Coupler device and method for the interconnection of KNX networks through an IP-network

Coupler and method device for the interconnection of a local KNX network with a remote KNX network through an IP-network, the interconnection being able to be initiated by said device, the remote KNX network being provided with another coupler device for interconnecting said remote KNX network with said IP-network, wherein the device comprises: a KNX interface for the KNX communications with the local KNX network; an IP interface for the IP communications through the IP-network; a data processing module configured to carry out program instructions for implementing a method which comprises: sending, through the IP interface, the ID of the remote coupler device to an identification server which is connected to the IP-network; requesting and receiving from the identification server an IP-network location of the remote coupler device; establishing a secured IP connection, through the IP interface, to the received IP-network location of the remote coupler device; interconnecting the KNX interface with the remote coupler device through the secured IP connection.
DESCRIPTION

COUPLER DEVICE AND METHOD FOR THE INTERCONNECTION OF KNX NETWORKS THROUGH AN IP-NETWORK

Technical field

[0001] The present disclosure relates to a device and method for KNX internetworking, in particular to a device and method for establishing secure and full connections between two or more KNX devices, especially between multisite networks.

Background

[0002] KNX is a worldwide standard, with more than 25 years. It is independent of manufacturers and applications systems. This standard was created to regulate the implementation of home and building automation solutions, providing the coexistence of devices from different manufacturers in the same installation.

[0003] KNX devices are connected via the KNX local bus used for the exchange of the operating information.

[0004] The KNX defines four supported physical mediums: TP - twisted pair; RF - radio frequency; PL - power line; IP - over Ethernet or Wi-Fi.

[0005] Said devices may be sensors or actuators that can be used to control building management devices such as (but not limited to): lighting, shading/shutters, security or alarm systems, energy management, heating, ventilation and air conditioning systems, signaling and monitoring systems, service interface and building control systems, remote control, metering, audio/video control.

[0006] All these functions through a unified system can be controlled, monitored and signaled with no additional centralized control center.

[0007] When the KNX standard was established, the need for multisite extensions, as well as security concerns were not known or at least so critical, as the protocol was effective prior to massive use of the Internet.
Up until this moment, there are some solutions, which are deemed to be merely workarounds to the above-mentioned technical problem.

The KNX standard was conceived for local use, inside a given building. Its extension to use over global networks (e.g. Internet) implies the encapsulation over TCP/IP (situation already provided by the KNX standard) and the use of a VPN (Virtual Private Network) as ways to provide and enhance security. The implementation of such a system is not reachable for a common electrician or to a KNX integrator.

Some known solutions comprise blocking the KNX inside the building and creating a gateway that converts KNX to proprietary protocols, that implement a subset of functions, and allow the remote control of the system. This is not really extending the KNX network, as new devices, new function and new DPT (data point types) will be outside of scope and will impose a revision of the proprietary protocols. In this we can include http and https web servers used to access the local KNX network, this methodology will be include in KNX standard in a near future.

It also possible to simply use a KNX/IP Gateway and port forwarding to provide access, however in this case without proper communication security to the local KNX network. This is the worst but still most used workaround for the identified problem.

The document CN102035842 discloses a solution to guarantee security inside a local KNX bus. However, CN102035842 is limited to ensuring local KNX security.

These facts are disclosed in order to illustrate the technical problem addressed by the present disclosure.

**General Description**

The claimed disclosure allows:

- The interconnection of two (or more) KNX networks in distinct physical sites;
- A remote and secure access to KNX networks, through the Internet, using KNX standard telegrams, end to end.

The disclosure comprises the following elements: at least two identical devices 2, one on each end of the connection and a secure communication channel 3 between these
devices. This system will provide the necessary secure interconnection of KNX local networks 1 of both sites 4 5, as shown on Fig. 1.

[0016] A usual KNX system is composed of several connected lines and areas, respectively by main lines and backbone lines. Line and Area Couplers are commonly used to establish these connections. These devices implement two or more interfaces that physically connect to each line/area creating its interconnection.

[0017] The disclosure allows some of these lines/areas to be located in different and remote located sites (buildings), being a distributed device where the same two or more interfaces are split between the different sites (see fig. 1).

[0018] The connection between said interfaces is a virtual connection, using security techniques. With this approach the telegrams of any of the KNX networks are forwarded from one side to another in a configurable manner. As in other Line/Area Couplers the information that transverses the KNX Distributed Virtual Coupler can still be filtered through switching tables which in the present disclosure may also be remotely established on-demand. The approach assures the compatibility with actual and upcoming KNX standards and also allows a secure reachable bidirectional connection over virtually any unsecure network such as the web cloud, while at the same time avoiding the issues in prior art broadcasting-based solutions.

[0019] One main advantage for the users (integrators and electricians) is the easy setup and automatic assured connection without requiring any security or computer skills.

[0020] It is disclosed a coupler device for the interconnection of a local KNX network with a remote KNX network through an IP-network, herewith device, the interconnection being able to be originated by said device,

the remote KNX network being provided with another coupler device for interconnecting said remote KNX network through said IP-network, herewith remote coupler device,

wherein the device comprises:
a KNX interface for the KNX communications with the local KNX network;
an IP interface for IP communications through the IP-network;
a data processing module configured to carry out program instructions;
the data processing module being configured to carry out:
sending, through the IP interface, the ID of the remote coupler device to an identification server which is connected to the IP-network;
requesting and receiving from the identification server an IP-network location of the remote coupler device;
originating a secured IP connection, through the IP interface, to the received IP-network location of the remote coupler device;
interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

[0021] In an embodiment, the interconnection being also able to be originated by the remote coupler device, the data processing module being further configured to carry out:
registering with the identification server by sending to the identification server the IP-network location of the device;
accepting a secured IP connection, whether directly or indirectly, through the IP interface, originating from the remote coupler device;
interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

