - - - - - --+   |
 7      |              17h               |   |
        +- - - - - - - - - - - - - - - --+  /
 8      |              0Dh               | /
        +--------------------------------+
 9      |              08h               | \
        +- - - - - - - - - - - - - - - --+  \
10      |              00h               |   |
        +- - - - - - - - - - - - - - - --+   |
11      |              00h               |   |
        +- - - - - - - - - - - - - - - --+    > source MAC 11)
12      |              00h               |   |
        +- - - - - - - - - - - - - - - --+   |
13      |              00h               |   |
        +- - - - - - - - - - - - - - - --+  /
14      |              77h               | /
        +--------------------------------+
15      |              08h               | \
        +- - - - - - - - - - - - - - - --+  > Type IP
16      |              00h               | /
        +--------------------------------+
17      |              45h               | \
        +- - - - - - - - - - - - - - - --+  > Beginn of IP header
18      |              00h               | /
        +--------------------------------+
19      |              00h               | \
        +- - - - - - - - - - - - - - - --+  > Total length IP
20      |              31h               | /
        +--------------------------------+
21      |              00h               | \
        +- - - - - - - - - - - - - - - --+  > IP identification
22      |              00h               | /
        +--------------------------------+
23      |              00h               | \
        +- - - - - - - - - - - - - - - --+  \
24      |              00h               |   |
```

11) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

```
         +--------------------------------+    |
 25      |              10h               |    |
         +- - - - - - - - - - - - - - - - -+    > Header checksum 12)
 26      |              11h               |    |
         +- - - - - - - - - - - - - - - - -+    |
 27      |              B3h               |    |
         +- - - - - - - - - - - - - - - - -+   /
 28      |              AFh               |  /
         +--------------------------------+
 29      |              00h               |  \
         +- - - - - - - - - - - - - - - - -+   \
 30      |              00h               |    |
         +- - - - - - - - - - - - - - - - -+    |
 31      |              00h               |    |
         +- - - - - - - - - - - - - - - - -+    |
 32      |              00h               |    |
         +--------------------------------+    |
 33      |              E0h               |    |
         +- - - - - - - - - - - - - - - - -+    |
 34      |              00h               |    |
         +- - - - - - - - - - - - - - - - -+    > Begin of UDP
 35      |              17h               |    |
         +- - - - - - - - - - - - - - - - -+    |
 36      |              0Dh               |    |
         +- - - - - - - - - - - - - - - - -+    |
 37      |              0Eh               |    |
         +- - - - - - - - - - - - - - - - -+    |
 38      |              57h               |    |
         +- - - - - - - - - - - - - - - - -+    |
 39      |              0Eh               |    |
         +- - - - - - - - - - - - - - - - -+   /
 40      |              57h               |  /
         +--------------------------------+
 41      |              00h               |  \
         +- - - - - - - - - - - - - - - - -+   > Length
 42      |              1Dh               |  /
         +--------------------------------+
 43      |              79h               |  \
         +- - - - - - - - - - - - - - - - -+   >
 44      |              61h               |  /
         +--------------------------------+ - - - - KNXnet/IP header - - - -
 45      |              06h               |    header size
         +--------------------------------+
 46      |              10h               |    protocol version /
         +--------------------------------+
 47      |              05h               |  \
         +- - - - - - - - - - - - - - - - -+   > service type identifier 0530h
 48      |              30h               |  /
         +--------------------------------+
 49      |              00h               |  \
         +- - - - - - - - - - - - - - - - -+   > total length, 21 octets
 50      |              15h               |  /
         +--------------------------------+ - - - - cEMI frame - - - -
 51      |              29h               |
         +--------------------------------+
 52      |              00h               |
         +--------------------------------+
 53      |              A0h               |
         +--------------------------------+
 54      |              60h               |
         +--------------------------------+
 55      |              AFh               |  \
         +- - - - - - - - - - - - - - - - -+   > Source address 10.15.254
 56      |              FEh               |  /
         +--------------------------------+
 57      |              11h               |  \
         +- - - - - - - - - - - - - - - - -+   > Target address 1.1.0
 58      |              00h               |  /
         +--------------------------------+
```

12) This Header Checksum must be changed with a different source MAC address.

```
 59     |              05h              | \
        +- - - - - - - - - - - - - - - -+  \
 60     |              03h              |   |
        +- - - - - - - - - - - - - - - -+   |
 61     |              D5h              |   |
        +- - - - - - - - - - - - - - - -+   |
 62     |              00h              |    > Property Value Read
        +- - - - - - - - - - - - - - - -+   |  Object = 0
 63     |              01h              |   |  Property = ID 1
        +- - - - - - - - - - - - - - - -+   |  number_of_elements = 1
 64     |              10h              |   |  start_index = 1
        +- - - - - - - - - - - - - - - -+  /
 65     |              01h              | /
        +-------------------------------+
```

The datagram carries this Frame.

