BS EN ISO 22510:2020

BSI Standards Publication

Open data communication in building automation, controls and building management — Home and building electronic systems — KNXnet/IP communication

Documents
Document Sharing Hub

Full document details: BS EN ISO 22510:2020

National foreword

This British Standard is the UK implementation of EN ISO 22510:2020. It is identical to ISO 22510:2019. It supersedes BS EN 13321-2:2012, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee RHE/16, Performance requirements for control systems.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2020
Published by BSI Standards Limited 2020

ISBN 978 0 580 99799 0

ICS 91.040.01; 35.240.67

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 29 February 2020.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS EN ISO 22510:2020

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN ISO 22510

February 2020

ICS 35.240.67; 91.040.01

Supersedes EN 13321-2:2012

English Version

Open data communication in building automation, controls and building management - Home and building electronic systems - KNXnet/IP communication (ISO 22510:2019)

Réseau ouvert de communication de données pour l'automatisation, la régulation et la gestion technique du bâtiment - Systèmes électroniques pour les foyers domestiques et les bâtiments - Communication KNX/IP (ISO 22510:2019)

Offene Datenkommunikation für die Gebäudeautomation und Gebäudemanagement - Elektrische Systemtechnik für Heim und Gebäude - Teil 2: KNXnet/IP-Kommunikation (ISO 22510:2019)

This European Standard was approved by CEN on 1 December 2019.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

© 2020 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No. EN ISO 22510:2020 E

Documents
Document Sharing Hub

Full document details: BS EN ISO 22510:2020

European foreword

This document (EN ISO 22510:2020) has been prepared by Technical Committee ISO/TC 205 "Building environment design" in collaboration with Technical Committee CEN/TC 247 “Building Automation, Controls and Building Management” the secretariat of which is held by SNV.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by August 2020, and conflicting national standards shall be withdrawn at the latest by August 2020.

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

This document supersedes EN 13321-2:2012.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Endorsement notice

The text of ISO 22510:2019 has been approved by CEN as EN ISO 22510:2020 without any modification.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO 22510:2020

Contents

Page

Foreword ... **v**
Introduction ... **vi**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **1**
4 Abbreviated terms ... **4**
5 Requirements ... **5**
5.1 Overview ... 5
5.1.1 KNXnet/IP document parts ... 5
5.1.2 Mandatory and optional implementation of IP protocols ... 7
5.2 Core ... 8
5.2.1 Use ... 8
5.2.2 KNXnet/IP frames ... 9
5.2.3 Host protocol independence ... 10
5.2.4 Discovery and self description ... 11
5.2.5 Communication channels ... 13
5.2.6 General implementation guidelines ... 15
5.2.7 Data Packet structures ... 19
5.2.8 IP Networks ... 38
5.2.9 Minimum supported services ... 47
5.3 Device management specification ... 48
5.3.1 Use ... 48
5.3.2 KNXnet/IP device management ... 48
5.3.3 Implementation rules and guidelines ... 59
5.3.4 Data packet structures ... 60
5.3.5 Minimum profiles ... 63
5.4 Tunnelling ... 64
5.4.1 Use ... 64
5.4.2 Tunnelling of KNX telegrams ... 64
5.4.3 Configuration and management ... 68
5.4.4 Frame structures ... 70
5.4.5 Minimum profiles ... 77
5.5 Routing ... 78
5.5.1 Use ... 78
5.5.2 KNXnet/IP routing of KNX telegrams ... 78
5.5.3 Implementation rules and guidelines ... 88
5.5.4 Configuration and management ... 91
5.5.5 Data packet structures ... 91
5.5.6 Minimum profiles ... 93
5.6 Remote diagnosis and configuration ... 94
5.6.1 Use ... 94
5.6.2 Remote diagnosis of KNXnet/IP devices ... 95
5.6.3 Configuration and management ... 95
5.6.4 Data packet structures ... 96
5.6.5 Certification ... 101
5.7 Secured communication ... 101
5.7.1 Use ... 101
5.7.2 Stack and communication ... 102
5.7.3 Management procedures ... 143
5.7.4 Synchronizing timers ... 146
Annex A (normative) **List of codes** ... **148**
Annex B (informative) **Binary examples of KNXnet/IP frames** ... **155**

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

Annex C (normative) **KNXnet/IP parameter object** **175**
Annex D (normative) **Common external messaging interface (cEMI)** **178**
Annex E (normative) **Coupler resources** **210**
Bibliography **221**

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

Foreword

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

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by the European Committee for Standardization (CEN) Technical Committee CEN/TC 247, *Building Automation, Controls and Building Management*, in collaboration with ISO Technical Committee TC 205, *Building environment design*, in accordance with the agreement on technical cooperation between ISO and CEN (Vienna Agreement).

A list of all parts in the ISO 16484 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

Introduction

This document is intended for the design of new buildings and the retrofit of existing buildings in terms of acceptable indoor environment, practical energy conservation and efficiency.

KNXnet/IP is a protocol designed to transport KNX home and building electronic system (HBES) control frames over an IP network. It is used as an infrastructure backbone for connecting KNX sub-networks, as a communication medium for KNX-IP devices and to provide IP based services for clients (e.g. connecting a tool software to a KNX installation). The main advantages of using IP for these purposes are that IP network infrastructure is inexpensive, available almost everywhere and that the distance of two communication parties on an IP network is virtually unlimited.

Widespread deployment of data networks using the Internet protocol (IP) presents an opportunity to expand building control communication beyond the local KNX control bus, providing:

— remote configuration;

— remote operation (including control and annunciation);

— fast interface from LAN to KNX and vice versa;

— WAN connection between KNX systems (where an installed KNX system is at least one line);

— an interface to super ordinate building management and energy management systems.

A KNXnet/IP system contains at least these elements:

— one EIB line with up to 64 (255) EIB devices; or
one KNX segment (KNX-TP1, KNX-RF, KNX-PL110);

— a KNX-to-IP network connection device (called KNXnet/IP server); and typically

— additional software for remote functions residing on e.g. a workstation (may be data base application, BACnet Building Management System, browser, etc.).

KNXnet/IP differentiates between unicast and multicast services. KNXnet/IP unicast services are used to connect a single client to a single KNXnet/IP server (e.g. KNXnet/IP Tunnelling). KNXnet/IP multicast services are mainly used to connect different KNX sub-networks using IP communication on the KNX backbone. The KNXnet/IP routing services are defined for this purpose. KNXnet/IP multicast services build on top of IP multicast.

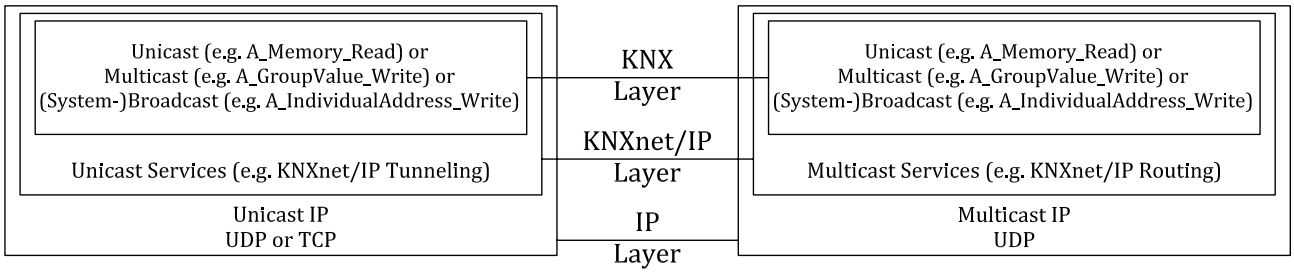

Figure 1 — Unicast and multicast in the sense of KNX, KNXnet/IP and IP

Figure 1 shows a typical scenario where a KNXnet/IP client (e.g. running ETS) accesses multiple KNX installed systems or KNX subnetworks via an IP network. The KNXnet/IP client may access one or more KNXnet/IP servers at a time. For subnetwork, routing server-to-server communication is possible.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

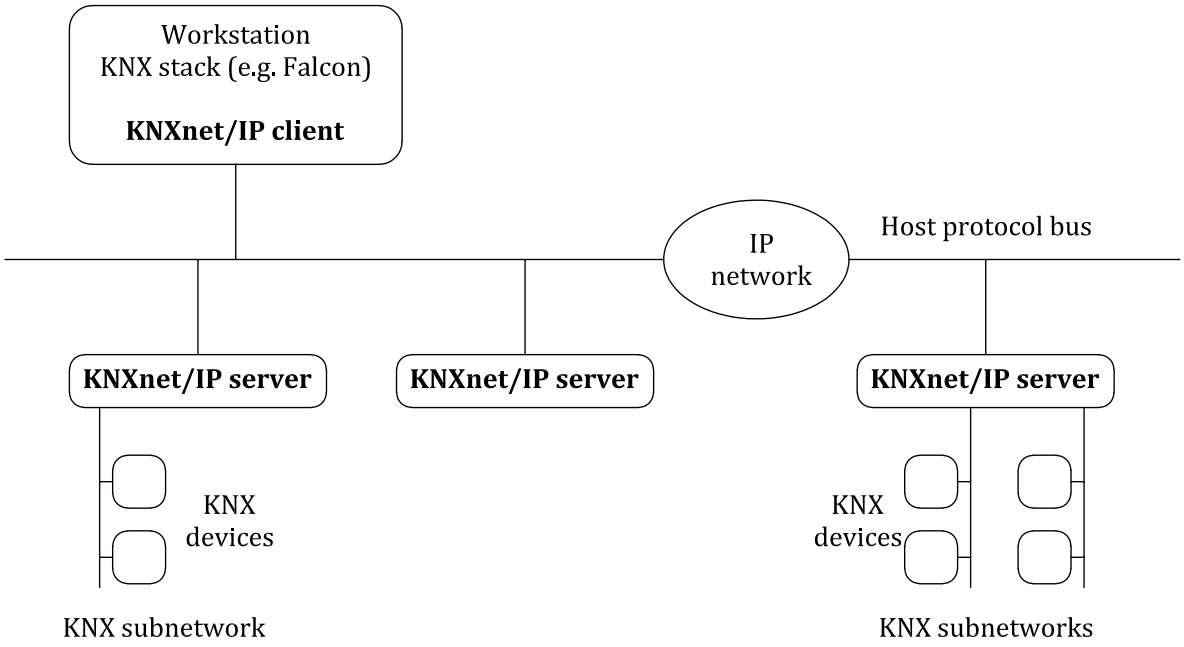

Figure 2 — Device types and configuration examples

Figure 2 shows device types and configuration examples. This document defines the integration of KNX protocol implementations within the Internet protocol (IP) named KNXnet/IP. It defines a standard protocol, which is implemented within KNX devices, Engineering Tool Software (ETS) and other implementations to support KNX data exchange over IP networks. In fact, KNXnet/IP provides a general framework, which accommodates several specialised “Service Protocols” in a modular and extendible fashion.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

Documents
Document Sharing Hub

Open data communication in building automation, controls and building management — Home and building electronic systems — KNXnet/IP communication

1 Scope

This document defines the integration of KNX protocol implementations on top of Internet protocol (IP) networks, called KNXnet/IP. It describes a standard protocol for KNX devices connected to an IP network, called KNXnet/IP devices. The IP network acts as a fast (compared to KNX twisted pair transmission speed) backbone in KNX installations.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
backbone key
key used for encryption and message authentication of secure KNXnet/IP multicast communication in a KNXnet/IP routing multicast group, configured by ETS and a shared secret between all members of the secure KNXnet/IP routing multicast group

3.2
cipher
generic term that denotes the encrypted data

Note 1 to entry: Cipher is the opposite of *plain* (3.22).

3.3
common external message interface
generic structure for medium independent *KNX* (3.14) messages

Note 1 to entry: cEMI (common EMI) frames are used to encapsulate KNX messages within Internet protocol (IP) packets.