[0022] It is disclosed a coupler device for interconnecting on-demand a local KNX network with a remote KNX network through an IP-network, herewith device,
the remote KNX network being provided with another coupler device for interconnecting said remote KNX network through said IP-network, herewith remote coupler device, the interconnection being able to be originated by the remote coupler device,
wherein the device comprises:
a KNX interface for the KNX communications with the local KNX network;
an IP interface for IP communications through the IP-network;
a data processing module configured to carry out program instructions;
the data processing module being configured to carry out:
registering with the identification server by sending to the identification server the IP-network location of the device;
accepting a secured IP connection, whether directly or indirectly, through the IP interface, originating from the remote coupler device;
interconnecting the KNX interface with the remote coupler device through the
secured IP connection, for transmitting and/or receiving KNX telegrams.

[0023] In an embodiment, the interconnection being able to be initiated by the remote
coupler device, the data processing module being further configured to carry out:
previously to registering with the identification server;
establishing a secured IP connection, herewith channel, through the IP interface to a
forwarding server which is connected to the IP-network and is reachable from the IP-
network;
receiving from the forwarding server an IP-network location, such that communications received by said forwarding server on that IP-network location are forwarded to the device through said channel;
using the IP-network location received from the forwarding server as the IP-network
location, instead of the IP-network location of the device, for subsequently
registering with the identification server.

[0024] In an embodiment, the interconnection being able to be initiated by the remote
coupler device, the data processing module being further configured to carry out:
verifying if the device is reachable from the IP-network;
if the device is reachable, not proceeding with establishing said channel with the
forwarding server and registering with the identification server with the IP-network
location of the device;
if the device is not reachable, proceeding with establishing said channel with the
forwarding server and registering with the identification server with the IP-network
location received from the forwarding server;
wherein being reachable from the IP-network means being reachable by connections
which have been originated from the IP-network.

[0025] In an embodiment, the interconnection being able to be initiated by said device,
the data processing module being further configured to carry out:
if the secured IP connection is not established or lost, and remains not established
after a predetermined number of retries or a predetermined time interval,
requesting and receiving from the identification server a new IP-network location of
the remote coupler device;
if the newly received IP-network location has changed from the previously received IP-network location, re-establishing the secured IP connection, through the IP interface, to the newly received IP-network location of the remote coupler device.

[0026] In an embodiment, the interconnection being able to be initiated by said device, the data processing module being further configured to carry out:

after establishing the secured IP connection and before interconnecting the KNX interface with the remote coupler device, sending through the secured IP connection, the ID of the remote coupler device.

[0027] In an embodiment, the interconnection being able to be initiated by the remote coupler device, the data processing module being further configured to carry out:

after interconnecting the KNX interface with the remote coupler device,
when a KNX filter is received from the remote coupler device,
applying that KNX filter to the sending and/or receiving of KNX telegrams such that only KNX telegrams complying with said KNX filter are sent and/or received to the remote coupler device.

[0028] In an embodiment, the IP-network location is an IP address-port number duplet.

[0029] In an embodiment, the secured IP connection is a secured TCP/IP connection, a VPN connection, a SSL connection, a TLS connection, SSH connection, private/public key cryptographic connection, or a symmetric cryptographic connection.

[0030] In an embodiment, the interconnection being able to be initiated by said device, the data processing module is further configured to, when receiving an IP-network location from the identification server, to further receive system data or instructions from the identification server, in particular power status data, power instruction data, firmware update status data, firmware instruction data, accessibility and/or visibility status data of the device for other remote devices, or combinations thereof.

[0031] In an embodiment, the IP-network communications comprise a service code identifier for distinguishing between different sets of communications in operation, whether partly sharing or fully sharing a set of coupler devices or for distinguishing between KNX telegrams and KNX filters.

[0032] In an embodiment, the device comprises three networking layers: a KNX stack layer for the KNX communications with the local KNX network, a switch layer for packet
switching and a secure connection layer for establishing and securing the IP communications through the IP network.

[0033] In an embodiment, the KNX stack layer is implemented by a KNX Bus Interface Module, BIM, a KNX Bus Coupling Unit, BCU, or a KNX Bus Access Unit, BAU.

[0034] In an embodiment, the KNX interface is a KNX twisted pair interface; a KNX radio frequency interface; a KNX power line interface; or a KNX IP-over Ethernet or Wi-Fi interface.

[0035] In an embodiment, the device is configured to deploy a IGD, Internet Gateway Device, subset of UPnP (Universal Plug and Play) for opening local router port forwarding from the IP network to the device.

[0036] In an embodiment, the coupler device is a virtual device.

[0037] It is also disclosed a method of operating a coupler device for the interconnection of a local KNX network with a remote KNX network through an IP-network, according to any of the mentioned embodiments, comprising the method steps referred to be carried out by said data processing module.

[0038] It is also disclosed a non-transitory storage media including program instructions for implementing a coupler device for the interconnection of a local KNX network with a remote KNX network through an IP-network, the program instructions including instructions executable to carry out any of the mentioned methods of operating a coupler device.

**Brief Description of the Drawings**

[0039] The following figures provide preferred embodiments for illustrating the description and should not be seen as limiting the scope of invention.

[0040] **Figure 1:** Schematic representation of an embodiment of secure communication channel between two devices, providing secure interconnection of KNX local networks according to the disclosure.

[0041] **Figure 2:** Schematic representation of an embodiment of devices according to the disclosure, having a secure communication channel, composed of three layers: KNX stack; switching system; and secure connection implementation.
Figure 3: Schematic representation of an embodiment of a secure communication channel between two devices for providing secure interconnection of KNX local networks, involving the use of a public external server for Virtual Private Port Forwarding and a public server for directory identification, according to the disclosure.