Source Address 10.15.254

Target address 1.1.0

A_PropertyValue_Read-PDU

Object = 0

Property = ID 1

number_of_elements = 1

start_index = 1

The BURDEN frame SHALL be received and processed by a KNX IP device or KNXnet/IP Router to the point that the device determines that this frame is not for itself or an underlying KNX Subnetwork. The BURDEN frame is defined as

```
        +-------------------------------+
  1     |              00h              |
        +-------------------------------+
  2     |              00h              |
        +-------------------------------+
  3     |              01h              | \
        +- - - - - - - - - - - - - - - -+  \
  4     |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
  5     |              5Eh              |   |
        +- - - - - - - - - - - - - - - -+    > destination MAC
  6     |              00h              |   |  (multicast for address 224.0.23.12)
        +- - - - - - - - - - - - - - - -+   |
  7     |              17h              |   |
        +- - - - - - - - - - - - - - - -+  /
  8     |              0Dh              | /
        +-------------------------------+
  9     |              08h              | \
        +- - - - - - - - - - - - - - - -+  \
 10     |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
 11     |              00h              |   |
        +- - - - - - - - - - - - - - - -+    > source MAC 13)
 12     |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
 13     |              00h              |   |
        +- - - - - - - - - - - - - - - -+  /
 14     |              77h              | /
        +-------------------------------+
 15     |              08h              | \
        +- - - - - - - - - - - - - - - -+  > Type IP
 16     |              00h              | /
```

13) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

```
           +--------------------------------+
   17      |              45h               | \
           +- - - - - - - - - - - - - - - - -+  > Beginn of IP header
   18      |              00h               | /
           +--------------------------------+
   19      |              00h               | \
           +- - - - - - - - - - - - - - - - -+  > Total length IP
   20      |              31h               | /
           +--------------------------------+
   21      |              00h               | \
           +- - - - - - - - - - - - - - - - -+  > IP identification
   22      |              00h               | /
           +--------------------------------+
   23      |              00h               | \
           +- - - - - - - - - - - - - - - - -+  \
   24      |              00h               |    |
           +--------------------------------+    |
   25      |              10h               |    |
           +- - - - - - - - - - - - - - - - -+     > Header checksum 14)
   26      |              11h               |    |
           +- - - - - - - - - - - - - - - - -+    |
   27      |              B3h               |    |
           +- - - - - - - - - - - - - - - - -+  /
   28      |              AFh               | /
           +--------------------------------+
   29      |              00h               | \
           +- - - - - - - - - - - - - - - - -+  \
   30      |              00h               |    |
           +- - - - - - - - - - - - - - - - -+    |
   31      |              00h               |    |
           +- - - - - - - - - - - - - - - - -+    |
   32      |              00h               |    |
           +--------------------------------+    |
   33      |              E0h               |    |
           +- - - - - - - - - - - - - - - - -+    |
   34      |              00h               |    |
           +- - - - - - - - - - - - - - - - -+     > Begin of UDP
   35      |              17h               |    |
           +- - - - - - - - - - - - - - - - -+    |
   36      |              0Dh               |    |
           +- - - - - - - - - - - - - - - - -+    |
   37      |              0Eh               |    |
           +- - - - - - - - - - - - - - - - -+    |
   38      |              57h               |    |
           +- - - - - - - - - - - - - - - - -+    |
   39      |              0Eh               |    |
           +- - - - - - - - - - - - - - - - -+  /
   40      |              57h               | /
           +--------------------------------+
   41      |              00h               | \
           +- - - - - - - - - - - - - - - - -+  > Length
   42      |              1Dh               | /
           +--------------------------------+
   43      |              79h               | \
           +- - - - - - - - - - - - - - - - -+  >
   44      |              61h               | /
           +--------------------------------+ - - - - KNXnet/IP header - - - -
   45      |              06h               |   header size
           +--------------------------------+
   46      |              10h               |   protocol version /
           +--------------------------------+
   47      |              05h               | \
           +- - - - - - - - - - - - - - - - -+  > service type identifier 0530h
   48      |              30h               | /
           +--------------------------------+
   49      |              00h               | \
           +- - - - - - - - - - - - - - - - -+  > total length, 21 octets
   50      |              15h               | /
           +--------------------------------+ - - - - cEMI frame - - - -
```

---

14) This Header Checksum must be changed with a different source MAC address.

```
51      |              29h              |
        +-------------------------------+
52      |              00h              |
        +-------------------------------+
53      |              A0h              |
        +-------------------------------+
54      |              60h              |
        +-------------------------------+
55      |              AFh              | \
        +- - - - - - - - - - - - - - - -+  > Source address 10.15.254
56      |              FEh              | /
        +-------------------------------+
57      |              12h              | \
        +- - - - - - - - - - - - - - - -+  > Target address 1.2.207
58      |              CFh              | /
        +-------------------------------+
59      |              05h              | \
        +- - - - - - - - - - - - - - - -+  \
60      |              03h              |   |
        +- - - - - - - - - - - - - - - -+   |
61      |              D5h              |   |
        +- - - - - - - - - - - - - - - -+   |
62      |              00h              |    > Property Value Read
        +- - - - - - - - - - - - - - - -+   |  Object = 0
63      |              01h              |   |  Property = ID 1
        +- - - - - - - - - - - - - - - -+   |  number_of_elements = 1
64      |              10h              |   |  start_index = 1
        +- - - - - - - - - - - - - - - -+  /
65      |              01h              | /
        +-------------------------------+
```

The datagram carries this Frame.

