2 The OSI Reference Model (ISO 7498)

2.1 Introduction to the OSI Reference Model

2.1.1 Why are Standards Necessary?

When data processing systems and communication networks of different manufacturers were first introduced, there was no compatibility between them. Intercommunication between systems of different levels and standards was therefore difficult and only possible with great efforts. The multifarious networks on the market were designed for very different purposes in the industry and business sectors. Each network was a self-contained system closed to communication with another.

2.1.2 Objective of the OSI Reference Model

The International Standardisation Organisation (ISO) in the seventies therefore decided to formulate a standard architecture for computer networks and information processing. The aim was to…

- develop a standardised model in data communications on the basis of which distributed networks and systems could talk to each other,
- achieve compatibility among networks without insisting on implementation of the complete model for each network,
- achieve open communication (Open System Interconnection OSI) among all users of a heterogeneous computer network.

2.2 The Principle of the OSI Reference Model (OSI)

OSI generally specifies the functions required for data communication and the interaction between these functions. The OSI model stipulates that higher-level functions build on lower-level functions according to well-defined rules. The result is therefore a hierarchical concept consisting of layers.

During the decoding of a telegram by the bus device, each layer decodes part of the telegram. During the creation of a telegram by a bus device, the grouped functions inside an OSI layer all contribute for a part to the telegram.
The above can be respectively compared to peeling off the layers of an onion to arrive to the core (= useful information) or adding onion layers to the core information.

2.2.1 The Seven Layers of the OSI Reference Model

- Layer 1: Physical Layer
  Generation of electrical signals, depending on the transmission medium used; connector pin assignment, control signals, bit coding.
- Layer 2: Data Link layer
  Layer for connections and procedures; line protocol, synchronisation, data security, error handling.
- Layer 3: Network Layer (also referred to as Packet Level)
  Determination of suitable paths for data transmission and switching the links involved (routing), packet control, transport protocol through the entire network.
- Layer 4: Transport Layer
  End-to-end control, transport control
- Layer 4 is responsible for the transport of the entire data volume (i.e. of all the blocks or packets); packet organisation, multiplexing.
- Layer 5: Session Layer (also referred to as Communication Layer)
  Control of logical connections, link to the application process, buffer management *(not used in KNX)*
- Layer 6: Presentation Layer
  Data format conversions, data compression, message coding *(not used in KNX)*.
- Layer 7: Application Layer (also referred to as Process Layer)
  Can be freely defined by the application.

2.2.2 The Two Types of Communication

- Connection-oriented Communication:

The established connection constitutes a logical link between two communication partners. Both partners regard this connection as a link that only exists between them (point-to-point [P2P] connection) and that can only be used by them.

The server (= the device wishing to establish a communication) must establish this connection according to some specified criteria (quality, cost, time delays, etc.) and uses resources to do so. In addition, a protocol or parameters, such as the block length of the data, may be defined. When the connection is established, the actual data exchange (data transport) is executed according to the protocol.

The connection may be aborted at any time by any of the partners (disconnection). The resources used for the link are then released and made available to other functions.

Such communication is highly secure, but is not very bandwidth effective (i.e. a lot of communication is needed to reach only one single communication partner).
When considering the OSI model, during this communication the Transport Layer as well as the Link Layer will generate messages confirming reception of data.
This type of communication can be compared to talking to one single person and checking after each question, whether he/she understood the question.

Use in KNX:

Downloading group addresses, parameters and/or application program from the PC (= ETS) into the KNX device

- Connectionless Communication

This type of communication uses no connection. It is therefore not possible for the involved devices

- to agree upon a protocol or parameters
- to fix criteria such as service quality, cost, etc.

As there is no connection, the address must be sent together with the useful data after each data request. There is no confirmation of the data received and the device sending the information (=server) does not guarantee the correct order of the transmitted data blocks.

Such communication is not secure, but very bandwidth effective: there is no need to establish or break down a communication connection and with one single message it is possible to reach all devices included in the installation. However, only one common acknowledgement of receipt is received. It can therefore not be ensured that all devices have correctly received the sent message.

When considering the OSI model, during this communication only the Link will generate acknowledgements of receipt.

This type of communication can be compared to talking to a group of persons and continuing with the next question if at least one person confirms that he/she understood the message.