Figure 4: Schematic representation of an embodiment of a switching system according to the disclosure.

Figure 5: Schematic representation of an embodiment of the communication protocol between two devices, installed in different sites when a direct connection is possible, according to the disclosure.

Figure 6: Schematic representation of an embodiment of the communication protocol, according to the disclosure, between two devices, installed in different sites when forwarding (with a previously established connection, i.e. port opening) is needed to reach the device behind a firewall.

Figure 7: Schematic representation of an embodiment of the filtering protocol, according to the disclosure, between two devices, installed in different sites.

Figure 8: Schematic representation of an embodiment of the fault tolerance mechanism according to the disclosure.

Figure 9: Schematic representation of an embodiment a worldwide KNX infrastructure, which connects through a WAN IP network, using the KNX Distributed Virtual Coupler 2 in five exemplary different roles.

**Detailed Description**

[0049] The disclosure comprises the use of identical devices in both (or many) sites. Each device is composed of three layers (fig. 2): a first layer 8, that implements the KNX stack; a second layer 7 that implements the switching system; and a third layer 6, which implements the secure connection.

[0050] In the first layer, to implement the KNX stack a standard Bus Interface Module (BIM), a standard Bus Coupling Unit (BCU) or a standard Bus Access Unit (BAU) is used for Twisted Pair, Power Line or Radio Frequency, physical and data link layers. When the KNX/IP connection is required, the implementation can either be done using an
embedded system or even a software application in a normal computer with Ethernet or Wi-Fi connection.

[0051] The third layer is implemented in three steps: the first step is a handshake between the two endpoints, allowing them to identify properly between them; the second step is the negotiation of a secure protocol for that connection, namely choosing a specific security algorithm to assure security in the communication. Then a direct forward of the KNX telegrams is implemented. This implementation of the established secure channel is agnostic to any specific security protocol, as the state of the art in this particular point is always evolving. This approach will guarantee that this solution will persist and be feasible in the future.

[0052] The module 6 that implements the secure connection handles three operation modes:

  Server mode - that runs when direct connections to the device exist 2. This mode implements the support to multiple connections 3 from the Wide Area Network (WAN). Each of these connections will represent an independent session (SI, S2, ..., Sn) such that these connections may be managed individually (for example, setting an individual filter for each connection).

  Locked mode - that runs when the device 2 is behind a firewall, which cannot be reached from the WAN. In this case a persistent connection is established form the Local Area Network (LAN) to a public external server (10 at Fig. 3) for Virtual Port Forwarding (SVPF), or shortly, forwarding server. This approach creates a unique channel from the device 2 to the SVPF, this channel implementing asynchronous support for multiple sessions (SI, S2, ..., Sn). The multiplexing of the multiple connections is preferably implemented on the forwarding external public server 10 in a sandboxed way, which provides exclusive access to each device 2. Having sessions allows that multiple connections may be transported in the same channel and still their traffic can be told apart. Also, as before, having sessions allows that these connections may be managed individually (for example, setting an individual filter for each connection).

  Client mode - this mode is used on the device 2 that initiates the connection. The secure connection represents a point-to-point client/server approach that is always started by a client request. In each connection in the level 6, one of the devices
operates on Client Mode and the other is on either Server Mode or Locked Mode.
This Client Mode only implements one session SI.

[0053] Between layers 6 and 8 the communication is just a direct "read or write" process in an embodiment, either using a simple function call (when in the same hardware) or through a serial connection (when in different hardware parts), without any particular security concerns, as both layers are located in the same device. This configures a switching system 7 that is detailed in fig. 4. This switching system interconnects multiple sessions (SI, S2, ..., Sn) implementing a many to many forwarding of all the packets traffic. Each session, in an embodiment, implements a two-way filter (inbound and outbound) that provides the needed decision of dropping or processing said packet. This approach allows the coexistence of multiple instances of the first layer 8, connecting in the local site 4 to different KNX lines/areas (like local Line/Area Couplers), in conjunction to the multiple connections from the WAN - these will represent the SI, ..., Sn switched sessions.

[0054] In an embodiment, the outbound filter of a coupler device may be set remotely, effectively consisting in an on-demand or on-request connection.

[0055] The switched system of the present disclosure overcomes the inconvenient of the multicast system over IP of the KNX standard, adding security and individual filtering capabilities, while avoiding unnecessary data traffic.

[0056] The second layer 7 interconnects multiple sessions with each one representing either:

One of the multiple clients of the Server Mode, from layer 6, or alternatively

One of the sessions of the Locked Mode, from layer 6;

One connection in Client Mode, from layer 6;

One or more KNX stacks 8 for connecting with one or more KNX physical interfaces.

[0057] When the interface 1 is connected to a KNX/IP network, the first layer 8 further implements standard KNX/IP server and client versions, to allow it to work as a normal distributed KNX Distributed Virtual Gateway, allowing the secure remote connection to a KNX local network in a different site.

[0058] In the Internet there are several firewalls and obstacles. To overcome these issues within this solution, a IGD (Internet Gateway Device) subset of UPnP (Universal Plug and
Play) is used to act on the router 9 for port forwarding when possible, or a public server 10 for Virtual Private Port Forwarding (SVPF), or shortly, forwarding server, to ensure the interconnection between the devices (two or more). When needed, the connections will start from the inside network (LAN), as a way to also guarantee the unconditional reach of any device (see fig. 3).