Source Address 10.15.254

Target address 1.2.207

A_PropertyValue_Read-PDU

Object = 0

Property = ID 1

number_of_elements = 1

start_index = 1

The IGNORE frame SHALL be received and processed by a KNX IP device or KNXnet/IP Router to the point in the IP stack that the device determines that this frame can be discarded without an answer because its multicast address (e.g. 239.0.23.13) is different from the expected multicast address (224.0.23.12). The IGNORE frame is defined as

```
        +-------------------------------+
 1      |              00h              |
        +-------------------------------+
 2      |              00h              |
        +-------------------------------+
 3      |              01h              | \
        +- - - - - - - - - - - - - - - -+  \
 4      |              00h              |   |
        +- - - - - - - - - - - - - - - -+   |
 5      |              5Eh              |   |
        +- - - - - - - - - - - - - - - -+    > destination MAC
 6      |              00h              |   |  (multicast for address 239.0.23.13)
        +- - - - - - - - - - - - - - - -+   |
 7      |              17h              |   |
        +- - - - - - - - - - - - - - - -+  /
 8      |              0Eh              | /
        +-------------------------------+
 9      |              08h              | \
        +- - - - - - - - - - - - - - - -+  \
10      |              00h              |   |
```

```
        +- - - - - - - - - - - - - - - - -+    |
 11     |              00h                |    |
        +- - - - - - - - - - - - - - - - -+     > source MAC 15)
 12     |              00h                |    |
        +- - - - - - - - - - - - - - - - -+    |
 13     |              00h                |    |
        +- - - - - - - - - - - - - - - - -+   /
 14     |              77h                | /
        +---------------------------------+
 15     |              08h                | \
        +- - - - - - - - - - - - - - - - -+  > Type IP
 16     |              00h                | /
        +---------------------------------+
 17     |              45h                | \
        +- - - - - - - - - - - - - - - - -+  > Beginn of IP header
 18     |              00h                | /
        +---------------------------------+
 19     |              00h                | \
        +- - - - - - - - - - - - - - - - -+  > Total length IP
 20     |              31h                | /
        +---------------------------------+
 21     |              00h                | \
        +- - - - - - - - - - - - - - - - -+  > IP identification
 22     |              00h                | /
        +---------------------------------+
 23     |              00h                | \
        +- - - - - - - - - - - - - - - - -+   \
 24     |              00h                |    |
        +---------------------------------+    |
 25     |              A4h                |    |
        +- - - - - - - - - - - - - - - - -+     > Header checksum 16)
 26     |              AEh                |    |
        +- - - - - - - - - - - - - - - - -+    |
 27     |              B3h                |    |
        +- - - - - - - - - - - - - - - - -+   /
 28     |              AFh                | /
        +---------------------------------+
 29     |              00h                | \
        +- - - - - - - - - - - - - - - - -+   \
 30     |              00h                |    |
        +- - - - - - - - - - - - - - - - -+    |
 31     |              00h                |    |
        +- - - - - - - - - - - - - - - - -+    |
 32     |              00h                |    |
        +---------------------------------+    |
 33     |              E0h                |    |
        +- - - - - - - - - - - - - - - - -+    |
 34     |              00h                |    |
        +- - - - - - - - - - - - - - - - -+     > Begin of UDP
 35     |              17h                |    |
        +- - - - - - - - - - - - - - - - -+    |
 36     |              0Dh                |    |
        +- - - - - - - - - - - - - - - - -+    |
 37     |              0Eh                |    |
        +- - - - - - - - - - - - - - - - -+    |
 38     |              57h                |    |
        +- - - - - - - - - - - - - - - - -+    |
 39     |              0Eh                |    |
        +- - - - - - - - - - - - - - - - -+   /
 40     |              57h                | /
        +---------------------------------+
 41     |              00h                | \
        +- - - - - - - - - - - - - - - - -+  > Length
 42     |              1Dh                | /
        +---------------------------------+
 43     |              78h                | \
```

15) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

16) This Header Checksum must be changed with a different source MAC address.

```
          +- - - - - - - - - - - - - - - - -+  >
  44      |             61h                 | /
          +---------------------------------+ - - - - KNXnet/IP header - - - -
  45      |             06h                 |    header size
          +---------------------------------+
  46      |             10h                 |    protocol version /
          +---------------------------------+
  47      |             05h                 | \
          +- - - - - - - - - - - - - - - - -+  > service type identifier 0530h
  48      |             30h                 | /
          +---------------------------------+
  49      |             00h                 | \
          +- - - - - - - - - - - - - - - - -+  > total length, 21 octets
  50      |             15h                 | /
          +---------------------------------+ - - - - cEMI frame - - - -
  51      |             29h                 |
          +---------------------------------+
  52      |             00h                 |
          +---------------------------------+
  53      |             A0h                 |
          +---------------------------------+
  54      |             60h                 |
          +---------------------------------+
  55      |             AFh                 | \
          +- - - - - - - - - - - - - - - - -+  > Source address 10.15.254
  56      |             FEh                 | /
          +---------------------------------+
  57      |             11h                 | \
          +- - - - - - - - - - - - - - - - -+  > Target address 1.1.207
  58      |             CFh                 | /
          +---------------------------------+
  59      |             05h                 | \
          +- - - - - - - - - - - - - - - - -+  \
  60      |             03h                 |   |
          +- - - - - - - - - - - - - - - - -+   |
  61      |             D5h                 |   |
          +- - - - - - - - - - - - - - - - -+   |
  62      |             00h                 |    > Property Value Read
          +- - - - - - - - - - - - - - - - -+   |  Object = 0
  63      |             01h                 |   |  Property = ID 1
          +- - - - - - - - - - - - - - - - -+   |  number_of_elements = 1
  64      |             10h                 |   |  start_index = 1
          +- - - - - - - - - - - - - - - - -+  /
  65      |             01h                 | /
          +---------------------------------+
```

The NETWORK test datagram is an ARP request that is not answerd by the DUT.