3.4
communication channel
logical connection between a *KNXnet/IP client* (3.16) and a *KNXnet/IP server* (3.20) or, in case of routing, between two or more KNXnet/IP servers

Note 1 to entry: A communication channel consists of one or more connections on the definition of the host protocol used for KNXnet/IP.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

3.5
dynamic host configuration protocol
communication protocol for automatic assignment of IP address settings

3.6
domain name system
assigns Internet names to IP addresses

3.7
engineering tool software
software used to configure *KNX* (3.14) devices

3.8
european installation bus
predecessor protocol to KNX

Note 1 to entry: Standard for building controls (EN 50090).

3.9
host protocol address information
structure holding the IP host protocol address information used to address a KNXnet/IP endpoint on another *KNXnet/IP device* (3.17)

3.10
individual address
unique *KNX* (3.14) address of a KNX device in a KNX system

3.11
IP channel
logical connection between two IP host/port endpoints

Note 1 to entry: IP channels are either a guaranteed, reliable TCP (transmission control protocol) or an unreliable point-to-point or multicast (in case of routing) UDP (user datagram protocol) connection.

3.12
Internet control message protocol
extension to the Internet protocol (IP) for error, control, and informational messages

Note 1 to entry: ICMP is defined by RFC[1)] 92 and supports packet containing error, control, and informational messages. The PING command, for example, uses ICMP to test an Internet connection.

3.13
Internet group management protocol
extension to the Internet protocol (IP) for management of IP multicasting in the Internet

Note 1 to entry: IGMP is defined in RFC 1112 as the standard for IP multicasting in the Internet. It is used to establish host memberships in particular multicast groups on a single network. By using Host Membership Reports, the mechanisms of the protocol allow a host to inform its local router that it wants to receive messages addressed to a specific multicast group. All hosts conforming to level 2 of the IP multicasting specification require IGMP.

3.14
KNX
standard for building controls

Note 1 to entry: See EN 50090, ISO/IEC 14543-3-1 to ISO/IEC 14543-3-7.

1) Request for Comment: Internet standards defined by the Internet Engineering Task Force (IETF) are firstly published as RFCs.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

3.15
KNX node
device implementing a *KNX* (3.14) protocol stack and fulfilling the requirements according to the KNX standard

3.16
KNXnet/IP client
application using the KNXnet/IP client protocol to get access to a *KNX* (3.14) subnetwork over an IP network channel

3.17
KNXnet/IP device
implementation of KNXnet/IP services on a *KNX node* (3.15) (*KNXnet/IP server* (3.20)) or any other hardware (*KNXnet/IP client* (3.16))

3.18
KNX/IP domain
all *KNXnet/IP devices* (3.17) in the same network with the same multicast address and the same backbone encryption (either no encryption or encryption with the same key)

3.19
KNXnet/IP router
special type of *KNXnet/IP device* (3.17) that routes *KNX* (3.14) protocol packets between KNX sub-networks

3.20
KNXnet/IP server
KNX (3.14) device with physical access to a KNX network implementing the KNXnet/IP server protocol to communicate with *KNXnet/IP clients* (3.16) or other KNXnet/IP servers (in case of routing) on an IP network channel

Note 1 to entry: A KNXnet/IP server is by design always also a *KNX node* (3.15).

3.21
KNXnet/IP tunnelling
services for point-to-point exchange of *KNX* (3.14) telegrams over an IP network between a *KNXnet/IP device* (3.17) acting as a server and a *KNXnet/IP client* (3.16)

3.22
plain
generic term that denotes unencrypted data, of which the content depends on the service and the user and not of confidentiality and authentication

Note 1 to entry: Plain is the opposite of *cipher* (3.2).

3.23
secure session
authenticated, authorized and encrypted *communication channel* (3.4) between one *KNXnet/IP client* (3.16) and one *KNXnet/IP server* (3.20) for unicast communication

3.24
session key
key used for encryption and message authentication in a *secure session* (3.23) between two KNXnet/IP communication parties, created using ECDH in the secure session setup procedure (providing perfect forward secrecy) and only valid for this individual session

3.25
subnet
portion of a network that shares a common address component known as the "subnet address"

Note 1 to entry: Different network protocols specify the subnet address in different ways.

Documents
Document Sharing Hub

© ISO 2019 – All rights reserved

Full document details: BS EN ISO 22510:2020

3.26
time to live
maximum number of IP routers a multicast UDP/IP datagram may be routed through

Note 1 to entry: Each IP router the datagram passes decrements the TTL by one; the local host adapter also does this. When the TTL has reached zero, the router discards the datagram. When sending a datagram from the local host adapter, a TTL of zero means that the datagram never leaves the host. A TTL of one means that the datagram never leaves the local network (it is not routed).

3.27
transmission control protocol over Internet protocol
connection-oriented communication over the Internet

3.28
user datagram protocol over Internet protocol
connection-less communication over the Internet

4 Abbreviated terms

Abbreviated term	Description
AddIL	Length of additional information
AES	Advanced Encryption Standard
Cn	Conditions are specified under note “n”
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
cEMI	Common External Message Interface
CTR	Counter Mode (of Operation)
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
ECDH	Elliptic Curve Diffie–Hellman
EIB	European Installation Bus
ETS	Engineering Tool Software
FDSK	Factory Default Setup Key
HPAI	Host Protocol Address Information
IA	Individual Address
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IV	Initialisation Vector
M	Mandatory
MAC	Message Authentication Code
MC	Message Code
MiM	Man-in-the-Middle
n/a	Not applicable
O	Optional
PBKDF2	Password-Based Key Derivation Function 2
R	Required
SHA	Secure Hash Algorithm

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

Abbreviated term	Description
TCP/IP	Transmission Control Protocol over Internet Protocol
TTL	Time To Live
X	Not allowed
UDP/IP	User Datagram Protocol over Internet Protocol

5 Requirements

5.1 Overview

5.1.1 KNXnet/IP document parts

5.1.1.1 General

This document defines the integration of KNX protocol implementations within the Internet protocol (IP) named KNXnet/IP.

This document defines a standard protocol, which is implemented within KNX devices, Engineering Tool Software and other implementations to support KNX data exchange over IP networks. In fact, KNXnet/IP provides a general framework, which accommodates several specialised “Service Protocols” in a modular and extendible fashion.

The KNXnet/IP specification consists of the following parts:

— Overview;

— Core specification;

— Device management;

— Tunnelling;

— Routing;

— Remote diagnosis and configuration;

— Secured communication.

KNXnet/IP supports different software implementations on top of the protocol. More specifically, these software implementations can be Building Management, Facility Management, Energy Management, or simply Data Base and SCADA (Supervision, Control and Data Acquisition) packages.

Most of these packages need be configured for the specific user application. In order to simplify this process and cut costs for engineering, KNXnet/IP provides simple engineering interfaces, namely a description “language” for the underlying KNX system. This may be done offline, for example generated as an ETS export file, or online by a mechanism that self-describes the underlying KNX system (reading data from the system itself).

KNXnet/IP supports:

— on-the-fly change-over between operational modes (configuration, operation);

— event driven mechanisms;

— connections with a delay time greater than $t_{\text{KNX\_transfer\_timeout}}$ (e.g. network connection via satellite).

5.1.1.2 Overview

"Overview" is this clause.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.1.1.3 Core specification

"Core specification" defines a standard protocol, which is implemented within KNXnet/IP devices and the Engineering Tool Software (ETS) to support KNX data exchange over IP networks.

This specific implementation of the protocol over the Internet protocol (IP) is called KNXnet/IP.

This specification addresses:

— definition of data packets sent over the IP host protocol network for KNXnet/IP communication;

— discovery and self-description of KNXnet/IP servers;

— configuration and establishment of a communication channel between a KNXnet/IP client and a KNXnet/IP server.

5.1.1.4 Device management

"Device management" defines services for remote configuration and remote management of KNXnet/IP servers.

5.1.1.5 Tunnelling

"Tunnelling" defines services for point-to-point exchange of KNX telegrams over an IP network between a KNXnet/IP device acting as a server and a KNXnet/IP client. This point-to-point exchange may be established by a super ordinate system for building automation or management functions or by an Engineering Tool Software. It supports all ETS functions for download, test, and analysis of KNX devices on KNX networks connected via KNXnet/IP servers. This includes changes of single KNX device object properties.

Tunnelling assumes that a data transmission round-trip between ETS or a KNXnet/IP tunnelling client and KNXnet/IP servers takes less than $t_{\text{KNX-transfer\_timeouts}}$.

5.1.1.6 Routing

"Routing" defines services, which route KNX telegrams between KNXnet/IP servers through the IP network.

5.1.1.7 Remote diagnosis and configuration

"Remote diagnosis and configuration" defines services for a point-to-point exchange of KNX telegrams over an IP network between KNXnet/IP routers and/or KNX/IP devices. The services provide means for diagnosing communication settings and for changing these remotely.

5.1.1.8 Secured communication

"Secured communication" defines services for a secured point-to-point exchange of KNX telegrams over an IP network between a KNXnet/IP client and a KNX/IP server. The services provide means for establishing secured communication sessions by authorized KNXnet/IP clients.

5.1.1.9 List of codes

Annex A gives a complete listing of all codes used by KNXnet/IP, to which KNXnet/IP implementations shall conform, depending on the parts of this document supported.

5.1.1.10 Binary examples

Annex B gives binary examples of KNXnet/IP frames.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.1.2 Mandatory and optional implementation of IP protocols

5.1.2.1 General

KNXnet/IP uses existing IP protocols where possible unless their use implies an undue burden with regard to memory and implementation requirements for the intended service.

The following table shows mandatory (M) and optional (O) implementation of IP protocols by KNXnet/IP service types. Although this table refers to the KNXnet/IP server, it also indicates which IP protocols shall be implemented by the KNXnet/IP client. Any non-applicable IP protocol is marked as "n/a". Table 1 shows KNXnet/IP service types and IP protocols.

Table 1 — KNXnet/IP service types and IP protocols

IP protocol	Service type					
	Core	Device management	Tunnelling	Routing	Remote diagnosis and configuration	Secured com-muni-cation
ARP	M	M	M	M	M	M
RARP	O	O	O	O	O	O
Support of fixed IP address	M	M	M	M	M	M
BootP (client)	M	M	M	M	M	M
DHCP (client)	M	M	M	M	M	M
UDP	M	M	M	M	M	M
TCP	O	O	O	n/a	n/a	M
ICMP	M	M	M	M	M	M
IGMP	M	M	n/a	M	O	M
It is essential that either BootP or DHCP is implemented by a KNXnet/IP device. TCP is mandatory for tunnelling v2 required for secured communication.						

Other Internet protocols like NTP (network time protocol), FTP (file transfer protocol), HTTP (hypertext transfer protocol), SMTP (simple message transfer protocol), DNS (domain name system), and SNMP (simple network management protocol) may be used but are not within the scope of the KNXnet/IP protocol.

5.1.2.2 Minimum KNXnet/IP device requirements

KNXnet/IP service types as defined in this document require the implementation of a minimal set of IP protocols for interworking.

KNXnet/IP servers shall implement these IP protocols: ARP, BootP, UDP, ICMP and IGMP. Other IP protocols may be required for specific services.

A KNXnet/IP client shall not assume that a KNXnet/IP server supports KNXnet/IP frames with a total length of more than 508.

5.1.2.3 Network environment

Because KNXnet/IP servers use IP, this document does not require any specific medium carrying the IP datagrams.

It is recommended to use a medium, which carries at least twice the bit rate of all KNXnet/IP routers connected to this medium. For a point-to-point connection, for example PPP, this would be at least 19 200 bit/s; for a network with up to 400 KNXnet/IP servers, this would be at least 8 Mbit/s.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

10BaseT is recommended as a minimum for KNXnet/IP servers using Ethernet as physical medium.

5.1.2.4 Addressing

KNXnet/IP servers are typically connected to one KNX subnetwork and to an IP network. Hence, KNXnet/IP servers have one distinct address for each network they are connected to: one KNX individual address and one IP address.

Additionally, KNXnet/IP servers are assigned to a KNXnet/IP project-installation using the same IP multicast address for all KNXnet/IP servers in a KNXnet/IP project-installation.

