(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 August 2017 (24.08.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/140367 A1

(51) **International Patent Classification**:
H04L 12/725 (2013.01) *H04L 12/66* (2006.01)

(21) **International Application Number**:
PCT/EP2016/053459

(22) **International Filing Date**:
18 February 2016 (18.02.2016)

(25) **Filing Language**: English

(26) **Publication Language**: English

(71) **Applicant**: **RENESAS ELECTRONICS CORPORATION** [JP/JP]; Toyosu Foresia, 3-2-24 Toyosu, Koto-ku, Tokyo, 135-0061 (JP).

(71) **Applicants** *(for MG only)*: **RENESAS ELECTRONICS EUROPE GMBH** [DE/DE]; Arcadiastrasse 10, 40472 Duesseldorf (DE). **RENESAS ELECTRONICS EUROPE LIMITED** [GB/GB]; Dukes Meadow, Millboard Road, Bourne End Buckinghamshire SL8 5FH (GB).

(72) **Inventors**: **MARDMOELLER, Christian**; c/o Renesas Electronics Europe GmbH, Arcadiastrasse 10, 40472 Duesseldorf (DE). **KULKARNI, Dnyaneshwar**; c/o Renesas Electronics Europe Limited, Dukes Meadow, Millboard Road, Bourne End Buckinghamshire SL8 5FH (GB). **HOFFLEIT, Thorsten**; c/o Renesas Electronics Europe GmbH, Arcadiastrasse 10, 40472 Duesseldorf (DE).

(74) **Agents**: **PIOTROWICZ, Pawel** et al.; Venner Shipley LLP, Byron House, Cambridge Business Park, Cowley Road, Cambridge Cambridgeshire CB4 0WZ (GB).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: MESSAGE HANDLER

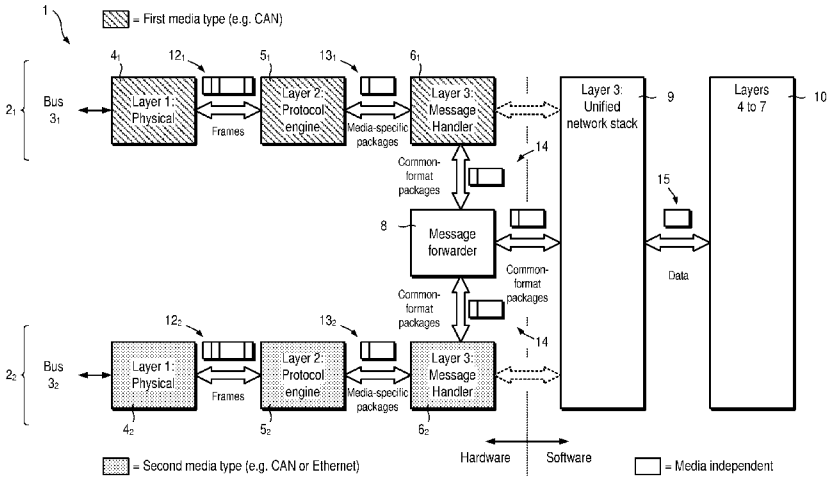


Fig. 2

(57) **Abstract**: A message handler (61, 62) is described. The message handler is configured, in response to receiving a data package (131, 132) which is formatted according to a given communications protocol, such as CAN or Ethernet, and which comprises package- directing data (22; Fig. 4) and payload data (23; Fig. 4), to generate package (14) having a predetermined data format, for example a layer-2 or layer-3 package, which comprises a header (24; Fig. 4) and payload data (25; Fig. 4). The header comprises an address generated in dependence upon the package-directing data and wherein the payload comprises the data package. The package (14) having a predetermined data format may be an IEEE 1722 frame.

WO 2017/140367 A1

Message handler

Field of the Invention

The present invention relates to a message handler and to a control unit comprising at least two message handlers and message forwarder configured to switch or route data packages between message handlers.

Background

A gateway allows information exchange between different network interfaces. The network interfaces can use the same or different OSI physical layer ("layer 1"), data link ("layer 2") and/or network layer ("layer 3") protocol(s), such as Ethernet, controller area network (CAN), FlexRay and media orientated systems transport (MOST). Examples of gateways include Ethernet switches, Digital Subscriber Line Access Multiplexers (DSLAMs) and fieldbus gateways.

Layer 2 gateways within the same protocol, such as Ethernet 10/100/1000 switches and CAN/CAN gateways, tend to be simple because frames in the data link layer are identical, although data rates can change. Layer 2 gateways across protocols, such as CAN/CAN-FD gateways, can provide re-framing without changing a packet (or "package") on the network layer. Layer 3 gateways, such as IPv4/IPv6 gateways or CAN/Ethernet gateways, can re-structure a packet for the different addressing and traffic control requirements. Layer 4 to 7 gateways, such as TCP/UDP gateways and AUTOSAR signal gateways, allow changes between different transport protocols and re-assembling of information.

Layer 2 gateways within the same protocol or protocol family tend to be implemented in hardware and examples of such gateways include the CAN/CAN gateway in Renesas RH850/C1M microcontroller and the CAN/CAN-FD gateway in Renesas RH850/D1M microcontroller.

Referring to Figure 1, layer 3 gateways are often implemented in software. A microcontroller has a set of network interfaces and software executed by a central processing unit of the microcontroller that can re-assemble a packet based on a configuration matrix. Examples of such gateways include a USB/UART converter and an automotive CAN/LIN/FlexRay gateway. Because communication protocols have

different addressing schemes and have different message structures, software provides flexibility to convert between different protocols.