The NETWORK frame SHALL be received and processed by a KNX IP device or KNXnet/IP Router to the point that the device determines that this frame is not addressed to it. The NETWORK frame is defined as

```
          +---------------------------------+
   1      |             00h                 |
          +---------------------------------+
   2      |             00h                 |
          +---------------------------------+
   3      |             FFh                 | \
          +- - - - - - - - - - - - - - - - -+  \
   4      |             FFh                 |   |
          +- - - - - - - - - - - - - - - - -+   |
   5      |             FFh                 |   |
          +- - - - - - - - - - - - - - - - -+    > destination MAC
   6      |             FFh                 |   |  (broadcast)
          +- - - - - - - - - - - - - - - - -+   |
   7      |             FFh                 |   |
          +- - - - - - - - - - - - - - - - -+  /
   8      |             FFh                 | /
          +---------------------------------+
   9      |             08h                 | \
```

```
      +- - - - - - - - - - - - - - - - -+  \
10    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+   |
11    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+    > source MAC 17)
12    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+   |
13    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+  /
14    |              77h                | /
      +---------------------------------+
15    |              08h                | \
      +- - - - - - - - - - - - - - - - -+  > Type ARP
16    |              06h                | /
      +---------------------------------+
17    |              00h                | \
      +- - - - - - - - - - - - - - - - -+  > Hardware type: Ethernet
18    |              01h                | /
      +---------------------------------+
19    |              08h                | \
      +- - - - - - - - - - - - - - - - -+  > Protocol type: IP
20    |              00h                | /
      +---------------------------------+
21    |              06h                | Hardware size
      +---------------------------------+
22    |              04h                | Protocol size
      +---------------------------------+
23    |              08h                | \
      +- - - - - - - - - - - - - - - - -+  > Opcode: request
24    |              00h                | /
      +---------------------------------+
25    |              08h                | \
      +- - - - - - - - - - - - - - - - -+  \
26    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+   |
27    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+    > Sender MAC 18)
28    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+   |
29    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+  /
30    |              77h                | /
      +---------------------------------+
31    |              00h                | \
      +- - - - - - - - - - - - - - - - -+  \
32    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+    > Sender IP
33    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+  /
34    |              00h                | /
      +---------------------------------+
35    |              00h                | \
      +- - - - - - - - - - - - - - - - -+  \
36    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+   |
37    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+    > Target MAC 19
38    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+   |
39    |              00h                |   |
      +- - - - - - - - - - - - - - - - -+  /
40    |              00h                | /
```

17) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

18) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

19) This MAC address has to be changed to the appropriate MAC address of the actual device used as the KNX IP load generator.

```
        +------------------------------+
  41    |             0Ah              | \
        +- - - - - - - - - - - - - - - -+  \
  42    |             00h              |   |
        +- - - - - - - - - - - - - - - -+    > Target IP
  43    |             00h              |   |
        +- - - - - - - - - - - - - - - -+  /
  44    |             01h              | /
        +------------------------------+
```

The KNX IP load generator SHALL send FORWARD frames when testing a KNXnet/IP Router.

The KNX IP load generator SHALL send PROCESS frames when testing a KNX IP device.

The KNX IP load generator SHALL be able to generate an additional multicast load by sending BURDEN frames in addition to the FORWARD or PROCESS frames.

The KNX IP load generator operates as follows.

The KNX IP load generator transmits FORWARD or PROCESS frames onto the network. After each transmission the KNX IP Load generator pauses for 20 ms.

The transmission of test frames is immediately locked when the KNX IP load generator receives a ROUTING_BUSY frame.

6.2 Tests

6.2.1 Sending data to IP

6.2.1.1 Wait time

Test

Through a manufacturer dependent way the device must send continuously (delay < 40 ms) ROUTING_INDICATION packets (e.g. by using a bus load generator for IP-Routers or a special application for IP-Devices). The Testtool should generate at least 10 ROUTING_BUSYs with random time between. The time between ROUTING_BUSYs must be greater than $t_w + t_{random,max} + t_{bd}$. The time $t_{d,i}$ between a ROUTING_BUSY packet and the next [20] ROUTING_INDICATION packet should be measured. Execute this test with different values for t_w (40 ms, 80 ms and 100 ms).

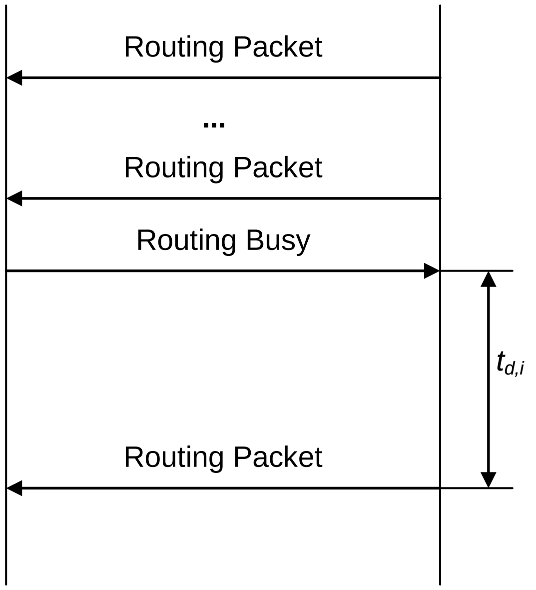


Figure 11 - Sequence for test 2.1.1

Acceptance

All differences $t_{d,i}$ must fulfil $t_w \le t_{d,i} < t_w + t_{random,max}$.
Further must be $t_{random,max} \ge t_{d,max} - t_{d.min} > t_{random,max}/2$.

[20] If a ROUTING_INDICATION is received in the first 200 µs of measuring, the first ROUTING_INDICATION will be ignored.

6.2.1.2 Slow start

Test

Through a manufacturer dependent way the device must send continuously (delay < 40 ms) ROUTING_INDICATION packets (e.g. by using a bus load generator for IP-Routers or a special application for IP-Devices). The Testtool should generate some ROUTING_BUSY packets with a delay < t_w for a second in order to fill the sending queue. The time $t_{slow,i}$ between the ROUTING_INDICATION packets for the next 100 ms must be measured.