5.1.2.5 KNXnet/IP device classes

A KNXnet/IP device can be a software implementation running on a PC. Hence, ETS or any other software implementing KNXnet/IP services is viewed as a KNXnet/IP device.

Definition of KNXnet/IP device classes ensures interoperability between KNXnet/IP devices as well as a minimum set of supported KNXnet/IP services for a given KNXnet/IP device class.

Table 2 lists mandatory and optional service types for device classes.

Device class A encompasses configuration and system maintenance tools. Except for object server services, all other KNXnet/IP services shall be implemented. ETS is a member of this device class.

Device class B defines the minimum set of KNXnet/IP services for KNXnet/IP routers.

Device class C defines the minimum set of KNXnet/IP services for any other KNXnet/IP device. Building, energy and facilities management systems are members of this KNXnet/IP device class. KNXnet/IP device classes are shown in Table 2.

Table 2 — KNXnet/IP device classes

Device class	Service type					
	Core	Device management	Tunnelling	Routing	Remote diagnosis and configuration	Secured communication
A (Configuration and system maintenance tools)	M	M	M	M	M	M
B (KNXnet/IP router)	M	M	M	M	O[a]	O[a]
C (any other KNXnet/IP device)	M	M	O	O	O	O
[a] If secure communication is supported, support of remote diagnosis and configuration is not allowed.						

5.2 Core

5.2.1 Use

The "Core" of the KNXnet/IP specification provides a general host protocol-independent framework, which accommodates several specialised “Service Protocols” in a modular and extendible fashion.

This specification addresses:

— definition of data packets sent over the non-KNX “host protocol” network for KNXnet/IP communication;

— discovery and self-description of KNXnet/IP servers; and

— configuration, establishment and maintenance of a communication channel between a KNXnet/IP client and a KNXnet/IP server.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.2 KNXnet/IP frames

5.2.2.1 General definitions

5.2.2.1.1 Data format

The KNXnet/IP frames described within this document are coded binary.

The data structure specifications are always noted in binary format.

5.2.2.1.2 Byte order

For binary structures, if not explicitly stated otherwise, the byte order shall be big endian mode (Motorola, non-swapped). For plain text formats, the byte order and formatting shall be according to the related protocol specifications.

5.2.2.1.3 Structures

All KNXnet/IP structures follow a common rule: the first octet is always the length of the complete structure (as some structures may have fields of variable length, e.g. strings) and the second octet is always an identifier that specifies the type of the structure. From the third octet on, the structure data follows. If the amount of data exceeds 252 octets, the length octet shall be FFh and the next two octets will contain the length as a 16 bit value. Then the structure data starts at the fifth octet.

5.2.2.2 Frame format

The communication between KNXnet/IP devices is based on KNXnet/IP frames. A KNXnet/IP frame is a data packet sent over the non-KNX network protocol that consists of a header, comparable to the IP header of an Internet protocol data packet and an optional body of variable length. Figure 3 shows KNXnet/IP frame binary format.

```
+-------------------------------+
|   KNXnet/IP Frame             |
|   (Fixed Length)              |
|                               |
+-------------------------------+
|   KNXnet/IP Frame             |
|   (Variable Length)           |
|   Optional                    |
+-------------------------------+
```

Figure 3 — KNXnet/IP frame binary format

The type of KNXnet/IP frame is described by a KNXnet/IP service type identifier in the header. KNXnet/IP services include, but are not limited to, information regarding discovery and description, communication channel (connection) management and KNX data transfer.

5.2.2.3 Header

5.2.2.3.1 Description

Every KNXnet/IP frame, without any exception, consists at least of the common KNXnet/IP header that contains information about the protocol version, the header and total packet length and the KNXnet/IP service type identifier. The KNXnet/IP header may be followed by a KNXnet/IP body, depending on the KNXnet/IP service.

Timestamp information and frame counters are not included in the common KNXnet/IP frame header as this information is closely linked with certain KNXnet/IP service types and will therefore be included in the body of these services as additional information for certain communication channel types. Figure 4 shows the KNXnet/IP header binary format.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Header Length               |   Protocol Version            |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Service Type Identifier                                     |
|   (2 Octet)                                                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Total Length                                                |
|   (2 Octet)                                                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 4 — KNXnet/IP header binary format

5.2.2.3.2 Header length

Although the length of the header is always fixed, it is possible that the size of the header changes with a new version of the protocol. The header length can be used as an index into the KNXnet/IP frame data to find the beginning of the KNXnet/IP body.

5.2.2.3.3 Protocol version

The protocol version information states the revision of the KNXnet/IP protocol that the following KNXnet/IP frame is subject to. It will be stored in binary coded decimal format. The only valid protocol version at this time is 1.0 (represented as hexadecimal 10h).

5.2.2.3.4 KNXnet/IP service

The KNXnet/IP service type identifier defines the kind of action to be performed and the type of the data payload contained in the KNXnet/IP body if applicable. The high octet of the KNXnet/IP service type identifier denotes the service type family and the low octet the actual service type in that family. For a detailed description of the services, see below.

5.2.2.3.5 Total length

The total length is expressing the total KNXnet/IP frame length in octets. The length includes the complete KNXnet/IP frame, starting with the header length of the KNXnet/IP header and including the whole KNXnet/IP body.

5.2.3 Host protocol independence

5.2.3.1 Host protocol

The KNXnet/IP core specification defines the integration of KNX protocol implementations on top of the Internet protocol (IP). It describes the general and host protocol independent as well as the host protocol specific parts of the KNXnet/IP communication.

5.2.3.2 Endpoints

KNXnet/IP defines the Host Protocol Address Information (HPAI) structure, which is the combination of IP address and port number. The HPAI is the data required to send a KNXnet/IP frame to another device. The KNXnet/IP specification uses the term KNXnet/IP endpoint as a logical view of a HPAI to address another KNXnet/IP device for certain well-defined purposes.

Every KNXnet/IP device shall support exactly one device related, bidirectional and connectionless endpoint for discovery if the host protocol requires discovery services. It shall support at least one bidirectional and connectionless endpoint for controlling and at least one bidirectional and connection-oriented endpoint for service type related data transmission. Figure 5 shows the KNXnet/IP server endpoints sample configuration.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

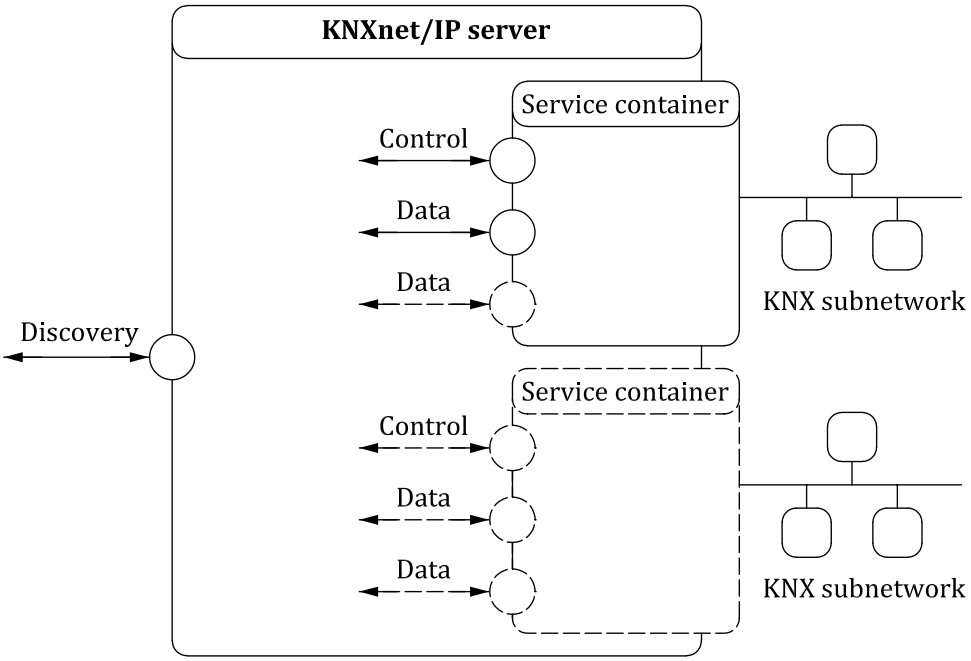

Figure 5 — KNXnet/IP server endpoints sample configuration

The control endpoint uniquely addresses one entity inside the KNXnet/IP server device that shall be capable of providing at least one KNXnet/IP service type.

This entity, called service container, may be connected to a KNX subnetwork. If the KNXnet/IP server device supports more than one KNX subnetwork connection, it is required that every KNX subnetwork shall be represented by a different control endpoint. The KNXnet/IP client therefore considers every service container represented by a control endpoint as one independent entity no matter if they are implemented in only one or two separate KNXnet/IP server devices.

These KNXnet/IP endpoints present a logical view to the communication of a KNXnet/IP device. The actual implementation of these endpoints with different host protocols may use transport medium dependent approaches that differ from this logical view. For example, the bidirectional KNXnet/IP endpoints could be implemented using two unidirectional channels with the host protocol. Therefore, it is possible for one KNXnet/IP endpoint to be represented by multiple HPAIs.

5.2.4 Discovery and self description

5.2.4.1 General

Particularly for networks supporting hot-plug, and where even the address assignment may take place at runtime (e.g. IP address assignment via BootP or DHCP), it is of significant importance to search for devices within a subnetwork without having the need to retrieve network parameters through a non-standardised way and manually input them in the client tool to establish a connection. Furthermore, to get a precise picture of the services supported by the KNXnet/IP server without implementing trial and error, a self-description mechanism is an important feature.

5.2.4.2 Discovery

Any KNXnet/IP server shall implement discovery according to this procedure. If applicable for the host protocol, it is recommended that a KNXnet/IP client implementation supports searching for KNXnet/IP servers instead of requiring manual input.

The discovery operation consists of a SEARCH_REQUEST data packet, sent via multicast using the discovery endpoint, which contains the HPAI of the KNXnet/IP client's discovery endpoint. The HPAI

© ISO 2019 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

may contain a unicast IP address to receive the answers from the different KNXnet/IP servers directly in a point-to-point manner. Typically, it should contain the KNXnet/IP system setup multicast address to ensure reception from KNXnet/IP servers that are on a different subnetwork. Figure 6 shows the discovery procedure.

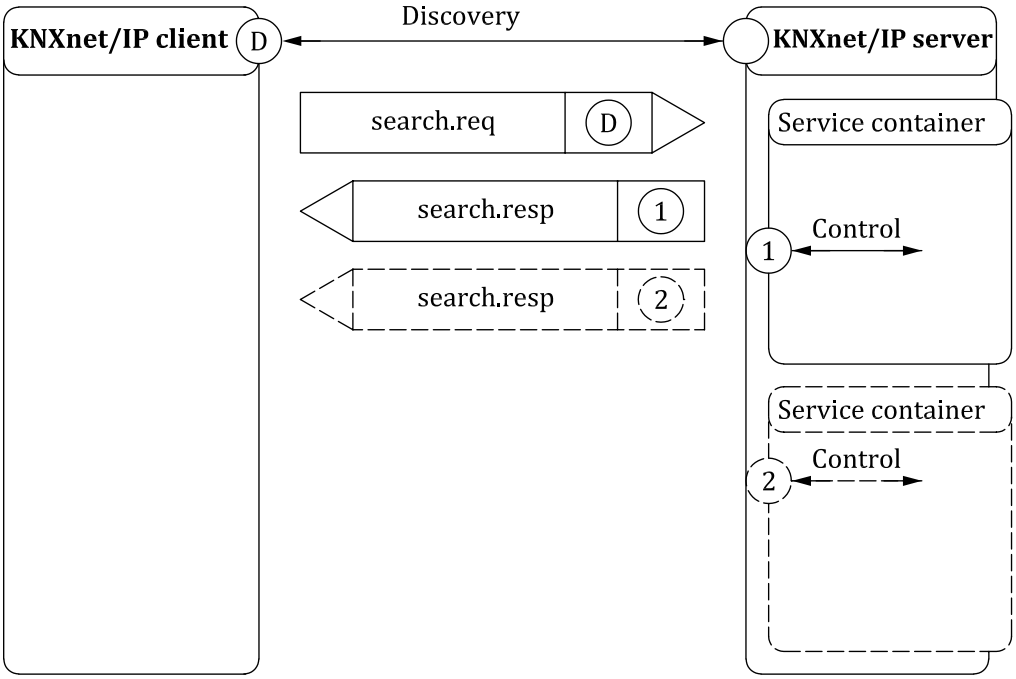

Figure 6 — Discovery procedure

After sending the request, the KNXnet/IP client shall wait for time SEARCH_TIMEOUT for SEARCH_RESPONSE frames from KNXnet/IP servers. After that period of time, any received SEARCH_RESPONSE frame shall be ignored by that client until it starts another discovery cycle. SEARCH_REQUEST frames received by clients from other clients shall be ignored.