Use in KNX:

- Normal operating mode of the KNX (telegrams with group addresses, Multicast, one to many communication)
- Broadcast (one to all communication)

2.2.3 Protocol Data Units (PDUs)

The Protocol data Unit of an OSI layer constitutes

- the data that is contributed by this Layer to the telegram when created;
- The data that is decoded by this Layer from the telegram when received;

2.3 Principle Functions of the Layers; Implementation in KNX

2.3.1 General

The underneath description focuses on the realization of KNX on the TP1 medium. For deviations to this for different media, please consult the relevant chapter in this documentation.

2.3.2 Application Layer (Layer 7)

The application layer basically provides functions for two tasks:

- Assisting the application program in sending and receiving useful information

  Implementation in KNX:
  KNX is a bus using objects for communication (called 'group objects'[1], similar to CAN bus). A group object can be, for instance, the switching state, the daylight intensity or the temperature in a room.
  The application program of a sensor "measures" the physical quantity (e.g. contact state, lux value, temperature, etc) and writes the value of this quantity in the appropriate group object. At the same time it will request to the system software to send the new value of the group object on the bus in order to inform the communication partners of the sensor of this updated value.
  The Application Layer of the addressed actuator(s) will ensure that new received value is written in the group object(s) concerned and subsequently inform the application program(s) of the actuator(s) of the update. The program(s) reads the value of the actuator's group object and executes the required function accordingly. This can be the switching of a relay, reducing the light intensity of a lamp, operating a valve, etc.

  Application programs in a KNX network communicate therefore by way of group objects. Hence no connection is established for data communication (refer to "Connectionless Communication").

  Note that the application program is unable to retrieve the source of the update of the group objects!!

- Assisting devices in understanding configuration messages called 'management services'.

  During the commissioning phase of bus devices, connection-oriented communication is established between devices to download the application programs. It is the Application that will help to understand these configuration commands.

---

[1] Previously referred to as 'communication objects'.

2.3.3 Presentation Layer (Layer 6)

This layer of the OSI model has the task of shielding layer 7 and the application program from the duty of having to deal with different forms of representation of the transmitted data. It interprets and, if necessary, adjusts or converts the syntax of the message.
For example, a file containing the particulars of a group of persons must be transmitted from one bus device to another. The first device uses data records containing fields in the order "name, first name", whereas the second device uses the reverse order. In this case it would be the task of the presentation layer to ensure that each device receives the data record in the correct order; layer 6 must therefore automatically switch the order of the data record fields as required.
With field buses in general, and therefore also in KNX, the problem of converting the form of representation in the transmitted data does not arise. Hence, KNX devices *do not need* an OSI layer 6.

2.3.4 Session Layer (Layer 5)

The principle task of layer 5 according to OSI is to control the communication between two communication partners. This involves:

- opening a dialog,
- closing a dialog,
- aborting or interrupting a dialog,
- continuing a dialog at a later stage,
- picking up a dialog at a defined, backdated point of time in the event of detected errors, so that a faulty data transmission can be repeated correctly.

As the messages exchanged in field buses are short (measured values), these buses require no explicit communication control. Hence, KNX devices *do not need* an OSI layer 5.

2.3.5 Transport Layer (Layer 4)

- **Function during connectionless Communication**

For connectionless communication, the Transport layer in KNX has the following function:

- When sending, ensuring that the value of the changed group object is sent by the link layer with the correct associated (sending) group address.
- When receiving, ensuring that the value of all group objects is updated to which the received group address is linked.

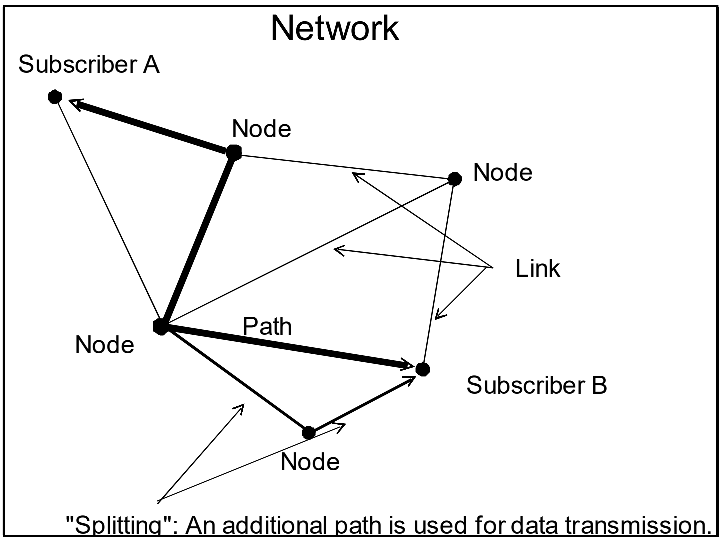
In connectionless communication, the Transport Layer is in other words responsible for checking the Association table loaded in bus devices. This table constitutes the relation between the supported group objects and the attributed group addresses.