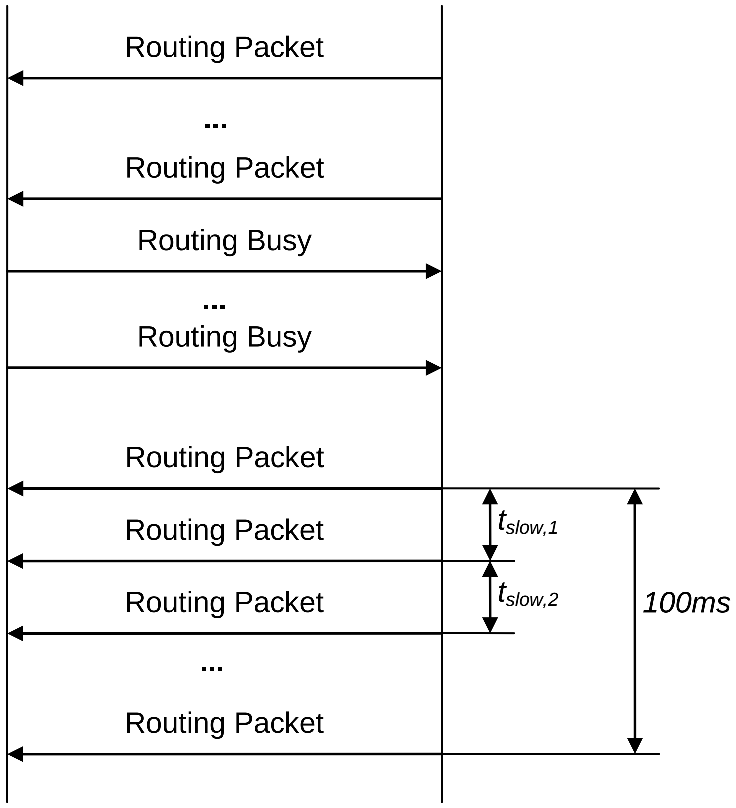


Figure 12 - Sequence for test 2.1.2

Acceptance

The delays between the ROUTING_INDICATION packets $t_{slow,i}$ must be greater than 5 ms.

6.2.1.3 More ROUTING_BUSYs

Test

Through a manufacturer dependent way the device must send continuously (delay < 40 ms) ROUTING_INDICATION packets (e.g. by using a bus load generator for KNXnet/IP-Routers or a special application for IP-Devices). The Testtool should generate at least 10 times a sequence of n ROUTING_BUSY packets with no delay. The time between sending the sequences must be greater than $t_w + t_{random,max} + n{*}t_{slowduration} + n{*}t_{bd}$. The time between the last ROUTING_BUSY packet of a sequence and the next [21] ROUTING_INDICATION packet should be measured.

Execute this test with different values of n (2, 3, 4 and 5).

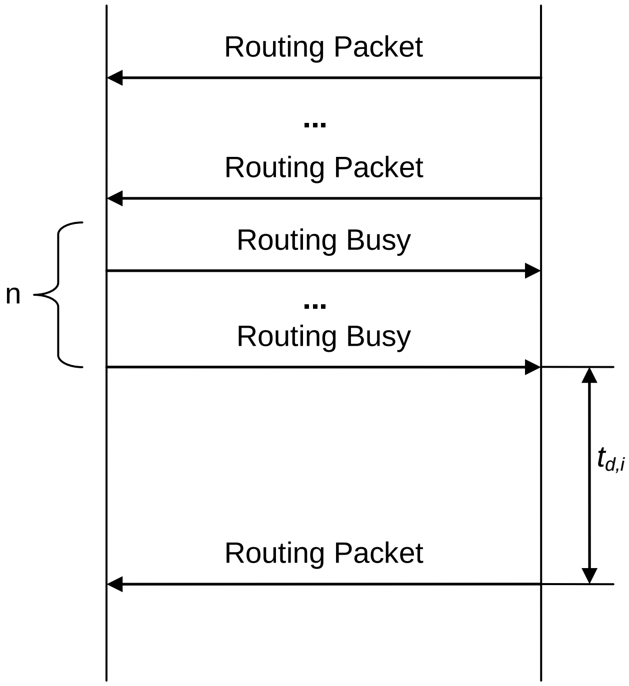


Figure 13 - Sequence for test 2.1.3

Acceptance

All differences $t_{d,i}$ must fulfil $t_w <= t_{d,i} < t_{w+n*trandom,max}$.
Further must be $n{*}t_{random,max} >= t_{d,max} – t_{d.min} > (n{*}t_{random,max})/2$.

[21] If a ROUTING_INDICATION is received in the first 200 µs of measuring, the first ROUTING_INDICATION will be ignored.

6.2.1.4 ROUTING_BUSY received by a device that had sent ROUTING_BUSY

Test

Send ROUTING_INDICATION packets on IP that are addressed to the device or to the Subnetwork if the device is a Router until a ROUTING_BUSY occurs. Through a manufacturer specific way, the processing could be slowed down to get earlier a ROUTING_BUSY. When receiving the ROUTING_BUSY send a test ROUTING_BUSY and a bus telegram via KNX TP.

The time between the test ROUTING_BUSY packet and the next [22] ROUTING_INDICATION packet should be measured.