Any KNXnet/IP server receiving a SEARCH_REQUEST service shall respond immediately with a SEARCH_RESPONSE data packet to the given HPAI using its discovery endpoint. Such a response contains only the HPAI of the KNXnet/IP server's control endpoint for all further communication.

Any KNXnet/IP server shall support discovery by processing search requests and sending correct responses. A KNXnet/IP server may support links to more than one KNX subnetwork, it shall however send a SEARCH_RESPONSE data packet for the control endpoint of each KNX subnetwork it supports connections to, even if it supports only one data connection at a time.

5.2.4.3 Self-description

Typically, after discovering a KNXnet/IP server, the KNXnet/IP client sends a DESCRIPTION_REQUEST through a unicast or point-to-point connection to all control endpoints of the KNXnet/IP server. It is REQUIRED that every KNXnet/IP implementation supports description requests. Furthermore, before a KNXnet/IP client communicates with a KNXnet/IP server, it should check if the server supports the services requested by the client using the self-description mechanism.

If a KNXnet/IP server receives a valid description request, it shall reply with a DESCRIPTION_RESPONSE packet providing information on the supported protocol version range, its own capabilities, state information and optionally a friendly name for this KNXnet/IP server's KNX connection. As a KNXnet/IP server may support links to more than one KNX subnetwork, it shall support responding to discovery requests for each potential KNX subnetwork connection announced by the discovery responses.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.5 Communication channels

5.2.5.1 General

A communication channel is the data endpoint connection between a KNXnet/IP client and a KNXnet/IP server. Data endpoint connections are established for services requiring point-to-point communication, for example tunnelling or device management. Any point-to-point connection between a KNXnet/IP client and a KNXnet/IP server shall be initiated by the client. Any KNXnet/IP server shall support at least one client connection at a time. It may support more than one client connection at a time, however it shall ensure that existing connections are not affected by accepting new connections (e.g. a KNXnet/IP server shall not accept a tunnelling connection on the same physical access to a KNX subnetwork in different modes, link or busmonitor layer).

5.2.5.2 Establishing a link

To establish a link between a KNXnet/IP client and a KNXnet/IP server, the client shall send a CONNECT_REQUEST frame to the control endpoint of the server. This request provides information on the requested connection type (e.g. data tunnelling or remote logging), general and connection type specific options (e.g. link layer or busmonitor mode) and the data endpoint HPAI that the client wants to use for this communication channel.

NOTE 1 The list of supported layers and services is supposed to be extended in further versions.

Before sending a connection request of a specific type (with specific options), the KNXnet/IP client should check against the self-description information received from the KNXnet/IP server if the server supports the requested connection type and/or all the requested options.

The KNXnet/IP server shall then send a CONNECT_RESPONSE frame in any case back to the KNXnet/IP client requesting to establish the connection, providing the status of the request (ACK/NACK with extended status information if applicable). If the request could be accepted by the server, the CONNECT_RESPONSE frame contains additionally an identifier as well as the HPAI of the data endpoint that the server now prepared for this communication channel

NOTE 2 It should be noted that a KNXnet/IP connection can consist, due to technical reasons, of multiple logical connections at the host protocol layer.

After sending the connection request, the KNXnet/IP client shall wait for the host protocol dependent time CONNECT_REQUEST_TIMEOUT (= 10 s) for the response frame from KNXnet/IP server. After that period of time, any received response frames shall be ignored by that client until it starts another connection request.

The current protocol specification assumes that a connection shall not be shared by multiple clients. A KNXnet/IP server shall therefore not accept multiple connect requests of the same type on, for example the same physical KNX connection, though it may of course implement numerous physical connections, exposing each logically as an independent KNXnet/IP server, if supported by the used host protocol. The client implementation can rely on that restriction and is not required to handle such a connection sharing scenario. Service family exceptions to this general rule are described in the corresponding service family KNXnet/IP chapter. Establishing a data connection is shown in Figure 7.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

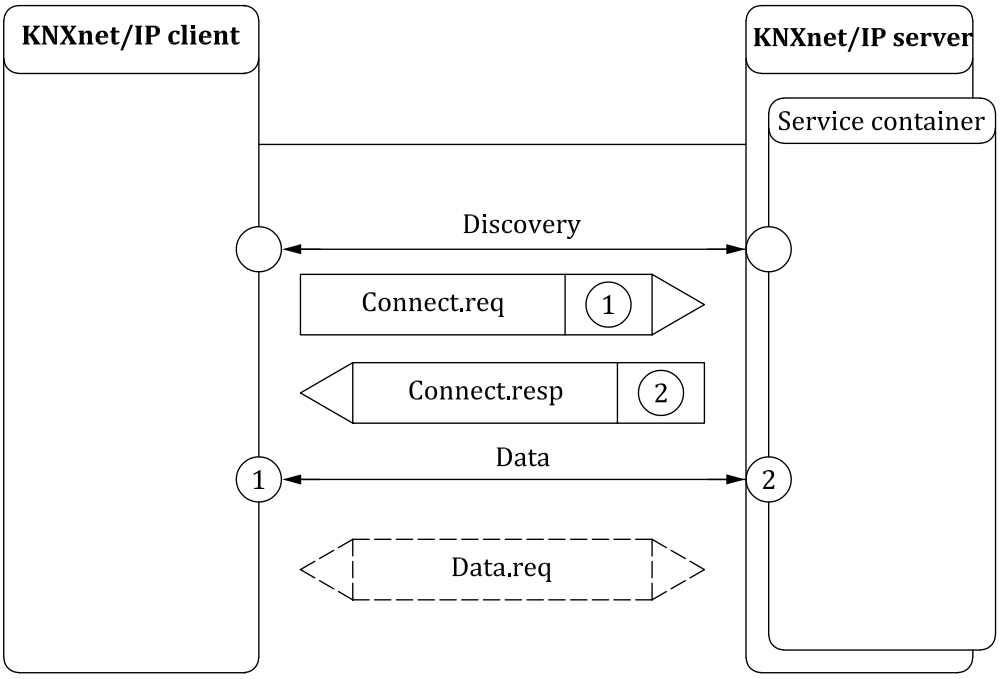

Figure 7 — Establishing a data connection

5.2.5.3 Connection header

5.2.5.3.1 Description

The body of every KNXnet/IP frame sent over an established communication channel starts with data fields that contain additional general information about the data connection. The amount of this data and what type of information is included is determined by several options during the connection phase of a communication channel. The total of these data fields is called connection header and its appearance varies greatly depending on the already mentioned connection options. Only the order in which the different data fields are stored in the connection header is fixed. Figure 8 shows a common connection header.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Communication Channel ID    |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Sequence Counter            |   service type specific       |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Connection Type Specific Header                             |
|   (variable length, optional)                                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 8 — Common connection header

5.2.5.3.2 Structure length

Structure length is the total length of the connection header.

5.2.5.3.3 Communication channel ID

The KNXnet/IP server assigns a communication channel ID to each established communication channel.

© ISO 2019 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.5.3.4 Sequence counter

The sequence counter is maintained for each communication channel. It shall be incremented by one independently for every communication channel for each KNXnet/IP frame sent over the data connection. Both KNXnet/IP devices maintaining the communication channel have independent sequence counters.

Every time a connection is established, the counter for this connection shall be set to zero, so the first KNXnet/IP frame sent on an established communication channel has a sequence counter of zero.

For connections inside a TCP connection, the sequence counter shall not be evaluated by the receiver. The sender may set the sequence counter to zero.

5.2.5.3.5 Connection type specific header items

The connection type specific header items are optional and of variable length depending on the type of the connection.

5.2.5.4 Heartbeat monitoring

Host protocols not providing mechanisms for lifetime check like UDP/IP need a procedure to identify failure of communication, may it be on the KNX or the tunnelling network. To detect such situations heartbeat monitoring is defined and shall be implemented by both KNXnet/IP clients and servers.

The client shall send a CONNECTIONSTATE_REQUEST frame regularly, i.e. every 60 s, to the server's control endpoint, which shall respond immediately with a CONNECTIONSTATE_RESPONSE frame (this counts as heartbeat response).

If the KNXnet/IP client does not receive the heartbeat response within the CONNECTIONSTATE_REQUEST_TIMEOUT (= 10 s) or the status of a received heartbeat response signalled any kind of error condition, the client shall repeat the CONNECTIONSTATE_REQUEST three times and then terminate the connection by sending a DISCONNECT_REQUEST to the server's control endpoint.

If the KNXnet/IP server does not receive a heartbeat request within 120 s of the last correctly received message frame, the server shall terminate the connection by sending a DISCONNECT_REQUEST to the client's control endpoint. The server shall not retrigger the timeout after messages received with unexpected sequence number.

5.2.5.5 Disconnecting

Typically, the client terminates the connection. During normal operation, the client shall send a DISCONNECT_REQUEST to the server's control endpoint to request termination of the data channel connection.

The client should try to disconnect gracefully if possible, even under error conditions. The server may disconnect from the client by sending a DISCONNECT_REQUEST in case of internal problems or if it receives invalid data packets; however, it is recommended to let the client terminate the connection.

The KNXnet/IP device receiving the DISCONNECT_REQUEST from the communication partner shall acknowledge the operation with a DISCONNECT_RESPONSE frame. This data packet signals the final termination of a previously established communication channel.

5.2.6 General implementation guidelines

5.2.6.1 General

This clause defines programming guidelines that shall be taken into account when implementing KNXnet/IP servers or clients, respectively.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.6.2 KNXnet/IP servers

— If a server receives a data packet with an unsupported version field, it shall reply with a NACK message indicating in the status field E_VERSION_NOT_SUPPORTED.

— If an invalid data packet is received, the implementation shall ignore the data packet without taking any further action.

— If a connection is established, all data packets shall be sent with the same protocol version.

— If a connection is established and the protocol version changes within the received data packets, the server shall shut down the connection.

5.2.6.3 KNXnet/IP clients

— If a client receives a data packet with an unsupported version field, it shall reply with a NACK message indicating in the status field E_VERSION_NOT_SUPPORTED. If a connection to that server sending an unsupported protocol version is established, the client shall disconnect. The client may try to reconnect then and re-establish the connection.

— If an invalid data packet is received, the implementation shall ignore the data packet without taking any further action.

— If a connection is established, all data packets shall be sent with the same protocol version.

— If a connection is established and the protocol version changes within the received data packets, the client shall disconnect from the server. The client may try to reconnect then and re-establish the connection.

5.2.6.4 KNXnet/IP router settings

5.2.6.4.1 KNXnet/IP router factory default settings

An important feature for KNXnet/IP routers is that they shall provide proper KNXnet/IP routing without any user intervention. This plug and play routing behaviour requires a standardised factory default configuration.

— Routers shall be shipped with a default individual address of FF00h.

— The routing multicast address equals the system setup multicast address.

— KNX broadcast telegrams shall be routed from one KNX subnetwork to another even if KNXnet/IP routing devices are still being used in their factory default configuration.

— KNXnet/IP routing shall already work even if a valid unicast IP address has not been obtained by or assigned to the KNXnet/IP routing device.

— Routing shall already work and tunnelling shall be available for (further) configuration of the KNXnet/IP router if an individual address has been assigned to a KNXnet/IP router via the KNX subnetwork.