[0059] Another public server 11 for identification (SID), or shortly put as identification server, implements a directory for registration of all the existing KNX Distributed Virtual Coupler devices 2, to guarantee that all the devices can be reached when needed or requested. The SID implements a local persistent table of equivalences between the unique device identifier (ID) and the way to reach it in the WAN. This server also implements a group of generic actions that provide a way to notify the devices, with the state of their visibility on the WAN and other secondary actions needed.

[0060] When one device 2 needs to connect to another device 2, the first device inquires the SID for the location of the second device, and then the first device connects to the address and port received from the SID. The client device does not even know if the connection is direct or through a public forwarding server 10. This approach allows on-demand connections, with known benefits, over permanent ones, for example a smaller bandwidth use and a lower exposition to unauthorized access to the information by third parties (e.g. hackers).

[0061] The KNX Distributed Virtual Coupler can be implemented as a standalone hardware device or as a virtual device that is embodied in software that may be able to run on any computer or an embedded system. It can be a product by itself or be incorporated in any other products that intend to have the ability to connect, using KNX standard, to a remote site where a KNX network exists.

[0062] The disclosure also proposes that the KNX telegrams (messages) may travel through the system under the following flow: the KNX telegram enters the system from the local KNX bus at site (4 or 5, as a bidirectional stream), then goes up through the KNX stack and is sent through the secure channel. At this stage it leaves the local site (4 or 5) and travels through the web. When the telegram arrives at the remote site (5 or 4) on the other end of the secure channel, it goes down through the KNX stack and it is finally delivered to the KNX bus on the remote site (5 or 4).
The disclosure implements the communication between two devices, installed in different sites 4 and 5, using the protocol for example explained by Fig. 5 when a direct connection is possible and using the protocol for example explained by Fig. 6 when the SVPF (forwarding with a previously established connection, i.e. port opening) is needed to reach the device behind a firewall. In these diagrams the site 5 acts as the client (the one that initiates the communication) and the site 4 acts as the server. These roles can be switched with no loss of sense to the person skilled in the field.

In a direct connection scenario (Fig. 5), first the coupler device 4 connects to the SID server 11 (packets CPI and CP2), and then sends a register packet PI to the SID server indicating the device 4 unique identifier (ID) and the IP network location of the device 4. This server updates the built-in persistent table with the new location information of device 4 and replies back with an acknowledge packet P2. This packet P2 may be also used to notify the device that some action is needed, for example, that the server has not been reachable and imposing the commutation to locked server mode (see below for locked server mode). When another coupler device 5 needs to connect to the first device 4, the second device first connects to the SID server (packets CPI and CP2), and then sends it a query packet P3, indicating the ID of the device to be located, to the SID server that replies with a response packet P4 providing the known network location of the required device 4. The P3/P4 communication may be retried in case of no-response or may be repeated at a later time (dashed lines). At this point A, the device 5 initiates a direct secure connection CPI to the obtained location of device 4 and waits for the respective acknowledge packet CP2.

When the acknowledge packet is received, the client device 5 optionally sends a target packet P5 to confirm the unique device identifier (ID) that it intends to connect to. When this target packet reaches device 4 the ID is confirmed (point B), the server site drops the communication if there is any problem with the ID. Sending this packet P5 with the target device ID enables this check.

Then the communication is established in a full duplex way with data packets DPI. A fault tolerance mechanism is also implemented as an independent protocol described in Fig. 8.
[0067] Sending the target device ID is also advantageous for having the same procedure as that of the forwarding server procedure (see below). This way, the originating device 5 does not need to know whether the target device 4 is directly reachable or reachable through a forwarding server, because the work flow is the same for both cases.

[0068] According to the present disclosure, DPI packets include KNX Telegrams and KNX filters. The DPI packet flow then allows a first device to define the filtering policy of a second coupler device, in fact creating an on-demand or on-request system, because the filter for the outbound layer of the second device is defined by the first device. It is also relevant to note that this filter may be set either by the client device (the device that initiated the connection) or by the server device (the device that received the connection request initiated by the client device), as the communication can be fully bidirectional.

[0069] In an indirect connection scenario (Fig. 6), first the coupler device 4 opens a secure connection CPI to the SVPF Server 10, or shortly, forwarding server, and waits for the acknowledgement CP2 of that connection. A channel is thus established between the device 4 and the SVPF server 10. The SVPF server provides the device with a network location that is reachable form the IP network such that communications received on that network location will be forwarded by the SVPF server 10 to the device 4. Then the device connects to the SID server 11 (packets CPI and CP2), and then sends a register packet P1 to the SID server including the device ID and network location which was provided by the SVPF server 10. This server updates the built-in persistent table with the new location information of device 4, which is the initially established connection from device 4 to the SVPF, or forwarding, Server 10, and replies back with an acknowledge packet P2. When another device 5 needs to connect to the first device 4 it first connects to the SID server (packets CPI and CP2), and then sends a query packet P3, indicating the ID of the device to be located, to the SID server that replies with response packet P4 providing the known network location of the required device 4. The P3/P4 communication may be retried in case of no-response or may be repeated at a later time (dashed lines). At this point A, the device 5 initiates a direct secure connection CPI to the obtained location of the device (in this case, the middle-located SVPF server) and waits for the respective acknowledge packet CP2.
Alternatively, the SVPF forwarding server 10 may have a fixed network location for communications forwarding and, in this case, the device 4 simply provides to the SID 11 the fixed network location of the specific SVPF it is configured to use.