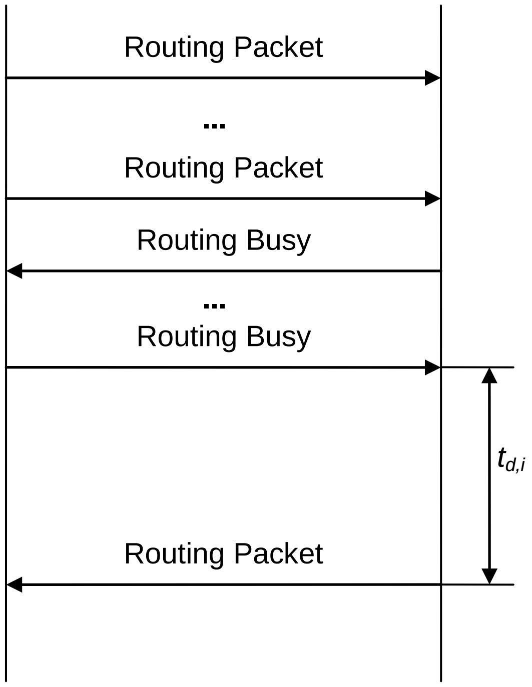


Figure 14 - Sequence for test 2.1.3

Acceptance

The DUT does not send the bus telegram received until the wait time sent with the test ROUTING_BUSY has elapsed.

[22] If a ROUTING_INDICATION is received in the first 200 µs of measuring, the first ROUTING_INDICATION will be ignored.

6.2.1.5 BUSY on KNX Subnetwork (only KNXnet/IP Routers)

Test

A busload generator or the Testtool must generate telegrams [23)] at the KNX Subnetwork (TP, RF, PL) that are to be routed by the KNXnet/IP Router. The Testtool should generate ROUTING_BUSYs with delay < t_w until a BUSY is sent on the KNX Subnetwork.

Acceptance

A BUSY must be seen on KNX Subnetwork (KNX TP1, KNX PL110 or KNX RF).

6.2.1.6 No BUSY on KNX Subnetwork (only KNXnet/IP Routers)

Test

A busload generator or the Testtool must generate telegrams at the KNX Subnetwork (KNX TP1, KNX PL110 or KNX RF) that must be routed by the KNXnet/IP Router. Further telegrams that must not be routed by the KNXnet/IP Router must be generated. The Testtool should generate ROUTING_BUSYs with delay < t_w.

Acceptance

A BUSY must be seen on KNX Subnetwork (KNX TP1, KNX PL110 or KNX RF) for those telegrams that SHOULD be routed.

After telegrams that are not routed no BUSY is allowed to be sent on KNX Subnetwork (KNX TP1, KNX PL110 or KNX RF).

23) Test telegrams SHALL be with KNX Individual Address, KNX Broadcast, and KNX Group Address. Individual tests SHALL be conducted with these different KNX address types.

6.2.1.7 Immediately stop sending on receiving ROUTING_BUSY

Test

Through a manufacturer dependent way the device must send continuously (delay < 40 ms) ROUTING_INDICATION packets (e.g. by using a bus load generator for KNXnet/IP Routers or a special application for KNX IP devices). The Testtool shall generate at least 100 ROUTING_BUSY frames (with t_w = 100 ms) with 125 ms between each. The time $t_{d,i}$ between a ROUTING_BUSY packet and the next ROUTING_INDICATION packet shall be measured with a resolution of less than 100 ns.

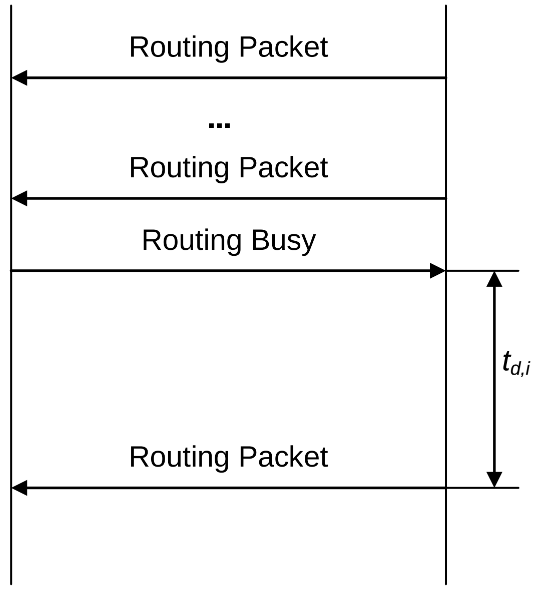


Figure 15 - Sequence for test 2.1.7

Acceptance

The test is passed if the time td,i is never smaller than tw.

If a ROUTING_INDICATION packet was sent within the time $t_{d,i}$ < 1 ms then the device passed the test.

6.2.1.8 Limiting data rate to 50 frames per second

Test

Through a manufacturer dependent way the device must send continuously (delay < 20 ms, at maximum allowed / possible transmission speed) ROUTING_INDICATION packets (e.g. by using a bus load generator for KNXnet/IP-Routers or a special application for IP-Devices). The Testtool should generate 10 times a ROUTING_BUSY packet (t_w = 100 ms) with 125 ms delay between ROUTING_BUSY packets. The Test Tool shall send this sequence every 5 s.

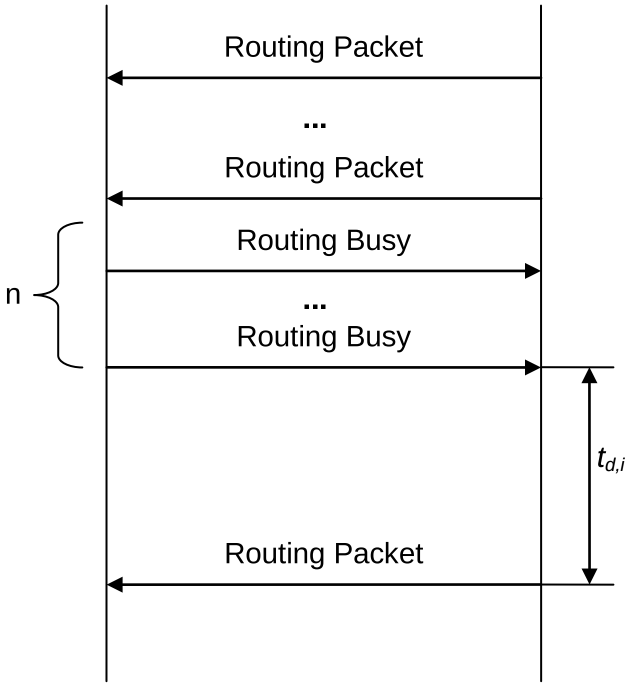


Figure 16 - Sequence for test 2.1.8

Acceptance

The number of frames shall never exceed 50 during any period with a length of one second.