5.2.6.4.2 KNXnet/IP router IP address assignment

KNXnet/IP routers shall support plug and play KNXnet/IP routing out of the box even if a valid unicast IP address was not acquired from a DHCP server, by manual input, or via ETS configuration. This may require that the IP stack can send and receive multicast IP messages although a valid unicast IP address has not been acquired.

If an IP stack does not support multicasting without an assigned unicast IP address then the KNXnet/IP router shall acquire a unicast IP address via AutoIP or by self-assigning the default KNXnet/IP source unicast IP address 0.0.0.0.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

If a valid unicast IP address was not acquired the KNXnet/IP router shall use the default KNXnet/IP source unicast IP address 0.0.0.0 for routing but shall not support KNXnet/IP tunnelling.

5.2.6.5 Initial setup procedures for KNXnet/IP servers

5.2.6.5.1 General

KNXnet/IP devices shall be configurable in the same way as traditional KNX devices. In an unconfigured state KNXnet/IP routers shall operate with default values enabling KNX telegrams to pass from one KNX subnetwork to another, enabling KNXnet/IP routers to transparently replace KNX routers (line and backbone couplers).

This requirement is set under the condition that KNX individual addresses are unique across the IP network. If KNXnet/IP devices of two independent installations are connected to the same IP network or a KNX project consists of multiple installations, the use of one factory default routing configuration cannot guarantee the delivery of KNX telegrams to the intended destination as the same KNX (individual or group) address could be present in different installations. Figure 9 shows a KNX project with multiple installations.

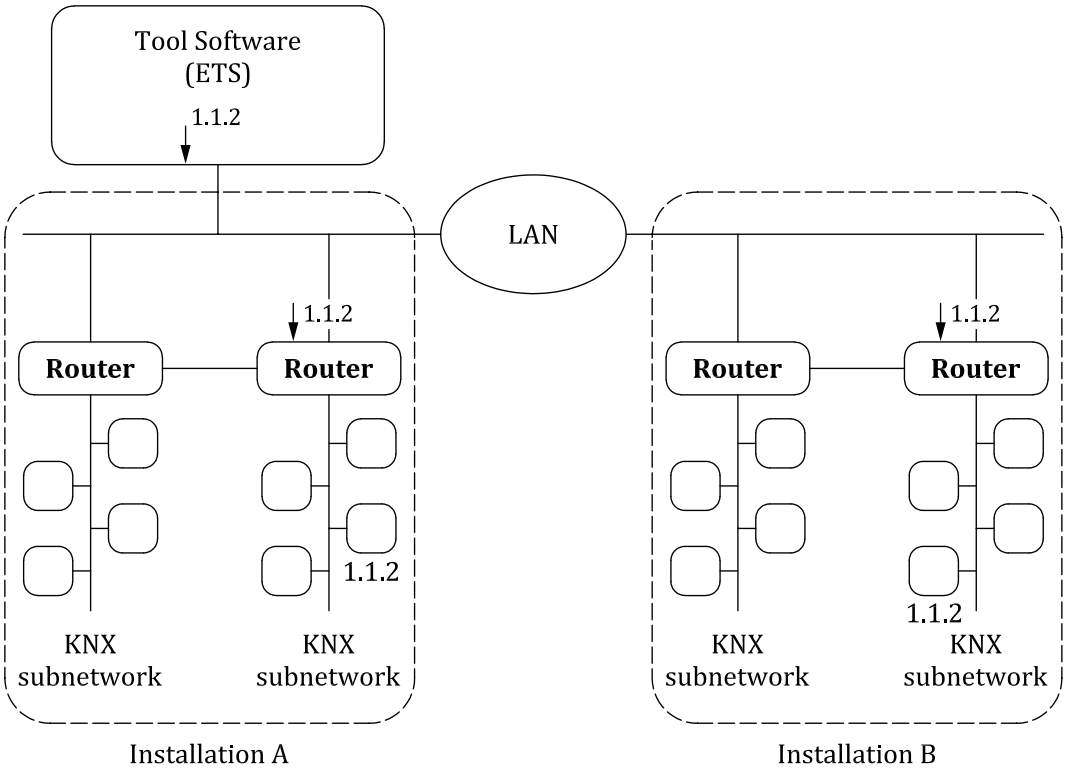

Figure 9 — KNX project with multiple installations

Under the precondition that only one installation is connected to the IP network the following rules apply:

a) Upon power-up, the KNXnet/IP device steps through the procedure described in KNXnet/IP core, 5.2.8.5, IP address assignment, to retrieve an individual IP address.

b) If connected to a KNX serial interface device (RS232) or KNX USB Interface, the tool software can only access KNX devices on the local KNX subnetwork (including the local KNXnet/IP router) as long as not all the KNXnet/IP router devices belonging to one project are successfully configured.

c) The tool software should always attempt to configure KNXnet/IP router devices first to gain access to KNX devices behind possibly still unconfigured KNXnet/IP routers.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

d) If directly connected to the IP network, the tool software can access all KNXnet/IP devices and through configured routers the corresponding KNX subnetworks.

e) The tool software should use the IP network for the setup of projects containing KNXnet/IP devices.

f) It is possible to completely configure KNXnet/IP devices in one single configuration procedure (KNX individual address assignment and parameter download).

g) Assignment of KNX individual address: If the tool software is connected through a KNX serial interface device (RS232) or KNX USB Interface to the KNX network, the traditional KNX management procedure applies for KNXnet/IP devices as well as other KNX nodes. Devices behind unconfigured routers are not reachable. If the tool software is connected to the IP network, the configuration procedure for KNXnet/IP devices (described below) shall apply. Before setting up KNX field media devices the tool software shall firstly configure all KNXnet/IP router devices.

h) Parameter download: If the tool software is connected through a serial interface to the KNX network, the traditional KNX management procedure applies for KNXnet/IP devices as well as other KNX nodes. If the tool software is connected to the IP network, it establishes management connections to every KNXnet/IP device to download the device parameters (see configuration procedure). After having configured all KNXnet/IP router devices, the tool software can access all other KNX nodes of the project for further set-up and download through a KNXnet/IP tunnelling connection.

i) If the SEARCH_RESPONSE answers reveal that KNXnet/IP devices use different IP address spaces, the tool software shall present an error message.

j) If the SEARCH_RESPONSE answers reveal that the IP address of a KNXnet/IP device is different from the network settings of the tool software and the IP address of the KNXnet/IP device is an AutoIP address, the tool software shall present an error message and user control button that enables to reset the KNXnet/IP device via KNXnet/IP routing with the KNX connectionless restart service.

5.2.6.5.2 Configuration procedure for configuration via KNXnet/IP routing

The tool software is connected to a KNX subnetwork (1.1) via a KNX serial interface device (RS232) or KNX USB interface. A second KNX subnetwork (1.2) is connected with the first via two KNXnet/IP routers and a LAN. All devices including the routers are initially unconfigured.

The tool software uses KNX management procedures and (indirectly) KNXnet/IP routing to configure the two KNXnet/IP routers.

This is the configuration procedure for assignment of the KNX individual address and configuration of parameters of a KNXnet/IP device:

a) The tool software requests the user to activate the programming mode of the KNXnet/IP device.

NOTE This can be done by pressing the programming button on the KNXnet/IP device.

b) The tool software sends a KNX read (broadcast). The KNXnet/IP router on this KNX subnetwork sends this broadcast to other KNXnet/IP routers which in turn forward it to their KNX subnetworks and process the telegram themselves if the programming mode is active.

c) The tool software shall check if more than one device answers with "programming mode active" and if so shall abort procedure with a descriptive message about the reason for aborting the procedure. The KNXnet/IP router on this KNX subnetwork forwards the answer(s) to this KNX subnetwork.

d) The tool software shall send a KNX write (broadcast) to write the KNX individual address into the KNXnet/IP device. The KNXnet/IP router on this KNX subnetwork sends this broadcast to other KNXnet/IP routers, which in turn forward it to their KNX subnetworks and process the telegram themselves if the programming mode is active.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

e) The tool software shall establish a KNX connection to the KNXnet/IP device to download the configuration parameters (properties). This requires that the KNXnet/IP router existing on the same KNX subnetwork as the tool software shall be configured first before other KNXnet/IP devices can be configured.

f) After downloading the parameters the tool software shall reset the KNXnet/IP device for the parameters to become effective.

5.2.6.5.3 Configuration procedure for configuration via KNXnet/IP device management

The tool software is directly connected to a LAN. Two KNX subnetworks, (1.1) and (1.2), are connected to the LAN via two KNXnet/IP routers. All devices including the routers are initially unconfigured.

The tool software shall use KNXnet/IP Device management to configure the two KNXnet/IP routers.

This is the configuration procedure for assignment of the KNX individual address and configuration of parameters of a KNXnet/IP device:

a) The tool software shall request the user to activate the programming mode of the KNXnet/IP device.

NOTE This can be done by pressing the programming button on the KNXnet/IP device.

b) The tool software shall send a KNXnet/IP SEARCH_REQUEST frame.

c) The tool software shall check if more than one KNXnet/IP device (service container) answers with device status “programming mode active” and if so, abort procedure.

d) The tool software shall use the IP address from the SEARCH_RESPONSE frame of the KNXnet/IP device to establish a device management connection to the device.

e) The tool software shall set the KNX individual address, project identifier and subnetwork information if applicable.

f) At this point, the tool software can download additional parameters if necessary.

g) After successfully disconnecting the device, management connection or sending a reset_req service to the device, the changed values shall be written and the device shall be restarted.

5.2.7 Data Packet structures

5.2.7.1 General

All KNXnet/IP frames shall have a common header, consisting of header length information, the protocol version, the KNXnet/IP service type identifier, and the total length of the KNXnet/IP frame. Figure 10 shows a KNXnet/IP message 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-+
|   KNXNETIP_SERVICE_TYPE                                       |
|   (2 octets)                                                  |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + sizeof(message body)                       |
|   (2 octets)                                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 10 — KNXnet/IP message header

Requests sent to connectionless KNXnet/IP endpoints shall include information about the return address. This HPAI for the reception of the response information is always the first data of the KNXnet/IP body of all these requests. Additional KNXnet/IP service related data may follow. Response packets do not contain this kind of return address information.

© ISO 2019 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.7.2 Common constants

5.2.7.2.1 HEADER_SIZE_10

This constant with value 06h shall identify the KNXnet/IP header as defined in protocol version 1.0.

5.2.7.2.2 KNXNETIP_VERSION_10

This constant with value 10h shall identify the KNXnet/IP protocol version 1.0.

5.2.7.3 Common error codes

5.2.7.3.1 E_NO_ERROR

This constant with value 00h shall identify a successful operation.

5.2.7.3.2 E_HOST_PROTOCOL_TYPE

This constant with value 01h shall identify that the requested host protocol is not supported by the KNXnet/IP device.

5.2.7.3.3 E_VERSION_NOT_SUPPORTED

This constant with value 02h shall identify that the requested protocol version is not supported by the KNXnet/IP device.

5.2.7.3.4 E_SEQUENCE_NUMBER

This constant with value 04h shall identify that the received sequence number is out of order.

5.2.7.4 Core KNXnet/IP services

5.2.7.4.1 SEARCH_REQUEST

This constant with value 0201h shall identify the KNXnet/IP service type sent by KNXnet/IP client to search available KNXnet/IP servers.

5.2.7.4.2 SEARCH_RESPONSE

This constant with value 0202h shall identify the KNXnet/IP service type sent by KNXnet/IP server when responding to a KNXnet/IP SEARCH_REQUEST.

5.2.7.4.3 DESCRIPTION_REQUEST

This constant with value 0203h shall identify the KNXnet/IP service type sent by KNXnet/IP client to a KNXnet/IP server to retrieve information about capabilities and supported services.

5.2.7.4.4 DESCRIPTION_RESPONSE

This constant with value 0204h shall identify the KNXnet/IP service type sent by KNXnet/IP server in response to a DESCRIPTION_REQUEST to provide information about the server implementation sent.