Function during connection-oriented Communication

In order to establish a connection-oriented communication, the initiating device will send a Transport Layer connection message, using the individual address[2] of the receiver as the destination address.
During an established connection oriented communication between two communicating bus devices, the Transport Layer of both devices will make use of the Transport Layer "ACK" (positive confirmation) or "NACK messages" (negative confirmation) to acknowledge or reject received telegrams.
On reception of a NACK message from Transport Layer, rejected telegrams (negative acknowledgement) are repeated up to three times. The repetition parameter is fixed in the system software.
Connection oriented communication is monitored by means of timers. If a telegram cannot be transmitted within the set time, or if neither "ACK" nor "NACK" are received from the communication partner, then an established connection is automatically cleared down.
The connection oriented communication is also monitored with an additional sequence number (between 0 and 15). If the sequence number does not have the expected value, the receiver will automatically clear down the established connection.

2.3.6 Network Layer (Layer 3)

A network is usually a combination of nodes which are interconnected by individual links according to a defined topology. The principal task of the OSI layer 3 (network layer) of such a network is to find suitable paths for data transmission, switch the links involved and ensure that the telegrams are directed toward the destination. This is often referred to as "routing".



The nodes in between subscriber A and Subscriber B must ensure a "switching service", so that the message sent by A reaches B. This is comparable to relaying calls in a public telephone network.

[2] Previously referred to as physical address.

A KNX Twisted Pair (KNX-TP) network has the following structure:

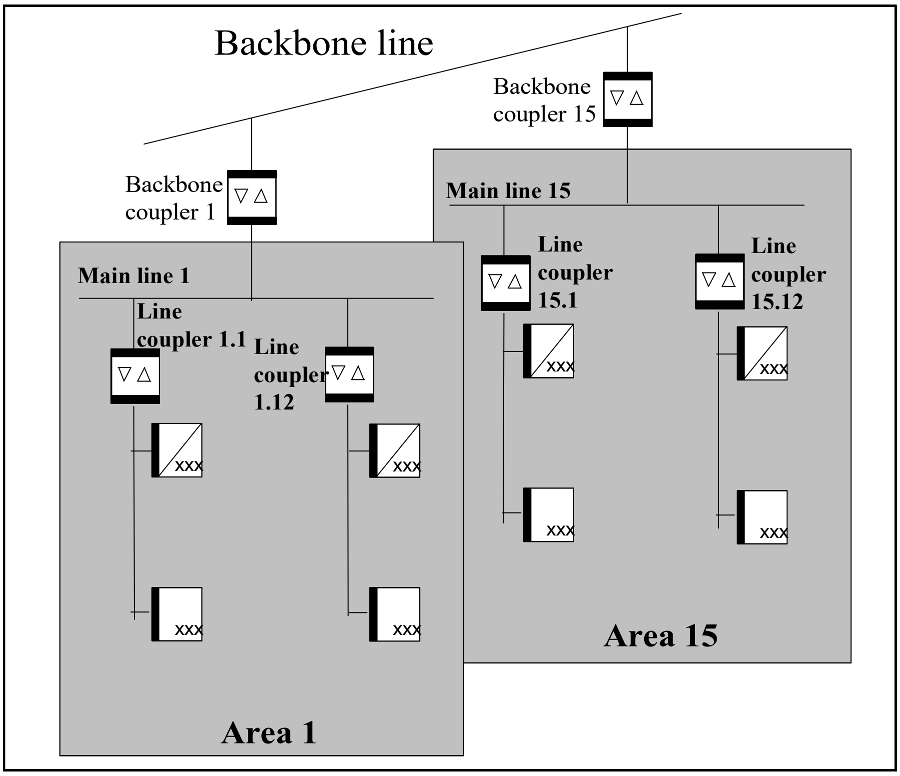


Figure 5: KNX TP network structure

The individual segments of the KNX-TP network constitute the links, whereas the backbone couplers and line couplers are the nodes of the network.