6.2.2 Receiving data from IP

6.2.2.1 Test sending of a correct ROUTING_BUSY

Test

Set the PID_ROUTING_BUSY_WAIT_TIME to value 100 ms [24]. Send ROUTING_INDICATION packets on IP that are addressed to the device or to the Subnetwork if device is a Router until a ROUTING_BUSY occurs. Through a manufacturer specific way, the processsing could be slowed down to get earlier a ROUTING_BUSY. Repeat this test with values of 80 ms, 60 ms and 40 ms for PID_ROUTING_BUSY_WAIT_TIME.

Acceptance

A correct formed ROUTING_BUSY with the correct wait time is seen.

---

[24] If the PID_ROUTING_BUSY_WAIT_TIME Property is read-only then this test shall be used to verify that the fixed busy wait time is sent with the ROUTING_BUSY_INDICATION.

```
        +-------------------------------+ - - - - KNXnet/IP header - - - -
 1      |             06h               |    header size
        +-------------------------------+
 2      |             10h               |    protocol version
        +-------------------------------+
 3      |             05h               | \
        +- - - - - - - - - - - - - - - -+  > service type identifier 0532h (ROUTING_BUSY)
 4      |             32h               | /
        +-------------------------------+
 5      |             00h               | \
        +- - - - - - - - - - - - - - - -+  > total length, 12 octets
 6      |             0Ch               | /
        +-------------------------------+ - - - - Busy Info - - - -
 7      |             06h               |    structure length
        +-------------------------------+
 8      |             00h               |    Device State
        +-------------------------------+
 9      |             00h               | \
        +- - - - - - - - - - - - - - - -+  > Routing busy wait time (64h = 100 ms)
10      |             64h               | /  [50h = 80 ms; 3Ch = 60 ms; 28h = 40 ms]
        +-------------------------------+
11      |             00h               | \
        +- - - - - - - - - - - - - - - -+  > Routing busy control field
12      |             00h               | /
        +-------------------------------+
```

6.2.2.2 Don't stop sending after BUSY

Test

Send ROUTING_INDICATION packets on IP that are addressed to the device and don't stop if a ROUTING_BUSY occurs. Through a manufacturer specific way, the processing could be slowed down to get earlier a ROUTING_BUSY.

Acceptance

Further ROUTING_BUSYs occur after the first and before t_w elapsed.

6.2.2.3 Send packets not addressed

Test

Send ROUTING_INDICATION packets (BURDEN) on IP that are not addressed to the device or the KNX Subnetwork for a duration of 10 s.

The delay between ROUTING_INDICATION packets SHALL be configurable [25].

Acceptance

A ROUTING_BUSY must not occur.

6.2.2.4 KNX Line down (only KNXnet/IP Routers)

Test

Switch bus off through the load switch. The Testtool on IP sends ROUTING_INDICATION packets on IP that are addressed to the Subnetwork for 2 s with a delay of 1 ms [26].

[25] Only the routing protocol is tested not the performance of the device. The delay may be set to ≥ 5ms.
[26] Only the routing protocol is tested not the performance of the device.

Acceptance

A ROUTING_BUSY must not occur.

A ROUTING_LOST_MESSAGE must not occur.

6.2.2.5 Send large number of packets (only KNXnet/IP Routers)

Test

A load generator or the Testtool on IP sends 1000 ROUTING_INDICATION packets that are addressed to the KNX Subnetwork (FORWARD frames) with a delay of 1 ms and reacts correct after receiving a ROUTING_BUSY.

Acceptance

Every ROUTING_INDICATION packet must be sent onto the KNX Subnetwork.

6.2.3 KNX IP performance

6.2.3.1 KNX IP device (BURDEN)

Test

The test setup for KNX IP devices SHALL be used (see 6.1.2.3).

The minimum number of PROCESS and BURDEN frames to be sent by the KNX IP load generator SHALL be at least 1 000 frames per second.

The KNX IP load generator SHALL continuously send BURDEN frames with a minimum transmission rate of 966 frames per second.

Every 30 milliseconds the KNX IP load generator SHALL send a PROCESS frame to the DUT via the IP network.

For 60 seconds the KNX IP load generator shall send PROCESS and BURDEN frames to the DUT via the IP network.

The KNX IP device under test SHALL count the number of PROCESS frames received and processed and capture this number in the Property PID_MSG_TRANSMIT_TO_KNX.

Acceptance

The number of PROCESS frames counted in PID_MSG_TRANSMIT_TO_KNX equals the number of PROCESS frames sent by the KNX IP load generator [27)].

6.2.3.2 KNX IP device (IGNORE)

Test

The test setup for KNX IP devices SHALL be used (see 6.1.2.3).

The minimum number of PROCESS and BURDEN frames to be sent by the KNX IP load generator SHALL be 1 000 frames per second.