When the acknowledge packet is received the client device 5 sends a target packet P5 to confirm the unique device identifier (ID) that it intends to connect to. At this point the device 5 can start to send data DPI to the device 4. When the SVPF Server 10 receives the target packet P5 it sends a request session packet P6 to the device 4, point C and waits for its acknowledge packet P7, point B. The P7 communication may be retried in case where the server needs to close de session with SVFP (dashed lines). At point B and if the client was not refused, any received data packets DPI are forwarded by the SPVF server 10 to the device 4 as DP2 packets. If the client was refused, the client connection is dropped. At point C, device 4 either refuses the client by checking the unique device identifier (ID) by sending correspondent acknowledge packet P7 to the SVPF Server or the communication is established in a full duplex way with data packets DP2. DP2 packets complement DPI packets with additional data to identify a session (e.g. a session ID) such that multiple clients can make use of the same channel between SVPF server 10 and the target device 4.

Alternatively, the client device 5 does not need to send a target packet P5 to confirm the unique device identifier (ID) that it intends to connect to. This is a simpler protocol, but sending the target device ID is advantageous because network locations provided by the SVPF forwarding server may be shared by different channels corresponding to different server devices. In fact, if both client and server devices are not reachable from the IP network and are connected through channels to the same SVPF server 10, these channels may in fact share the same location at the SVPF folder because the traffic will be differentiated with the distinct target device IDs and corresponding distinct sessions.

A fault tolerance mechanism if also implemented as an independent protocol described in Fig. 8.

According to the present disclosure, DPI and DP2 packets include KNX Telegrams and KNX filters. The DP1/DP2 packet flow then allows a first device to define the filtering policy of a second coupler device, in fact creating an on-demand or on-request system,
because the filter for the outbound layer of the second device is defined by the first device. It is also relevant to note that this filter may be set either by the client device (the device that initiated the connection) or by the server device (the device that received the connection request initiated by the client device), as the communication can be fully bidirectional.

[0075] The filtering protocol is implemented at the session level (Fig. 7) within the switching system 7. This diagram (Fig. 7) represents the data packet DPI flow between Session i (Si) and Session j (Sj). These sessions can be swapped, with no loss of sense, and even combined, i.e. each layer may have an inbound, an outbound or an inbound and an outbound filter. At Fig. 7a the flow of a generic data packet is represented. When the packet arrives session Si it is forwarded to the outbound filter Fi that evaluates it and return a Boolean. If the answer was True the packet is then forwarded to the switch 7, otherwise it is dropped. Then the switch forwards the packet to the Session Sj that sends it to its own inbound filter Fj for checking. If the answer is True the packet is processed, otherwise it is dropped. Each of the filters implements both outbound and inbound checks to be able to work in any flow direction.

[0076] The filter uses a filtering mask, which preferably implements two independent masks, one that controls the inbound flow and other that controls the outbound flow, but it may implement only inbound or only outbound filter flow masks.

[0077] In filtering flows and in the particular case that the data packet only contains a new filter, for example a new filtering mask, such that the filter is able to update itself, the answer is always False and the update then takes place on the filter (see Fig. 7b and 7c). This mechanism allows a first coupler device to define the filtering policy of a second coupler device, in fact creating an on-demand or on-request system. It is also relevant to note that this filter may be set either by the client device (the device that initiated the connection) or by the server device (the device that received the connection request initiated by the client device), as the communication can be fully bidirectional.

[0078] Filtering will normally select KNX group addresses. Given the advantages of a link according to the present disclosure (namely, a secure, reachable, bidirectional, on-demand filtered connection), it is particularly suited for interconnecting KNX group addresses.
Filtering may also select individual KNX device addresses. However, this is mostly for setting up and configuring individual devices. Even if it could be used to interchange data between specific KNX device, it is normally considered a superior KNX solution to use KNX group addresses.

The disclosure is optionally powered by a complete fault-tolerant implementation that is based on the secure connection over TCP protocol, that implements, by itself, the guarantee that the packets reach the other end and that they are delivered in the correct order. This secure connection is presented in this document, by connection packet CPI and respective acknowledge packet CP2. The two packets (CPI and CP2) are an abstraction of the need packets to establish a secure connect oriented socket, as this system intends to be agnostic to the security technology used (SSL, TLS, private/public keys, symmetric cryptography, etc) therefore any details on these packets is not needed.

On top of the secure connection layer an extra fault-tolerance is implemented (Fig. 8). When a client device needs to start a new connection - point D - it executes several attempts to connect (CPI packet) to the SID Server, if a valid connection is established packet CP2 is received and the process goes on, otherwise the connection is aborted and the device informed. Then the device obtains the device location (using packets P3 and P4) and executes several attempts to connect to device - point E. If the connection is not established, the device informs the SID server that the device is not reachable at the previously received location, using an action packet P3 and receiving a response packet P4. At this point, the external retries are incremented, if a maximum is reached the process aborts and the device informed. If the response P4 indicates a new location (i.e. the location of device has changed since the last time device 5 inquired the SID server about this), the device goes back to point E and restarts the connection attempts to the device, otherwise the device goes back to point D and restart the whole process. When a successful connection is established by the reception of the acknowledge packet CP2 from device to device 5 the communication is established in a full duplex way with data packets DPI. If during the normal communication process the connection fails - point G - the process goes back to point D and the process restarts.
The fault-tolerance described in Fig. 8 is also implemented, similarly, by the disclosure when the public SVPF Server 10 exists in the middle of the communication between device 5 and device 4.

In an embodiment, the packet contents includes:

P1 - register packet is the one used to register or update the SID Server persistent table, with the format: {size | service code | device unique identifier | data}. Optionally, the packet may contain the IP location that should be used for contacting the packet originator. Normally, this information may be easily derived from the actual communications in which P1 is sent (for example, IP address is the IP address of the packet originator and port number is a predetermined port number service).

P2 - acknowledge packet is sent by SID Server in response to a register packet, with the format: {size | service code | error code | data (optional)}. This packet is also used to notify the device that some action is needed, either the registration information is not reachable (imposing the commutation to locked server mode), a new firmware is available for download, etc.