Loops between two lines are not permitted in a KNX-TP network. As the structure of the KNX network is relatively uncomplicated, routing can be solved by fairly simple means.

When a bus device is sending a telegram, the network layer will insert the routing counter value (typically stored in EEPROM) in the sent telegram. This routing counter value is only evaluated by the network layer of backbone or line couplers in the installation (not by normal end devices).

- In the case where the value is 0, in all cases the received telegram will not be routed by the receiving backbone or line couplers
- In the case where the value is between 1 and 6, the received telegram will be routed by the received backbone or line coupler, depending on:
  - In connectionless communication: whether the group address used in the telegram as destination address is contained in its filter table;
  - In connection-oriented communication, whether the individual address used in the telegram as destination address is a device located in the line or area, which is located at “the other side” of the backbone/line coupler where the telegram was originally received. During routing, the coupler will decrement the received value of the routing counter.
- In the case where the value is 7, in all cases the received telegram will be routed by the receiving backbone or line couplers.

It is checked during certification conformity testing whether KNX devices only use routing counter values equal to or lower than 6. The use of routing counter 7 is restricted to ETS only.
The routing of telegrams in the above described way, ensure that telegrams are only routed to lines and areas where the telegram is needed (or on its way to its final destination). This considerably reduces bus load.
The routing mechanism works in a similar way in media couplers between TP and PL, PL backbone couplers, IP routers ... For more information, see chapter ‘couplers’.

2.3.7 Data Link Layer (Layer 2)

General Task of Link Layer
According to OSI, this layer has the task of ensuring the "error-free" transmission of telegrams within a link, i.e. between two nodes of a network (read for KNX: line or backbone couplers) or between a bus device and a node.
Hence information such as synchronisation characters, sequence numbers, error check field and other control characters are included in telegrams, in addition to the actual data to be transmitted.

General Structure of KNX Link Layer Telegram
The figure below illustrates the structure of a KNX telegram transmitted by layer 2.

Control field	Source address	Receiver address	N_PDU	Check field
8 bit	16 bit	16 bit		8 bit

The source address is the individual address of the bus device.
The receiver address can either be a group address (connectionless communication) or an individual address (connection-oriented communication): this is indicated by the first bit in the N_PDU (see later).
The N_PDU is the information that is destined to/included in the telegram by the OSI layers above Link Layer (in the case of KNX: Network, Transport and Application Layer).
The other fields are explained in detail underneath.

Structure of control field

The control field, comprising of 8 bits, has the following structure: Please note that inside a character, D0 is the first bit to be sent on the bus.

D7	D6	D5	D4	D3	D2	D1	D0
1	**0**	R	**1**	*P*	*P*	**0**	**0**

The **bold values** of the control field must be set to the indicated value: if not, the Link Layer will reject the telegram as an invalid KNX telegram.
D0 and D1 constitute the preamble bits of the telegram: the preamble bits ensure that any spikes that may occur on the bus line are not misinterpreted as start bits.
The bits marked with *P* determine the sending priority of the telegram: four levels are defined:

00	Priority 1	System functions
10	Priority 2	Alarm functions
01	Priority 3	Normal mode, high priority
11	Priority 4	Normal mode, low priority

Note: The term "priority" refers to the priority defined at layer 7 for the group objects concerned. This priority is passed through all layers down to layer 2.

The bit D5 (R) indicates whether the telegram is a repeated telegram (= value 0) or not (= value 1).
As the zero overrides the one (see Layer 1), telegrams of higher priority are handled first in the event of a collision. The same applies to telegrams that are repeated; they too will be handled with (higher) priority.

Structure of Source address

The underneath figure shows the structure of the individual address of the sender.
If two devices belonging to one and the same electrical segment (see Layer 1) transmit simultaneously, their telegrams differ in at least one item: the individual address of the sender. As a result, a collision will occur, at the latest, upon transmission of the sender address. The bus device whose telegram has a 0 and occupies the first position in the sender address while all the other devices are simultaneously sending a 1, can transmit its telegram without any interruption (CSMA/CA, again note the sending sequence of the bits: data bit 0 is sent first, line and area addresses are sent before the device address).