5.2.7.4.5 CONNECT_REQUEST

This constant with value 0205h shall identify the KNXnet/IP service type sent by KNXnet/IP client to establish a communication channel with a KNXnet/IP server.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.7.4.6 CONNECT_RESPONSE

This constant with value 0206h shall identify the KNXnet/IP service type sent by KNXnet/IP server in response to a CONNECT_REQUEST telegram.

5.2.7.4.7 CONNECTIONSTATE_REQUEST

This constant with value 0207h shall identify the KNXnet/IP service type sent by KNXnet/IP client requesting the connection state of an established connection with a KNXnet/IP server.

5.2.7.4.8 CONNECTIONSTATE_RESPONSE

This constant with value 0208h shall identify the KNXnet/IP service type sent by KNXnet/IP server when receiving a CONNECTIONSTATE_REQUEST for an established connection.

5.2.7.4.9 DISCONNECT_REQUEST

This constant with value 0209h shall identify the KNXnet/IP service type sent by KNXnet/IP device, typically the client, to terminate an established connection.

5.2.7.4.10 DISCONNECT_RESPONSE

This constant with value 020Ah shall identify the KNXnet/IP service type sent by KNXnet/IP device, typically the server, in response to a DISCONNECT_REQUEST.

5.2.7.4.11 SEARCH_REQUEST_EXTENDED

This constant with value 020Bh shall identify the KNXnet/IP service type sent by KNXnet/IP client, to search available KNXnet/IP servers and receive extended information about the available KNXnet/IP servers.

5.2.7.4.12 SEARCH_RESPONSE_EXTENDED

This constant with value 020Ch shall identify the KNXnet/IP service type sent by KNXnet/IP server in response to a SEARCH_REQUEST_EXTENDED.

5.2.7.5 Placeholders

5.2.7.5.1 Host Protocol Address Information (HPAI)

The Host Protocol Address Information structure (HPAI) shall be the address information required to uniquely identify a communication channel on the host protocol. Its size shall vary between different host protocols. For the specific definition of the HPAI, consult the host protocol dependent addendums of the KNXnet/IP specification. Figure 11 shows the HPAI structure binary format.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Host Protocol Code          |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Host Protocol Dependent Data                                |
|   (variable length)                                           |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
```

Figure 11 — HPAI structure binary format

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.7.5.2 Connection Request Information (CRI)

The Connection Request Information structure (CRI) shall be the additional information needed for different types of communication channels to fulfil a connection request. As this structure shall contain two substructures including host protocol independent data as well as host protocol dependent information, the specific definition of the CRI can be found in the description of the connection type with consultancy of the host protocol dependent parts of the KNXnet/IP specification. Figure 12 shows the CRI structure binary format.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Connection Type Code        |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Host Protocol Independent Data                              |
|   (variable length, optional)                                 |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Host Protocol Dependent Data                                |
|   (variable length, optional)                                 |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
```

Figure 12 — CRI structure binary format

5.2.7.5.3 Connection Response Data Block (CRD)

The Connection Request Data Block structure (CRD) shall be the data block returned with the CONNECT_RESPONSE frame. As this structure shall contain two substructures including host protocol independent data as well as host protocol dependent information, the specific definition of the CRD can be found in the description of the connection type with consultancy of the host protocol dependent parts of the KNXnet/IP specification. Figure 13 shows the CRD structure binary format.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Connection Type Code        |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Host Protocol Independent Data                              |
|   (variable length, optional)                                 |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Host Protocol Dependent Data                                |
|   (variable length, optional)                                 |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                                                               |
```

Figure 13 — CRD structure binary format

5.2.7.5.4 Description Information Block (DIB)

The Description Information Block structure (DIB) shall be used by a KNXnet/IP server to return a specific block of device information when responding to a DESCRIPTION_REQUEST.

At least two DIB structures shall be returned with information about the device capabilities on (1) device hardware and (2) supported service families.

More than two DIB structures may be returned in one DESCRIPTION_RESPONSE frame.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

The first octet of each DIB shall contain the length of the DIB structure. The second octet shall declare the DIB structure type. Then the actual data of the DIB shall be appended. The structure shall always have an even number of octets that may have to be achieved by padding with 00h in the last octet of the DIB structure. Figure 14 gives a description of the structure binary format.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Description Type Code       |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Description Information Block data                          |
|   (?? octets)                                                 |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 14 — Description structure binary format

Table 3 lists the valid description type codes.

Table 3 — Description type codes

Description type	Value	Description
DEVICE_INFO	01h	Device information (e.g. KNX medium)
SUPP_SVC_FAMILIES	02h	Service families supported by the device
IP_CONFIG	03h	IP configuration of the device
IP_CUR_CONFIG	04h	Current IP configuration of the device
KNX_ADDRESSES	05h	KNX addresses used by/assigned to the device
SECURED_SERVICES	06h	Secured services
TUNNELLING_INFO	07h	Tunnelling information
EXTENDED_DEVICE_INFO	08h	Extended device information
Reserved	09h–FDh	Reserved for future use
MFR_DATA	FEh	DIB structure for further data defined by device manufacturer
not used	FFh	Not used

The device information DIB shall have the structure as given below in Figure 15.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Description Type Code       |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   KNX medium                  |   Device Status               |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   EIB physical address / KNX Individual Address               |
|   (2 Octets)                                                  |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Project-Installation identifier                             |
|   (2 Octets)                                                  |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   KNXnet/IP device serial number                              |
|   (6 octets)                                                  |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   KNXnet/IP device routing multicast address                  |
|   (4 octets)                                                  |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   KNXnet/IP device MAC address                                |
|   (6 octets)                                                  |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Device Friendly Name                                        |
|   (30 octets)                                                 |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
                              ....
|                                                               |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 15 — Structure of device information DIB

a) KNX medium:

The KNX medium codes shall be as specified in Table 4 below.

Table 4 — KNX medium codes

KNX medium code	KNX medium
01h	reserved
02h	TP1
04h	PL110
08h	reserved
10h	RF
20h	KNX IP

Exactly one single bit shall be set.

b) Device status:

The device status octet shall be as specified in Table 5 below:

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

Table 5 — Device status octet

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Programming mode

c) Project-installation identifier:

The project-installation identifier shall be defined as shown in Table 6:

Table 6 — Project-installation identifier

bit 15–4	bit 3–0
Project number	Installation number

The project-installation identifier shall solely be assigned by ETS and shall be used to uniquely identify KNXnet/IP devices in a project with more than one KNX installation, i.e. more than 15 areas with 12 lines, or in an environment with more than one KNX project.

d) KNXnet/IP device KNX serial number:

The KNXnet/IP device serial number shall be the KNX serial number of the KNXnet/IP device. This information may be used to identify the device or set the individual address.

e) KNXnet/IP device routing multicast address:

The KNXnet/IP device routing multicast address shall be the multicast address used by a KNXnet/IP router for KNXnet/IP routing. KNXnet/IP devices that do not implement KNXnet/IP routing shall set this value to 0.0.0.0. This information may be used if KNXnet/IP routing messages need to be sent to KNXnet/IP routers that do not use the default KNXnet/IP routing multicast address, which shall be equal to the KNXnet/IP system setup multicast address.

f) KNXnet/IP device MAC address:

The KNXnet/IP device MAC address is the Ethernet MAC address of the KNXnet/IP device. This information may be used to identify the device on the Ethernet to a server allocating network resources, specifically the unicast IP address for the KNXnet/IP device.

g) Device friendly name:

The device friendly name may be any NULL (00h) terminated ISO/IEC 8859-1 character string with a maximum length of 30 octets. This name may be used to identify the device to a user. Unused octets are filled with the NULL (00h) character.

The extended device information DIB shall have the structure as given in Figure 16 below.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Description Type Code       |
|   (1 octet)                   |   (1 octet) = 08h             |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Medium Status               |  Reserved                     |
|   (1 octet)                   | (1 octet) = 00h               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Maximal Local APDU length                                   |
|   (2 octets)                                                  |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Device Descriptor Type 0                                    |
|   (2 octets)                                                  |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                                                               |
|                                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 16 — Extended device information DIB

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

h) Medium status:

The contents of this field shall be identical as the contents of the Property PID_MEDIUM_STATUS (PID: 51) in the router object.

— For an IP/TP1 KNXnet/IP router (mask 091Ah), this shall be the connection state of the TP1 connection at the secondary side.

— For a KNXnet/IP tunnelling server on TP1, this shall be the connection state of the TP1 connection.

— For all other KNXnet/IP devices, the field COMMUNICATION_IMPOSSIBLE shall have the value FALSE.

i) Maximal local APDU length:

This shall be the maximal APDU length that is supported by the KNXnet/IP device.

The contents of this field shall be identical as the contents of the Property PID_MAX_LOCAL_APDU_LENGTH (PID: 69) in the cEMI server object.

NOTE This is not be confused with the PID_MAX_APDU_LENGTH (PID: 58) in the router object and which is available in the KNXnet/IP router.

j) Device descriptor type 0:

This shall be the device descriptor type 0 (mask version) of the KNXnet/IP device.

The contents of this field shall be identical as the contents of the Property PID_DEVICE_DESCRIPTOR (PID: 83) in the device object.

The supported service families DIB shall have this structure, see Figure 17:

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Description Type Code       |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Service Family ID           |   Service Family version      |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Service Family ID           |   Service Family version      |
|   (1 Octet)                   |   (1 Octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   ....                        |   ....                        |
|                               |                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Service Family ID           |   Service Family version      |
|   (1 Octet)                   |   (1 Octet)                   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 17 — Supported services families DIB

The service family IDs shall be the high octet of the service type ID. A list of service type IDs can be found in Annex A, Table A.2.

The version of a service family shall refer to the version of the corresponding KNXnet/IP chapter document. This version is only updated when the document itself is updated. Any version of a service family shall be backwards compatible with previous versions, i.e. all services shall be implemented and supported. The service family version is an integer. It does not represent the manufacturer's implementation version.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

The service family version shall refer to the implemented version of the KNXnet/IP service family. The service family versions are defined in the specification papers of the services explicitly or through the definition of profiles.

EXAMPLE If Core v2 is supported then the supported service families DIB include 02h for the core service family (02h).

When used within a SEARCH_RESPONSE or DESCRIPTION_RESPONSE frame, only the supported service families indicated in Table 7 shall be included. When used within a SEARCH_RESPONSE_EXTENDED frame, all supported service families shall be included.

Table 7 — Allowed supported service families

Supported family	Code	Allowed in SEARCH_RESPONSE/ DESCRIPTION_RESPONSE	Allowed in SEARCH_RESPONSE_EXTENDED
Core	02h	Yes	Yes
Device management	03h	Yes	Yes
Tunnelling	04h	Yes	Yes
Routing	05h	Yes	Yes
Remote logging	06h	Yes	Yes
Remote configuration and diagnosis	07h	Yes	Yes
Object server	08h	Yes	Yes
Security	09h	No	Yes
Other		No	Yes

NOTE "Other" denotes placeholder for possible future extensions.

The manufacturer data DIB has this structure, see Figure 18:

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Structure Length            |   Description Type Code       |
|   (1 octet)                   |   (1 octet)                   |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   KNX Manufacturer ID                                         |
|   (2 Octets)                                                  |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Any manufacturer specific data                              |
|   (?? Octets)                                                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 18 — Manufacturer data DIB

The KNX manufacturer ID shall be added to clearly identify the manufacturer. This information is not necessarily encoded in the KNXnet/IP serial number (6 octets).

The manufacturer data DIB may contain any manufacturer specific data.

5.2.7.6 Discovery

5.2.7.6.1 SEARCH_REQUEST

The SEARCH_REQUEST frame shall be sent by a KNXnet/IP client via multicast to the discovery endpoints of any listening KNXnet/IP server. Communication to and from the discovery endpoint of KNXnet/IP devices is only allowed using UDP datagrams. As communication with the discovery endpoint shall be connection- and stateless, the client's discovery endpoint address information shall be included in the KNXnet/IP body. If this information contains a TCP address, the SEARCH_REQUEST frame shall be discarded. Figure 19 shows the SEARCH_REQUEST frame binary format.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
                        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-+
|   SEARCH_REQUEST                                              |
|   (0201h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + sizeof(HPAI)                               |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Discovery endpoint                                          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 19 — SEARCH_REQUEST frame binary format

5.2.7.6.2 SEARCH_RESPONSE

The SEARCH_RESPONSE frame shall be sent by the KNXnet/IP server as an answer to a received SEARCH_REQUEST frame. It shall be addressed to the client's discovery endpoint using the HPAI included in the received frame.

The HPAI of the server's own control endpoint shall be carried in the KNXnet/IP body of the SEARCH_RESPONSE frame along with the description of the device hardware and the supported service families DIB, which shall include the actual service family version and is not restricted to 01h.

EXAMPLE If the tunnelling v2 is supported, then the service family version for tunnelling is 02h.

If the KNXnet/IP server supports more than one KNX connection, the server shall announce each of its own control endpoints in a single SEARCH_RESPONSE frame.

KNXnet/IP servers supporting TCP shall only report the UDP address of their control endpoint in the SEARCH_RESPONSE frame, see Figure 20.