P3 - a client device uses query/action packet, either to obtain a device location or to notify the SID server of existing problem to connect a certain device. The format is: {size | service code | device unique identifier | data (optional)}.

P4 - acknowledge packet sent by SID Server in response to a query/action packet. This packet contains the device location, if it exists on the persistent table, and has the format: {size | service code | error code | data (optional)}, the data including the network location of the device.

P5 - the target packet if send by the client device and has the purpose to indicate the unique identifier of the device that he wants to contact. This packeted has a fixed size and only includes the {device unique identifier}.

P6 - request session packet sent by SVPF, or forwarding, server to the device operating in Locked mode with the format: {service code | session ID | unique identifier of client device}.

P7 - acknowledge packet is the response to P6 with the format: {service code | session ID | error code}.

DPI - represents a data packeted use in the full duplex communication and has the format: {size | service code | KNX Telegram or KNX Filter or other payload data}. 

17
DP2 - represents a data packeted use in the full duplex communication, between the SVPF Server and the device in Locked Mode. In this situation the data packet is complemented with the session information and has the format: \{size | session ID | DPI\}. This session ID is an identifier used between the SPVF forwarding server and a server in locked mode (not reachable from the ID network) in order to allow multiplexing multiple clients in the same channel.

The packeted data can be at least of two types: a KNX telegram or a KNX filter. A KNX telegram is a normal KNX datagram being sent to another network. A KNX filter is a filter definition (e.g. a binary mask filter definition or a query formatted filter definition) setting the filter that the device receiving the filter is to use from then onwards. The distinction between KNX telegram or KNX filter may be made using different service codes, but other approaches are possible, for example using another packet field.

[0084] This disclosure is essentially applicable as (but not limited to):
A virtual cable to remotely connect two (or more) KNX installations on different sites;
A secure remote connection for ETS (Engineering Tools Software) to any KNX installation;
A way to physically interconnect KNX devices that are on different sites, with the guarantee of security over that connection;
A base connection for (but not limited to) mobile applications and web browser applications used to remote control any building where a KNX coupler according to the disclosure is installed;
Virtualizing the control of any KNX infrastructure over the cloud, either as a SaaS (software as a service) or PaaS (Platform as a service).
A development of virtual KNX devices, with any purpose, that are stored and run on the cloud, as this system can be embedded on those devices to connect to the KNX site.

[0085] The disclosure can be used in very different scenarios. To illustrate some of the possible uses, Fig. 9 represents the use of the KNX Distributed Virtual Coupler 2 in five different roles. This diagram illustrates a possible Worldwide KNX infrastructure, which connects through a WAN IP network 17 from a specific owner. This means that all represented devices are connected and belong to the same KNX project. As specified in the KNX standards a KNX topology is divided into lines and areas that are interconnected with Line/Area Couplers 14 and also by the disclosed device 2 that also works as a Line/Area Coupler. The Individual Address X.X.X (as define by KNX standard) where the
first X is the area, the second is the line in that area and the third is the device in that line, is shown for the couplers. Each device 15 represents any type of KNX device (actuator, dimmer, shutter, sensor, etc) and has its own number between 1 and 64 (see KNX standard). Its Individual Address will incorporate the area, line and number to constitute the device Individual Address. This same logic is used to address the couplers (also KNX devices).

[0086] The diagram represents a KNX project with five different areas, numbered 1 to 5:
Areas 1 and 2 - these areas have a main line 16 and two extra lines coupled to it with Line/Area Coupler device 14. One of the devices 2a/2b will be in client mode and the other 2a/2b in server or locked mode.
Area 3 - represents a standalone KNX Distributed Virtual Coupler 2c that connects a LAN 18 to the rest of the project. This device 2c exposes the KNXnet/IP server protocol (see KNX standard) allowing the connection of any IP based KNX device or Engineering Tool Software (ETS), which is the provided (by KNX Association) software to configure any KNX installation.
Area 4 - represents, for example, the connection of a Web Server 13 or another system, which incorporates the disclosed device 2d, allowing that system to interact with all KNX devices of this project.
Area 5 - represents any standalone virtual/embed device 12, which incorporates the disclosed device 2e, allowing that device to interact with all KNX devices of this project.

[0087] In this use case, the disclosed device 2 is always used as an area coupler, coupling all the five represented areas. But technically, by KNX standard, area and line coupler implement the same role, so they could also be used as line couplers in any mixed topology designed by the installer.

[0088] As seen previously, with regard to the filtering protocol, each session can implement its own filter, either in the inbound and outbound flows. The filtering masks can be pushed around (especially in the DPI packet flow) in the system overcoming the multicast limitation and implementing a, many to many filtered, connection between the devices 2. This feature also optimizes the network traffic reducing it to just the needed data flow.
This way, a filter can be defined remotely, for example, when a session is first established by immediately sending a filter mask. The filter can also be redefined dynamically during a session, by sending a filter mask. When a filter is set remotely this may also be seen as an on-demand request from the device sending the new filter mask. Alternatively to a filter mask, a filter may be defined by a filter query definition or other suitable filter definition methods.

The IP connection can be initiated automatically, for example when the coupler device is switched on or when a physical connection is detected at both IP and KNX interfaces.

The IP connection can also be initiated on-demand, for example, when a KNX communication is received by the coupler device and then the corresponding IP connection is established subsequently.

The IP connections may also time out after a predetermined time. However, always-on is best, because the communications are preferably bidirectional and thus both parties in a connection are always ready to receive data from the other party.

As according to KNX standard every KNX device should be configured with ETS, the disclosed device 2 also preferably implements KNX management protocol in order to allow its own configuration with this software.