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Area				Line				Device address							
0= backbone				0 = main line				0 = coupler							
1 to 15 = area				1 to 15 line				1 to 64= line							
								Above 64: line extension, other line segment							

Structure of Receiving address

The receiving address is an individual address as indicated above for point-to-point connections (connection-oriented communication). For multicast or broadcast addressing (connectionless communication), addresses called “group addresses” are used as receiver addresses and have the following structure:

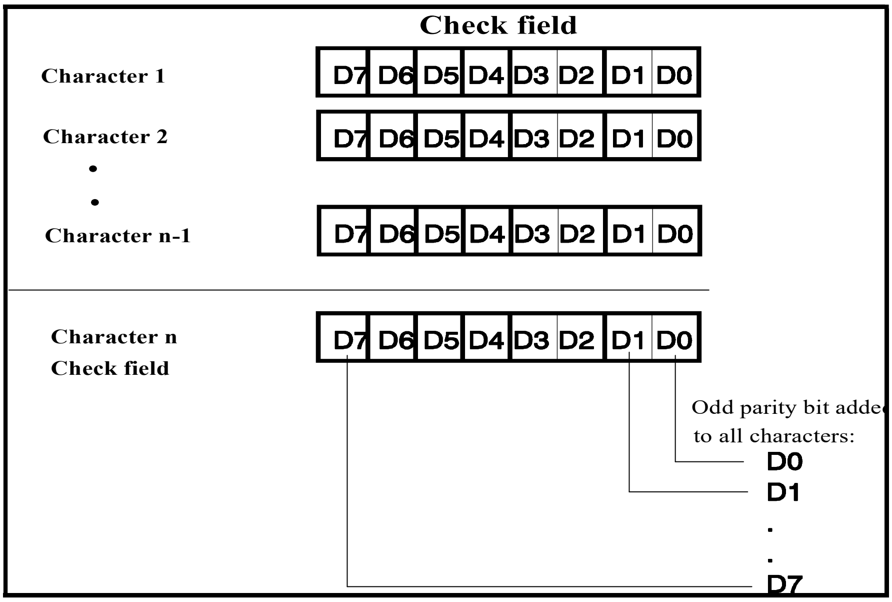
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	Main Group				Subgroup										
	Main Group				Middle Group			Sub Group							

D7
T

The group address is coded on 15 bits (Bit D15 is a reserved bit).
The first row shows how the group address is represented in a two level hierarchy (main/sub), while the second in a three level hierarchy (main/middle/sub). Note that physically on the bus, there is no difference between a two and a three level group address!
The bit D7 in the next character (marked ‘T’) determines whether the preceding address is of the type group (=value 1) or individual (=value 0).

Structure of the check field

The check field is generated according to the cross check method:



Acknowledgment of telegrams by Link Layer

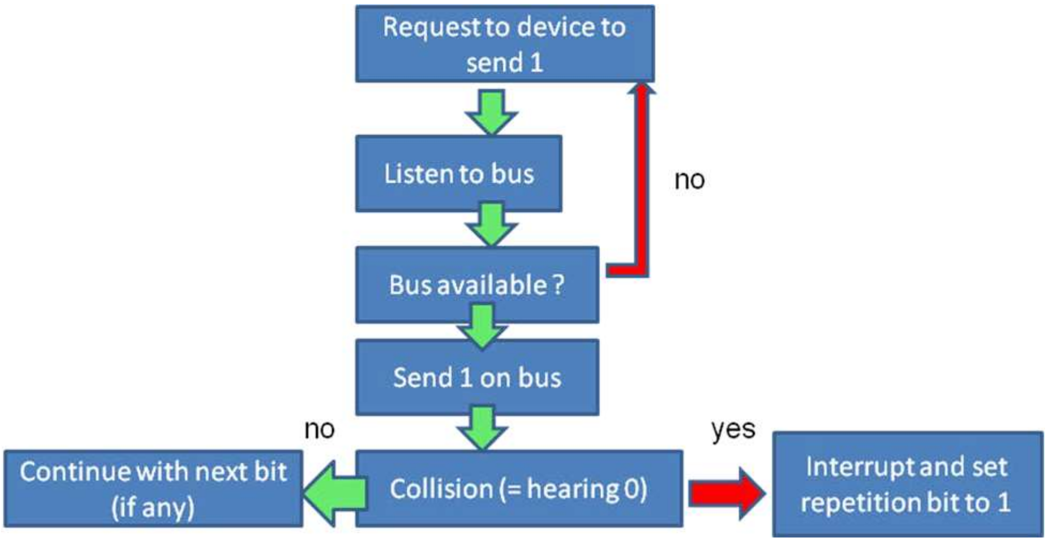
To ensure transmission acknowledgement, it is necessary that the Link Layer of the addressed bus devices or the line/backbone coupler immediately send an acknowledgement all within a specified time frame (Immediate Acknowledgement / Immediate Not Acknowledge ("IACK", "INACK")).
The acknowledgement of the type "BUSY" controls the data flow. It must be transmitted in the same time frame as IACK, INACK by the device that is momentarily busy.
If the sending layer 2 receives an INACK or BUSY message, or a faulty or no IACK acknowledgement, then the telegram or message will be repeated.
Repetition telegrams are marked as such to ensure that bus devices that have acknowledged a transmission with IACK, do not send this message to the upper layers (i.e. Network Layer) a second time.
The following formats for Immediate Acknowledgements are defined:

- Negative Acknowledgement: coding 0Ch (00001100b)
- Busy Acknowledgement: coding C0h (11000000b)
- Positive Acknowledgement: coding CCh (11001100b)

All the addressed devices acknowledge *simultaneously* (group acknowledgement). If several devices of one electrical segment (link) acknowledge differently, due to the selected coding the BUSY message overwrites the IACK and INACK messages. INACK overwrites IACK.

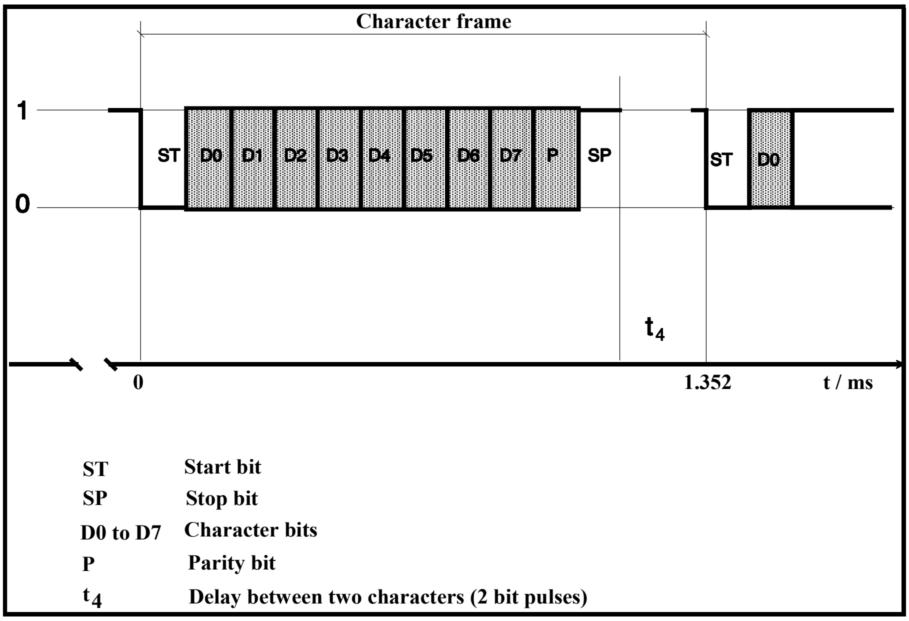
Error detection and bus access control
In KNX, layer 2 uses the cross-checking method for error detection. This is a combination of horizontal and vertical parity checks.

CSMA/CA is used for bus access control. Information on the status of the bus is conveyed to layer 2 via layer 1 ("Bus available" or "Collision").



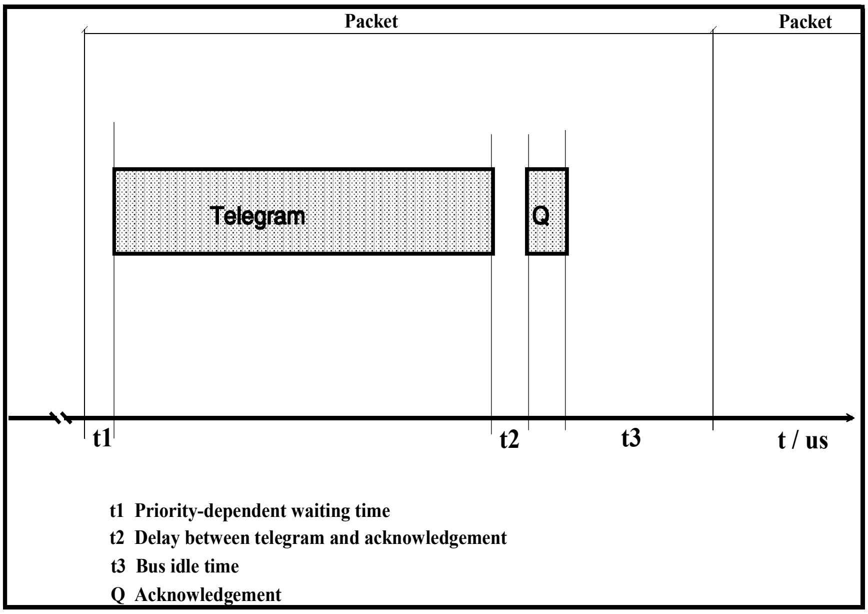
Telegrams of the first and second priorities as well as repetition telegrams can be transmitted immediately when the Layer 1 indicates that the bus is available, whereas messages of the lower priorities are delayed by the time t_1 (= 3 bit pulses).