```
                        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-+
|   SEARCH_RESPONSE                                             |
|   (0202h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + sizeof(HPAI) + sizeof(Description)         |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Control endpoint                                            |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   DIB                                                         |
|   device hardware                                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   DIB                                                         |
|   supported service families                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 20 — SEARCH_RESPONSE frame binary format

© ISO 2019 – All rights reserved
Documents Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.7.6.3 SEARCH_REQUEST_EXTENDED

a) Definition:

The SEARCH_REQUEST_EXTENDED frame shall be sent by a KNXnet/IP client to either the discovery endpoints of any listening KNXnet/IP servers or to the control endpoint of a specific KNXnet/IP server.

Communication to and from the discovery endpoint of KNXnet/IP devices is only allowed using UDP datagrams. As communication with the discovery endpoint shall be connectionless and stateless, the KNXnet/IP client's discovery endpoint address information shall be included in the KNXnet/IP body. If this information contains a TCP address, the SEARCH_REQUEST_EXTENDED frame shall be discarded.

Communication to and from the control endpoint of KNXnet/IP devices is allowed using UDP or TCP.

b) Binary format:

Figure 21 shows the binary format of SEARCH_REQUEST_EXTENDED frame.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+
|         Header Length         |        Protocol Version       |                     |
|         (06h)                 |        (10h)                  |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+                     |
|                     Service Type Identifier                   |  KNXnet/IP          |
|                     (020Bh)                                   |  Header             |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+                     |
|                          Total Length                         |                     |
|                          (variable)                           |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+

+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+
|                          HPAI                                 |                     |
|                          Discovery Endpoint                   |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+                     |
|                       SRPs                                    |                     |
|                       (variable)                              |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+
```

Figure 21 — Binary format of SEARCH_REQUEST_EXTENDED frame

c) Search Request Parameter (SRP):

The KNXnet/IP client may include zero or more Search Request Parameter Blocks (SRP) to transfer additional information regarding the search.

NOTE 1 This can be used, for example to restrict the set of devices that are expected to respond or to influence the type of DIBs that the client is interested in.

The general format of a SRP block is given in Figure 22:

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|      Structure Length         | M |        Type Code          |
|         (1 octet)             |        (1 octet)              |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                     Type Specific Data                        |
|                     (?? octets)                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 22 — Binary format of Search Request Parameter Block

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

The KNXnet/IP server shall interpret the SRPs in the following way:

— If the KNXnet/IP server supports the SRP, it shall apply the selection criteria as specified with the specific SRP type.

— Otherwise:

  — If the M (Mandatory) bit is set, the server shall not respond to the search request.

  — If the M (Mandatory) bit is not set, the server shall ignore this SRP and react as if this SRP were not present.

If multiple SRPs are present, the expected result of their combination is specified with the individual SRP types. If the evaluation of any of the SRPs leads to the decision to not respond to the search request then the following SRPs shall not influence this decision and can thus be skipped.

d) SRP type overview:

The SRP type overview is shown in Table 8.

Table 8 — SRP type overview

SRP type	Code	M/O
Invalid	00h	See below
Select by Programming Mode	01h	M
Select by MAC Address	02h	M
Select by Service SRP	03h	M
Request DIBs	04h	M
Reserved	05h to 7Fh	n/a

The KNXnet/IP client shall never use the invalid code 00h. The KNXnet/IP server shall handle an SRP with the invalid code 00h as any other unknown SRP code, i.e. if the M bit is set, the server shall not respond, if the M bit is not set, the SRP shall be ignored.

NOTE 2 The invalid code 00h serves to test the behaviour of the server for unknown SRPs.

NOTE 3 The codes 01h and 02h match the SELECTOR encoding in KNXnet/IP demote configuration and diagnosis. This does however not mean that these two encodings have to be identical for additional selectors/SRP types.

e) SRP type “Select by Programming Mode SRP”:

The client shall include this SRP to indicate that it is interested only in the response from KNXnet/IP servers in which the programming mode is currently enabled.

If programming mode is not enabled in the KNXnet/IP server then it shall not respond to this search request.

The binary format of select by programming mode SRP is shown in Figure 23.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|       Structure Length          | M |         Type Code          |
|          (02h)                  |             (01h)              |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 23 — Binary format of select by programming mode SRP

f) SRP type “Select by MAC Address”:

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

The client shall include this SRP to indicate that it is interested only in the response from the KNXnet/IP server with the given MAC address.

If the KNXnet/IP server's MAC address is different from the given MAC address then it shall not respond to this search request.

The binary format of Select by MAC address SRP is shown in Figure 24.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|      Structure Length         | M |        Type Code          |
|         (08h)                 |           (02h)               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|                      MAC address                              |
|                      (6 octets)                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 24 — Binary format of Select by MAC address SRP

g) SRP type "Select by Service":

The client shall include this SRP to indicate that it is interested only in responses from KNXnet/IP servers supporting the given KNXnet/IP service family in at least the given version.

If the KNXnet/IP server does not support the given service family or supports the given service family only in a lower version then it shall not respond to this search request.

The binary format of Select by Service SRP is shown in Figure 25.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|      Structure Length         | M |        Type Code          |
|         (04h)                 |           (03h)               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|       Service Family ID       |          Minimum Version      |
|         (1 octet)             |          (1 octet)            |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 25 — Binary format of Select by Service SRP

h) SRP type "Request DIBs":

The client shall include this SRP to indicate that it is interested in the listed DIBs. This SRP shall not influence the decision of the KNXnet/IP server whether or not to respond to the search request.

If no "Request DIBs" SRP is present in the request, the KNXnet/IP server shall at least report the basic set of DIBs consisting of the "Device Information DIB", "Extended Device Information DIB" and "Supported Services DIB". The server may include in addition any number of other DIBs in the response.

If a Request DIBs SRP is present in the request then the KNXnet/IP server shall at least report the set of DIBs that are listed in the SRP and that are supported by the server. The server may include in addition any number of other DIBs in the response. The server shall ignore description types that are not recognized or not supported.

This SRP has variable length. If the client is interested in an odd number of DIBs, it shall add an additional description type 0 to make the structure length even.

The binary format of the Request DIBs SRP is shown in Figure 26.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|      Structure Length         | M |        Type Code          |
|         (1 octet)             |           (04h)               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|       Description Types       |          …                    |
|         (? octets)            |                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
```

Figure 26 — Binary format of the Request DIBs SRP

5.2.7.6.4 SEARCH_RESPONSE_EXTENDED

a) Definition:

The KNXnet/IP server shall send the SEARCH_RESPONSE_EXTENDED frame as an answer to a received SEARCH_REQUEST_EXTENDED frame. It shall be addressed to the KNXnet/IP client's discovery endpoint using the HPAI included in the received SEARCH_REQUEST_EXTENDED frame.

The HPAI of the KNXnet/IP server's own control endpoint shall be carried in the KNXnet/IP body of the SEARCH_RESPONSE_EXTENDED frame along with the description of the device hardware and the supported service families. If the KNXnet/IP server supports more than one KNX connection, the KNXnet/IP server shall announce each of its own control endpoints in a single SEARCH_RESPONSE_EXTENDED frame.

KNXnet/IP servers supporting TCP shall only report the UDP address of their control endpoint in the SEARCH_RESPONSE_EXTENDED frame.

The KNXnet/IP server may report the DIBs in the response in any order. Each DIB shall be present only once in the response.

b) Binary format:

The binary format of the SEARCH_RESPONSE_EXTENDED frame is shown in Figure 27.

```
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+
|         Header Length         |        Protocol Version       |                     |
|         (06h)                 |        (10h)                  |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+                     |
|                     Service Type Identifier                   |  KNXnet/IP          |
|                     (020Ch)                                   |  Header             |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+                     |
|                         Total Length                          |                     |
|                         (variable)                            |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+

+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+
|                         HPAI                                  |                     |
|                         Control Endpoint                      |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+                     |
|                       DIBs                                    |                     |
|                       (variable)                              |                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+---------------------+
```

Figure 27 — Binary format of the SEARCH_RESPONSE_EXTENDED frame

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.7.7 Self-description

5.2.7.7.1 DESCRIPTION_REQUEST

The DESCRIPTION_REQUEST frame shall be sent by the KNXnet/IP client to the control endpoint of the KNXnet/IP server to obtain a self-description of the KNXnet/IP server device.

The KNXnet/IP body shall contain the return address information of the client's control endpoint.

The DESCRIPTION_REQUEST frame binary format is shown in Figure 28.

```
                        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-+
|   DESCRIPTION_REQUEST                                         |
|   (0203h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + sizeof(HPAI)                               |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Control endpoint                                            |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 28 — DESCRIPTION_REQUEST frame binary format

5.2.7.7.2 DESCRIPTION_RESPONSE

The DESCRIPTION_RESPONSE frame shall be sent by the KNXnet/IP server as an answer to a received DESCRIPTION_REQUEST frame. It shall be addressed to the client's control endpoint using the HPAI included in the received frame.

The size of the KNXnet/IP body varies depending on the number of DIB structures sent by the server in response to the client's DESCRIPTION_REQUEST. The supported service families DIB shall include the actual service family version and is not restricted to 01h.

EXAMPLE If the tunnelling v2 is supported, then the service family version for tunnelling is 02h.

The DESCRIPTION_RESPONSE frame binary format is shown in Figure 29.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
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-+
|   DESCRIPTION_RESPONSE                                        |
|   (0204h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + sizeof(Description)                        |
|                                                               |

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

KNXnet/IP body

+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   DIB                                                         |
|   device hardware                                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   DIB                                                         |
|   supported service families                                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   DIB                                                         |
|   other device information (optional)                         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 29 — DESCRIPTION_RESPONSE frame binary format

5.2.7.8 Connection management

5.2.7.8.1 CONNECT_REQUEST

The CONNECT_REQUEST frame shall be sent by the KNXnet/IP client to the control endpoint of the KNXnet/IP server. As for every request using control communication, the KNXnet/IP body shall begin with the return address information of the client's control endpoint.

Next follows the CRI, a variable data structure that shall include all additional information that is specific to the requested connection type (and to the underlying host protocol). The exact definition of this structure can be found in the description of the specific connection type, see Table 9.

Table 9 — Connection types

Connection type	Value	V.	Description
DEVICE_MGM_CONNECTION	03h	1	Data connection used to configure a KNXnet/IP device.
TUNNEL_CONNECTION	04h	1	Data connection used to forward KNX telegrams between two KNXnet/IP devices.
REMLOG_CONNECTION	06h	1	Data connection used for configuration and data transfer with a remote logging server.
REMCONF_CONNECTION	07h	1	Data connection used for data transfer with a remote configuration server.
OBJSVR_CONNECTION	08h	1	Data connection used for configuration and data transfer with an object server in a KNXnet/IP device.

Inside the CRI one octet shall determine the type of communication channel requested by this frame and two octets are reserved for the options for this channel. Additional KNXnet/IP documents may define more connection types and additional connection type specific connection options.

The HPAI of the client's data endpoint meant for the requested data connection shall complete the body of the CONNECT_REQUEST frame, see Figure 30.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
                        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-+
|   CONNECT_REQUEST                                             |
|   (0205h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
| HEADER_SIZE_10 + sizeof(HPAI) + sizeof(HPAI) + sizeof(CRI)    |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Control endpoint                                            |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Data endpoint                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   CRI                                                         |
|   Connection request Information                              |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 30 — CONNECT_REQUEST frame binary format

5.2.7.8.2 CONNECT_RESPONSE

The CONNECT_RESPONSE frame shall be sent by the KNXnet/IP server as an answer to a received CONNECT_REQUEST frame. It shall be addressed to the client's control endpoint using the HPAI included in the received frame.

The size of the KNXnet/IP body varies according to the success or failure of the client's CONNECT_REQUEST.

If the connection request could be successfully fulfilled with all the requested options, the body shall contain a communication channel ID that uniquely identifies this connection with the KNXnet/IP server. The communication channel ID shall be the first octet of the body.

The second octet of the body shall contain the status information of the connection request. This status information can contain error information regarding the request itself or regarding the connection type specific information.

Table 10 — Common CONNECT_RESPONSE status codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	The connection was established successfully.
E_CONNECTION_TYPE	22h	1	The requested connection type is not supported by the KNXnet/IP server device.
E_CONNECTION_OPTION	23h	1	One or more requested connection options are not supported by the KNXnet/IP server device.
E_NO_MORE_CONNECTIONS	24h	1	The KNXnet/IP server device could not accept the new data connection because its maximum amount of concurrent connections are already busy.

The HPAI of the server's data endpoint prepared for this data connection shall be the next block of data in the body of a successful CONNECT_RESPONSE frame.

The Connection Response Data Block containing connection type specific response data shall complete the body of a successful CONNECT_RESPONSE frame, see Figure 31.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
                        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-+
|   CONNECT_RESPONSE                                            |
|   (0206h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 2 + sizeof(HPAI) + sizeof(CRD)             |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Communication Channel ID    |   Status                      |
|                               |                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Data endpoint                                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   CRD                                                         |
|   Connection Response Data Block                              |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 31 — CONNECT_RESPONSE frame binary format

5.2.7.8.3 CONNECTIONSTATE_REQUEST

The CONNECTIONSTATE_REQUEST frame shall be sent by the KNXnet/IP client to the control endpoint of the KNXnet/IP server. The first octet of the KNXnet/IP body shall contain the communication channel ID that the KNXnet/IP server uses to uniquely identify the data connection for this connection state request. The second octet shall be reserved for future use.

The HPAI with the return address information of the client's control endpoint shall be added after the communication channel ID.

For the CONNECTIONSTATE_REQUEST frame binary format, see Figure 32.