The term "comprising" whenever used in this document is intended to indicate the presence of stated features, integers, steps, components, but not to preclude the presence or addition of one or more other features, integers, steps, components or groups thereof.

The disclosure should not be seen in any way restricted to the embodiments described and a person with ordinary skill in the art will foresee many possibilities to modifications thereof. The above described embodiments are combinable. The following claims further set out particular embodiments of the disclosure.
C L A I M S

1. Coupler device for the interconnection of a local KNX network with a remote KNX network through an IP-network, herewith device, the interconnection being able to be originated by said device, the remote KNX network being provided with another coupler device for interconnecting said remote KNX network through said IP-network, herewith remote coupler device, wherein the device comprises:
   a KNX interface for the KNX communications with the local KNX network;
   an IP interface for IP communications through the IP-network;
   a data processing module configured to carry out program instructions;
   the data processing module being configured to carry out:
      sending, through the IP interface, the ID of the remote coupler device to an identification server which is connected to the IP-network;
      requesting and receiving from the identification server an IP-network location of the remote coupler device;
      originating a secured IP connection, through the IP interface, to the received IP-network location of the remote coupler device;
      interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

2. Coupler device according to claim 1, the interconnection being also able to be originated by the remote coupler device, the data processing module being further configured to carry out:
   registering with the identification server by sending to the identification server the IP-network location of the device;
   accepting a secured IP connection, whether directly or indirectly, through the IP interface, originating from the remote coupler device;
interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

3. Coupler device for interconnecting on-demand a local KNX network with a remote KNX network through an IP-network, herewith device, the remote KNX network being provided with another coupler device for interconnecting said remote KNX network through said IP-network, herewith remote coupler device, the interconnection being able to be originated by the remote coupler device,

wherein the device comprises:

a KNX interface for the KNX communications with the local KNX network;

an IP interface for IP communications through the IP-network;

a data processing module configured to carry out program instructions;

the data processing module being configured to carry out:

registering with the identification server by sending to the identification server the IP-network location of the device;

accepting a secured IP connection, whether directly or indirectly, through the IP interface, originating from the remote coupler device;

interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

4. Coupler device according to claim 2 or 3, the data processing module being further configured to carry out:

previously to registering with the identification server;

establishing a secured IP connection, herewith channel, through the IP interface to a forwarding server which is connected to the IP-network and is reachable from the IP-network;

receiving from the forwarding server an IP-network location, such that communications received by said forwarding server on that IP-network location are forwarded to the device through said channel;
using the IP-network location received from the forwarding server as the IP-network location, instead of the IP-network location of the device, for subsequently registering with the identification server.

5. Coupler device according to the previous claim, the data processing module being further configured to carry out:
verifying if the device is reachable from the IP-network;
if the device is reachable, not proceeding with establishing said channel with the forwarding server and registering with the identification server with the IP-network location of the device;
if the device is not reachable, proceeding with establishing said channel with the forwarding server and registering with the identification server with the IP-network location received from the forwarding server;
wherein being reachable from the IP-network means being reachable by connections which have been originated from the IP-network.

6. Coupler device according to claim 1 or 2, or according to claim 4 or claim 5 when dependent on claim 2, the data processing module being further configured to carry out:
if the secured IP connection is not established or lost, and remains not established after a predetermined number of retries or a predetermined time interval, requesting and receiving from the identification server a new IP-network location of the remote coupler device;
if the newly received IP-network location has changed from the previously received IP-network location, re-establishing the secured IP connection, through the IP interface, to the newly received IP-network location of the remote coupler device.

7. Coupler device according to claim 1, 2 or 6, or according to claim 4 or claim 5 when dependent on claim 2, the data processing module being further configured to carry out:
after establishing the secured IP connection and before interconnecting the KNX interface with the remote coupler device, sending through the secured IP connection, the ID of the remote coupler device.

8. Coupler device according to any of the previous claims, the data processing module being further configured to carry out:
   after interconnecting the KNX interface with the remote coupler device,
   when a KNX filter is received from the remote coupler device,
   applying that KNX filter to the sending and/or receiving of KNX telegrams such that only KNX telegrams complying with said KNX filter are sent and/or received to the remote coupler device.

9. Coupler device according to any of the previous claims, wherein the IP-network location is an IP address-port number duple.

10. Coupler device according to any of the previous claims, wherein the data processing module is further configured to, when receiving an IP-network location from the identification server, to further receive system data or instructions from the identification server, in particular power status data, power instruction data, firmware update status data, firmware instruction data, accessibility and or visibility of the device for other remote devices, or combinations thereof.

11. Coupler device according to any of the previous claims, wherein the device comprises three networking layers: a KNX stack layer for the KNX communications with the local KNX network, a switch layer for packet switching and a secure connection layer for establishing and securing the IP communications through the IP network.

12. Coupler device according to the previous claim, wherein the KNX stack layer is implemented by a KNX Bus Interface Module, BIM, a KNX Bus Coupling Unit, BCU, or a KNX Bus Access Unit, BAU, and/or the KNX interface is a KNX twisted pair interface; a KNX radio frequency interface; a KNX power line interface; or a KNX IP - over Ethernet or Wi-Fi - interface.
13. Coupler device according to any of the previous claims, wherein the coupler device is a virtual device.

14. Method of operating a coupler device according to any of the claims 1-13, for the interconnection of a local KNX network with a remote KNX network through an IP-network, comprising the method steps referred to be carried out by said data processing module.

15. Non-transitory storage media including program instructions for implementing a coupler device for the interconnection of a local KNX network with a remote KNX network through an IP-network, the program instructions including instructions executable to carry out the method of claim 14.