Since the transmitted character is heard bit-by-bit, the (or several) device(s) which simultaneously sends a 0 bit will not notice collision and will therefore continue to send as if no collision had occurred. This means that no time is lost since there is always an incoming telegram.
The bits of the telegrams are transmitted character by character (1 character = 8 bits) according to the Start-Stop procedure (even parity, 1 stop bit). A delay t_4 of 2 bit pulses separates one character from the next. This means that a total of 13 bit pulses are required per character.



The protocol foresees a transmission break t_2 of at least one character width after the sending of the entire telegram. The next character width is reserved for the reception of acknowledgement, which must be sent by the addressed bus devices of the link or of the line coupler / backbone coupler / line repeater.

A bus idle time t_3 of 50 bit pulses is observed after the transmission of the acknowledgement (IACK, INACK or BUSY).



The total time t_{tot} required for the transmission of the telegram and the acknowledgement and the observance of the bus idle time is calculated as follows:

$$t_{tot} = t_1 + t_3 + (n+2) * 13 * bitpulses,$$
$$8 \le n \le 23$$
$$20ms \le t_{tot} \le 40ms$$

2.3.8 Physical Layer (Bit Transmission, Layer 1)

The lowest layer in the OSI Reference Model is called the physical layer. This layer is concerned with the nature of the signals. It has the task of identifying the bits received from layer 2, converting them to physical signals such as voltages, currents and electromagnetic waves (radio or optical signals) and finally transmitting these signals over the bus transmission media (copper cable, optical fibre etc).

Principal aim of the services provided by layer 1 is to shield layer 2 from the physical means used for bit transmission. It is thus ensured that the upper layers of the network remain independent of the transmission physics used, so that principally the transmission media could be changed without having any effect upon the upper layers.
The physical layer contains the protocol, defines the cabling and wiring between devices and contains the specifications for electromechanical components such as plugs and connectors.

The bit transmission layer of the KNX network is characterised as follows:

The KNX-TP network comprises one or several electrical segments. Each segment has one or two power supplies but - as per definition - does not contain any line couplers.

Technical data of an electrical segment:

- Random topology.
- Total capacitance of one segment:
  without bus devices, line coupler, line repeater: 100 nF max.
  with bus devices, line coupler, line repeater: 120 nF max.
  (measured at 10 KHz)
- Bus line resistance between power supply and bus device, line coupler or line repeater: 25Ω max.
  Bus line resistance between two bus devices, line couplers or line repeaters: 50Ω max.
- Minimum resistance between two power supplies: 15Ω
  Minimum bus line length between two power supplies: 200 m.
- Voltage drop on bus line between power supply and bus device or line coupler: 5 V max.
  This criterion determines how the bus devices may be physically arranged in the bus line of a segment. It is not permitted, for instance, to have a conglomeration of 64 bus devices installed at one end of a 350 m bus line and the power supply at the other.
- Maximum bus line length of a segment: 1000 m
  Maximum bus line length between two devices: 700 m
  Max. bus line length between power supply and bus device: 350 m
- No terminating resistances required.
- The bus devices are supplied with a rated voltage of 24 V DC through the bus.
- Max. number of bus devices of an electrical segment: 64.