```
                        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-+
|   CONNECTIONSTATE_REQUEST                                     |
|   (0207h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 2 + sizeof(HPAI)                           |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Communication Channel ID    |   reserved                    |
|                               |                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Control endpoint                                            |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 32 — CONNECTIONSTATE_REQUEST frame binary format

5.2.7.8.4 CONNECTIONSTATE_RESPONSE

The CONNECTIONSTATE_RESPONSE frame shall be sent by the KNXnet/IP server as an answer to a received CONNECTIONSTATE_REQUEST frame. It shall be addressed to the client's control endpoint using the HPAI included in the received frame.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

The first octet of the KNXnet/IP body shall contain the communication channel ID that the KNXnet/IP client passed to the KNXnet/IP with the CONNECTIONSTATE_REQUEST frame.

The second octet of the KNXnet/IP body shall contain the status information of the connection state request. The following Table 11 lists the status codes that are defined for the CONNECTIONSTATE_RESPONSE frame in Figure 33:

Table 11 — CONNECTIONSTATE_RESPONSE status codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	The connection state is normal.
E_CONNECTION_ID	21h	1	The KNXnet/IP server device could not find an active data connection with the specified ID.
E_DATA_CONNECTION	26h	1	The KNXnet/IP server device detected an error concerning the data connection with the specified ID.
E_KNX_CONNECTION	27h	1	The KNXnet/IP server device detected an error concerning the KNX subsystem connection with the specified ID.

```
                        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-+
|   CONNECTIONSTATE_RESPONSE                                    |
|   (0208h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 2                                          |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                        KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Communication Channel ID    |   Status                      |
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 33 — CONNECTIONSTATE_RESPONSE frame binary format

5.2.7.8.5 DISCONNECT_RESPONSE

The DISCONNECT_REQUEST frame binary format is shown in Figure 34. The DISCONNECT_RESPONSE frame binary format is shown in Figure 35.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

```
                        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-+
|   DISCONNECT_REQUEST                                          |
|   (0209h)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 2 + sizeof(HPAI)                           |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Communication Channel ID    |   reserved                    |
|                               |                               |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HPAI                                                        |
|   Control endpoint                                            |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 34 — DISCONNECT_REQUEST frame binary format

```
                        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-+
|   DISCONNECT_RESPONSE                                         |
|   (020Ah)                                                     |
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   HEADER_SIZE_10 + 2                                          |
|                                                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

                         KNXnet/IP body
+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+-7-+-6-+-5-+-4-+-3-+-2-+-1-+-0-+
|   Communication Channel ID    |   Status                      |
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 35 — DISCONNECT_RESPONSE frame binary format

5.2.8 IP Networks

5.2.8.1 General

The KNXnet/IP protocol is used to tunnel or route KNX data over the widely spread Internet protocol (IP), enabling remote access and maintenance across long distances, as well as usage as high-speed backbone for KNX networks.

This part of the document defines which IP parameters and features are supported by KNXnet/IP.

It is assumed that the reader is familiar with the Internet protocols TCP and UDP.

5.2.8.2 Physical vs. logical network

IP networks are not like KNX networks: KNX networks are physical busses by nature. This implies that all devices on the channel will by default receive all packets transmitted on the network. In addition, when a new device is added to the network it is not necessary that other devices on the network become aware of it before they can exchange packets. To transmit a telegram over KNX, it is only necessary that a device be capable of physically transmitting the telegram on the bus, nothing more. If a device is simply physically connected to a KNX network, it is capable of exchanging telegrams with other devices on the channel.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

By contrast, an IP network is not physical, but logical in nature. There are a number of different physical media that can support IP communications and any of them should be capable of supporting tunnelling KNXnet/IP frames. Because it is dealt with a logical channel, it is necessary to "construct" the channel by informing each device on the channel of the existence of the other devices on that channel. In other words, before a device can transmit a packet to some other device on an IP channel it shall be made aware of how to specifically send a packet to that device, i.e. its IP address.

Another significant difference between physical and logical networks is that in the case of typical physical networks it is possible to calculate fixed upper bounds on the length of time it will take a packet to traverse from one device to another once the packet is transmitted on the channel. This is not always possible for IP networks. The deviation of packet delivery times between KNXnet/IP devices on an IP channel are much higher than those experienced with native KNX devices.

The IP channel is used as an intermediary transport mechanism for the KNX telegrams by a variety of KNXnet/IP devices. When a KNXnet/IP packet is transported on an IP channel, an IP message encapsulating the KNX telegram is sent to other KNXnet/IP devices on that IP channel. The IP channel is specified by the list of unicast IP addresses, exactly one for each KNXnet/IP device. There is no maximum to the number of KNXnet/IP devices on a single IP network.

5.2.8.3 Transport mechanisms

IP is a network level protocol. It is designed to operate over a wide range of physical media and link layer protocols. As such, this document does not specify anything about the link or physical layers of the IP stack. The IP protocol stack is shown in Figure 36.

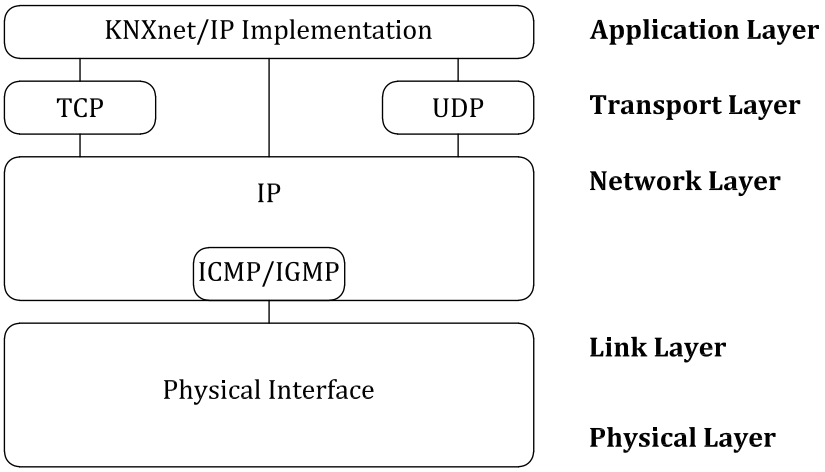

Figure 36 — IP protocol stack

Three most common mechanisms used to transport IP packets are raw IP, TCP and UDP. TCP and UDP are transport protocols built on top of IP.

TCP implements a reliable, connection-oriented end-to-end transport service. It includes provisions to guarantee the correct transmission and to preserve the ordering of the received data stream.

UDP on the other hand implements a best effort datagram service. Raw IP is a mechanism implemented by a number of operating systems to provide host applications with direct access to the IP layer, bypassing any transport service.

Although it is possible to use any of these protocols as a KNXnet/IP host protocol, from an application point of view it is much easier to simply exclude raw IP and restrict the specification to TCP and UDP.

© ISO 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020

5.2.8.4 UDP and TCP

5.2.8.4.1 General

Since the KNX protocol itself employs end-to-end acknowledgment, it is not necessary to guarantee tunnelling IP packets containing KNX data telegrams received between the KNXnet/IP devices. Using a reliable transport service introduces unnecessary protocol overhead in this case, which makes TCP less efficient than UDP or raw IP. On the other hand, there will be configuration and status messages exchanged between KNXnet/IP devices that do not contain tunnelled KNX data and yet need to be received reliably.

TCP has the advantages of reliable delivery service and hence will guarantee that the received packet ordering is preserved. On the down side, it does not support multicast addressing and is less efficient than UDP. TCP also consumes more resources of the KNXnet/IP device to implement than UDP. UDP is more efficient in carrying tunnelled KNX data messages, but for the lack of a reliable delivery, service will not guarantee that the packet ordering is preserved.

Given the increased efficiencies of UDP regarding the transport of KNX data messages and its support of multicast addressing, it will be used as the default to communicate between KNXnet/IP devices. All KNXnet/IP devices shall support UDP. The reliability advantages of TCP may be supported in addition to UDP. TCP support in KNXnet/IP devices is OPTIONAL.

KNXnet/IP servers supporting both UDP and TCP shall accept unicast communication on both IP protocols. Responses and related communication like KNXnet/IP connections shall always use the IP protocol (UDP or TCP) of the original request.

5.2.8.4.2 UDP

To address the reliability and sequencing issue of UDP there shall be acknowledgement frames sent and sequence counters added to the connection header to help in sequencing them.

Using UDP, datagrams can be sent using either unicast or multicast addressing. Unicast is point to point meaning that a datagram is sent from one IP host to a single other IP host. When sending the same datagram to multiple IP hosts as it is necessary for routing of KNX data, it is much more efficient to use multicast addressing. It is therefore recommended that KNXnet/IP devices support both unicast and multicast IP addressing although it is not required that a KNXnet/IP device supports multicasts in order to inter-operate with a KNXnet/IP device that does.

5.2.8.4.3 TCP

a) Opening of a TCP connection:

   TCP connections shall always be initiated by the KNXnet/IP client.

   A KNXnet/IP server shall support at least one single TCP connection at a time and may support any number of further additional concurrent TCP connections at the same time. If no more TCP connections are possible, the KNXnet/IP server shall indicate this by not creating the TCP connection. No KNXnet/IP error code is needed for that.

   A KNXnet/IP client shall not assume that a KNXnet/IP server supports more than one concurrent TCP connection.

b) Closing of a TCP connection:

   1) Sequence of actions to close a TCP connection:

      — The **client** shall close a TCP connection in the following order: first any KNXnet/IP connections, then any KNXnet/IP session, at last the TCP connection.

© ISO 2019 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 22510:2020