16. Method of operating a coupler device for the interconnection of a local KNX network with a remote KNX network through an IP-network, herewith device, the interconnection being able to be originated by said device, the remote KNX network being provided with another coupler device for interconnecting said remote KNX network through said IP-network, herewith remote coupler device, wherein the device comprises: a KNX interface for the KNX communications with the local KNX network; an IP interface for IP communications through the IP-network; and a data processing module configured to carry out program instructions; the method comprising the steps of:

sending, through the IP interface, the ID of the remote coupler device to an identification server which is connected to the IP-network;

requesting and receiving from the identification server an IP-network location of the remote coupler device;

originating a secured IP connection, through the IP interface, to the received IP-network location of the remote coupler device;

interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.
17. Method according to claim 16, the interconnection being also able to be originated by the remote coupler device, the method comprising the steps of:
registering with the identification server by sending to the identification server the IP-network location of the device;
accepting a secured IP connection, whether directly or indirectly, through the IP interface, originating from the remote coupler device;
interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

18. Method of operating a coupler device for interconnecting on-demand a local KNX network with a remote KNX network through an IP-network, herewith device, the remote KNX network being provided with another coupler device for interconnecting said remote KNX network through said IP-network, herewith remote coupler device, the interconnection being able to be originated by the remote coupler device,
wherein the device comprises a KNX interface for the KNX communications with the local KNX network; an IP interface for IP communications through the IP-network; and a data processing module configured to carry out program instructions;
the method comprising the steps of:
registering with the identification server by sending to the identification server the IP-network location of the device;
accepting a secured IP connection, whether directly or indirectly, through the IP interface, originating from the remote coupler device;
interconnecting the KNX interface with the remote coupler device through the secured IP connection, for transmitting and/or receiving KNX telegrams.

19. Method according to claim 17 or 18, the method comprising the steps of, previously to registering with the identification server:
establishing a secured IP connection, herewith channel, through the IP interface to a forwarding server which is connected to the IP-network and is reachable from the IP-network;
receiving from the forwarding server an IP-network location, such that communications received by said forwarding server on that IP-network location are forwarded to the device through said channel;
using the IP-network location received from the forwarding server as the IP-network location, instead of the IP-network location of the device, for subsequently registering with the identification server.

20. Method according to the previous claim, the method comprising the steps of:
verifying if the device is reachable from the IP-network;
if the device is reachable, not proceeding with establishing said channel with the forwarding server and registering with the identification server with the IP-network location of the device;
if the device is not reachable, proceeding with establishing said channel with the forwarding server and registering with the identification server with the IP-network location received from the forwarding server;
wherein being reachable from the IP-network means being reachable by connections which have been originated from the IP-network.

21. Method according to claim 16 or 17, or according to claim 19 or claim 20 when dependent on claim 17, the method comprising the steps of:
if the secured IP connection is not established or lost, and remains not established after a predetermined number of retries or a predetermined time interval, requesting and receiving from the identification server a new IP-network location of the remote coupler device;
if the newly received IP-network location has changed from the previously received IP-network location, re-establishing the secured IP connection, through the IP interface, to the newly received IP-network location of the remote coupler device.

22. Method according to claim 16, 17 or 21, or according to claim 19 or claim 20 when dependent on claim 17, the method comprising the steps of:
after establishing the secured IP connection and before interconnecting the KNX interface with the remote coupler device, sending through the secured IP connection, the ID of the remote coupler device.

23. Method according to any of the claims 16-22, the method comprising the steps of:
   after interconnecting the KNX interface with the remote coupler device,
   when a KNX filter is received from the remote coupler device,
   applying that KNX filter to the sending and/or receiving of KNX telegrams such that only KNX telegrams complying with said KNX filter are sent and/or received to the remote coupler device.

24. Method according to any of the claims 16-23, wherein the IP-network location is an IP address-port number duple.

25. Method according to any of the claims 16-24, wherein the method comprising:
   when receiving an IP-network location from the identification server, to further receive system data or instructions from the identification server, in particular power status data, power instruction data, firmware update status data, firmware instruction data, accessibility and or visibility of the device for other remote devices, or combinations thereof.

26. Method according to any of the claims 16-25, wherein the KNX stack layer is implemented by a KNX Bus Interface Module, BIM, a KNX Bus Coupling Unit, BCU, or a KNX Bus Access Unit, BAU, and/or the KNX interface is a KNX twisted pair interface; a KNX radio frequency interface; a KNX power line interface; or a KNX IP - over Ethernet or Wi-Fi - interface.

27. Method according to any of the claims 16-26, wherein the coupler device is a virtual device.
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L12/46
ADD. H04L12/40

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:
  * "A" document defining the general state of the art which is not considered to be of particular relevance
  * "E" earlier application or patent but published on or after the international filing date
  * "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  * "O" document referring to an oral disclosure, use, exhibition or other means
  * "P" document published prior to the international filing date but later than the priority date claimed

Date of the actual completion of the international search: 15 September 2016

Date of mailing of the international search report: 23/09/2016

Name and mailing address of the ISA:
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016

Authorized officer:
Itani, Maged

Form PCT/ISA/210 (second sheet) (April 2005)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>RIZZO G ET AL: &quot;IPv6 mapping to non-IP protocols; IPv6 MAPPING TO NON-IP PROTOCOLS; DRAFT-RIZZO-6L0-6LEGACY-03.TXT, INTERNET ENGINEERING TASK FORCE, IETF; STANDARDWORKINGDRAFT, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH-1205 GENEVA, SWITZERLAND, 28 March 2015 (2015-03-28), pages 1-13, XP015105718, [retrieved on 2015-03-28] sections 2 to 8</td>
<td>A</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>CN 103036901</td>
<td>A 10-04-2013</td>
<td>NONE</td>
</tr>
</tbody>
</table>