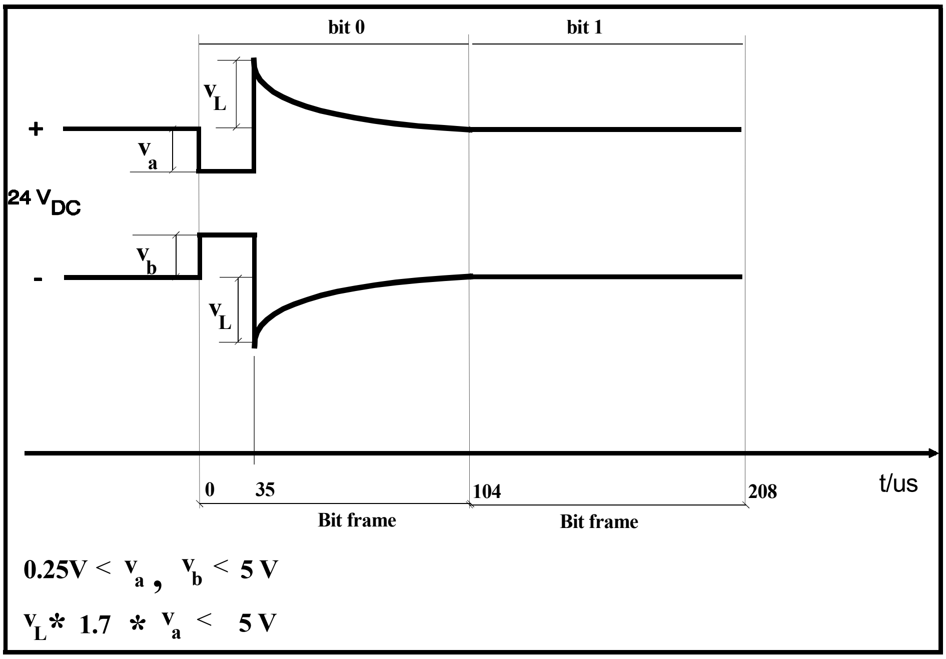
Technical data of the bus line:

Type	Twisted Pair, two pairs
Load resistance per line	max . 37 Ω/km (Loop 74 Ω/km)
Load capacitance line / line	max 100 nF/km (800 Hz)
Screening	Thin strip with drain terminal
Number of twists	min. 5/m
Line diameter	0.8 mm

TP cable which meets the KNX requirements in Volume 9 of the KNX specifications (e.g. YCYM 2x2x0.8 or J-Y(St)Y 2x2x0.8) can be authorised (without the KNX logo) or certified (with the KNX logo) by KNX Association. Only the green standard TP cable guarantees the cable lengths above. The maximum length of all other cables per line segment is indicated in the data sheet of the respective cable.
If distribution boards are used, these may house a data rail in addition to the bus line.

Transmission method used:
Time division multiplexing, baseband, symmetrical

Bit representation:



Absence of DC current, not self-timing.

Transmission speed:
9.6 Kbit/s

Transmission delay:
To ensure collision detection and correction, the transmission delay must not exceed 10 µs. This condition is automatically met if you

- use only the authorised bus cables
- observe the maximum distance of 700 metres between any two bus devices and
- do not install more than 64 bus devices in one electrical segment.

Bus coupling method:
Repeater, symmetrical

Connection techniques used for cables:
Bus terminal (terminal block) with four terminals per line for connection to the bus device.
Overvoltage protection terminal in connection with the bus terminal.
Data rail-to-wire connector for connecting the bus line to the data rail.

Connection technique used for data rails:
Bus devices designed for DIN rail mounting have a contact block containing spring-finger connectors which establish the connection to the conductors of the data rail.

3 KNX Telegram structure

As already indicated above, each layer will contribute to a part of the telegram when a device generates a telegram and will decode part of the telegram when receiving.
Or in other words:

- When receiving a telegram, it will process the relevant information and pass on the rest to the upper layers;
- When creating a telegram, it will insert its relevant information before it passes on the data to the lower layers.

As already indicated, the general structure of a telegram is as follows:

<table>
<tr><td>Control field</td><td>Source address</td><td>Receiver address</td><td colspan="3">N_PDU</td><td>Check field</td></tr>
<tr><td>8 bit</td><td>16 bit</td><td>16 bit</td><td>8 bits</td><td colspan="2">T_PDU</td><td>8 bit</td></tr>
<tr><td></td><td></td><td></td><td></td><td>6 bits</td><td>A_PDU</td><td></td></tr>
</table>

The N-PDU, as the part of the LL telegram destined for/created by the Network Layer, contains the following information:

- The **first bit** indicates whether the receiver address needs to be interpreted as a
  - Individual addressed telegram (value 0)
  - Group addressed telegram (value 1)