The KNX IP load generator SHALL continuously send BURDEN frames with a minimum transmission rate of 966 frames per second.

The remaining bandwidth for the network (e.g. 10 MBit/s network = 14 400 frames/s) shall be filled with IGNORE frames.

[27)] This test may be used to determine the maximum number of frames per second a DUT can process.

Every 30 milliseconds the KNX IP load generator SHALL send a PROCESS frame to the DUT via the IP network.

For 60 seconds the KNX IP load generator shall send PROCESS and BURDEN frames to the DUT via the IP network.

The KNX IP device under test SHALL count the number of PROCESS frames received and processed and capture this number in the Property PID_MSG_TRANSMIT_TO_KNX.

Acceptance

The number of PROCESS frames counted in PID_MSG_TRANSMIT_TO_KNX equals the number of PROCESS frames sent by the KNX IP load generator [28)].

6.2.3.3 KNX IP device (NETWORK)

Test

The test setup for KNX IP devices SHALL be used (see 6.1.2.3).

The minimum number of PROCESS and NETWORK frames to be sent by the KNX IP load generator SHALL be 1 000 frames per second.

The KNX IP load generator SHALL continuously send NETWORK frames with a minimum transmission rate of 966 frames per second.

Every 30 milliseconds the KNX IP load generator SHALL send a PROCESS frame to the DUT via the IP network.

For 60 seconds the KNX IP load generator shall send PROCESS and NETWORK frames to the DUT via the IP network.

The KNX IP device under test SHALL count the number of PROCESS frames received and processed and capture this number in the Property PID_MSG_TRANSMIT_TO_KNX.

Acceptance

The number of PROCESS frames counted in PID_MSG_TRANSMIT_TO_KNX equals the number of PROCESS frames sent by the KNX IP load generator [29)].

6.2.3.4 KNXnet/IP Router (FORWARD)

Test

The test setup for KNXnet/IP Routers SHALL be used (see 6.1.2.2).

The minimum number of FORWARD and BURDEN frames to be sent by the KNX IP load generator SHALL be 1 000 frames per second.

The KNX IP load generator SHALL continuously send BURDEN frames with a minimum transmission rate of 966 frames per second.

Every 30 milliseconds the KNX IP load generator SHALL send a FORWARD frame to the DUT via the IP network.

For 60 seconds the KNX IP load generator shall send FORWARD and BURDEN frames to the DUT via the IP network.

28) This test may be used to determine the maximum number of frames per second a DUT can process.

29) This test may be used to determine the maximum number of frames per second a DUT can process.

The DUT SHALL count the number of FORWARD frames received and processed and capture this number in the Property PID_MSG_TRANSMIT_TO_KNX. The Test Tool SHALL count the number of FORWARD frames forwarded onto the KNX subnetwork.

FORWARD frames SHALL be processed and sent to the recipient on the bus.

Acceptance

The number of telegrams forwarded onto the bus SHALL be equal to the number of FORWARD frames sent by the KNX IP load generator [30)].

6.2.3.5 KNXnet/IP Router (IGNORE)

Test

The test setup for KNXnet/IP Routers SHALL be used (see 6.1.2.2).

The minimum number of FORWARD and IGNORE frames to be sent by the KNX IP load generator SHALL be 1 000 frames per second.

The KNX IP load generator SHALL continuously send BURDEN frames with a minimum transmission rate of 966 frames per second.

The remaining bandwidth for the network (e.g. 10 MBit/s network = 14 400 frames/s) shall be filled with IGNORE frames.

Every 30 milliseconds the KNX IP load generator SHALL send a FORWARD frame to the DUT via the IP network.

For 60 seconds the KNX IP load generator shall send FORWARD and IGNORE frames to the DUT via the IP network.

The DUT SHALL count the number of FORWARD frames received and processed and capture this number in the Property PID_MSG_TRANSMIT_TO_KNX. The Test Tool SHALL count the number of FORWARD frames forwarded onto the KNX subnetwork.

FORWARD frames SHALL be processed and sent to the recipient on the bus.

Acceptance

The number of telegrams forwarded onto the bus SHALL be equal to the number of FORWARD frames sent by the KNX IP load generator [31)].

6.2.3.6 KNXnet/IP Router (NETWORK)

Test

The test setup for KNXnet/IP Routers SHALL be used (see 6.1.2.2).

The minimum number of FORWARD and NETWORK frames to be sent by the KNX IP load generator SHALL be 1 000 frames per second.

The KNX IP load generator SHALL continuously send NETWORK frames with a minimum transmission rate of 966 frames per second.

The remaining bandwidth for the network (e.g. 10 MBit/s network = 14 400 frames/s) shall be filled with IGNORE frames.

30) This test may be used to determine the maximum number of frames per second a DUT can process.
31) This test may be used to determine the maximum number of frames per second a DUT can process.

Every 30 milliseconds the KNX IP load generator SHALL send a FORWARD frame to the DUT via the IP network.

For 60 seconds the KNX IP load generator shall send FORWARD and NETWORK frames to the DUT via the IP network.

The DUT SHALL count the number of FORWARD frames received and processed and capture this number in the Property PID_MSG_TRANSMIT_TO_KNX. The Test Tool SHALL count the number of FORWARD frames forwarded onto the KNX subnetwork.

FORWARD frames SHALL be processed and sent to the recipient on the bus.

Acceptance

The number of telegrams forwarded onto the bus SHALL be equal to the number of FORWARD frames sent by the KNX IP load generator [32)].

[32)] This test may be used to determine the maximum number of frames per second a DUT can process.