Layer 4 to 7 gateways are generally implemented in software. Re-assembling data in higher layers tends to require significant configurability and flexibility and this can only be practically achieved using a software-based implementation, although hardware-based solutions are possible, for example, in time-critical, conversion-specific cases, such as video data.

Framing of network layer packages is usually carried out in hardware and is transparent for the routing mechanism. Examples in which packet framing occurs include an Ethernet MAC, a FlexRay communication controller and a CAN protocol engine. All protocols tend to have an individual, optimised interface to exchange information with a higher layer.

Layer 3 gateways within same protocol family also tend to be easy to implement in hardware. Only minimal changes in packet structure are required in situations involving, for example, CAN/CAN-FD conversion and IPv4/IPv6 conversion.

Layer 3 gateways involving different protocols tend to be difficult to implement in hardware because of a need to reformat a packet so as to accommodate different addressing, routing and traffic control schemes. Packet conversion generally requires a flexible, configurable data manipulation unit. This can result in a gap in information because not all frame information may be convertible, such as in the case of Ethernet length to CAN DLC.

Addressing scheme conversion needs configurable translation tables between both network layers. Traffic control scheme conversion typically needs either a new layer 4 transport protocol or other high-level support.

Increasingly, communication systems comprise more than one type of bus system which are interconnected. This makes forwarding of data more complex.

US 2014/0133350 A1 describes a gateway module for connecting at least two sub-networks including first and second sub-networks supporting data transmission in accordance with a first and second communications protocols respectively. The gateway

module has a configurable hardware circuit which is adapted to manipulate and forward data from the first sub-network and to the second sub-network taking into consideration the first and second communications protocols. However, as the number and type of communication protocols increases, this type of gateway becomes increasingly complex or requires hardware modification. Thus, as new communication interfaces are introduced, not only must a new interface module be added, but also the gateway module may need modification.

Summary

According to a first aspect of the present invention there is provided a message handler. The message handler is configured, in response to receiving a data package which is formatted according to a given communications protocol and which comprises package-directing data and payload data, to generate a package having a predetermined data format, for example a layer-2 or layer-3 package, which comprises a header and payload data. The header comprises an address generated in dependence upon the package-directing data and the payload comprises the package.

Thus, different message handlers for different communications protocols can be provided in a control unit and be used to generate packages having a common data format. This can help to facilitate exchange of data between two or more communications buses and a host processor, and to enable a hardware-based switch or router to be used for exchanging data between the buses. Thus, the same routing mechanism can be used regardless of the specific communication protocols being used, whether they are the same or different protocols.

The incoming data package may exclude cyclic redundancy check data. The incoming data package may include a timestamp.

The package-directing data may comprise a message identity or a message content identifier (which can be used to identify multicast destinations), or a destination node address. The address may be a stream identifier.

The message handler may be configured to copy data from a first field in the package-directing data into a corresponding field in the header. The message handler may be configured to add predetermined data to a second field in the header. The predetermined data may be supplied by a host processor. The message handler may be configured to add calculated data to the header in dependence upon data from the first field and/or from a second field to a second field in the header.

The message handler may be configured, in response to receiving a package having a predetermined data format which comprises a header and payload data, to extract a data package which is formatted according to a given communications protocol and which comprises package-directing data and payload data.

According to a second aspect of the present invention there is provided a message handler configured, in response to receiving a package having a predetermined data format which comprises a header and payload data, to extract a data package which is formatted according to a given communications protocol and which comprises package-directing data and payload data.

The package having the predetermined data format may be an IEEE 1722 frame. The predetermined data format may comply with IEEE 1722. The package having the predetermined data format may be an AVTP control message. The address may be a stream ID. The package having the predetermined data format may be a layer 2 frame. The package having the predetermined data format may be an IP packet, for example, an IPv4 or IPv6 packet.

The given communications protocol may be ISO 11898-1 (or "CAN 2.0"). The given communications protocol may be controller area network flexible data rate protocol (CAN-FD) protocol. The given communications protocol may be FlexRay protocol. The given communications protocol may be IEEE 802.3 (i.e. Ethernet). The given communications protocol may be media oriented systems transport (MOST) protocol. The given communications protocol may be Ethernet based. The given communications protocol may be EtherCAT, CC-Link IE Field, PROFINET, EtherNet/IP, Modbus TCP or FL-net.

According to a third aspect of the present invention there is provided a network interface module which is hardware-implemented and which comprises a protocol engine and a message handler and which is configured to exchange data packages formatted according to a given communications protocol with the protocol engine.

According to a fourth aspect of the present invention there is provided a central processing unit sub-system comprising a central processing unit, memory and a computer program stored in memory or other storage which, when executed by the central processing unit, causes the central processing unit to execute a message handler.

According to a fifth aspect of the present invention there is provided a message forwarder configured to exchange packages having a predetermined data format with

first and second message handlers for handling messages according to first and second respective communications protocols. The message forwarder may be a layer-2 switch and/or layer-3 router.

According to a sixth aspect of the present invention there is provided a control unit comprising a first message handler or network interface module configured to handle data packages formatted according to a first communications protocol, a second message handler or network interface module configured to handle data packages formatted according to a second communications protocol and a message forwarder configured to exchange packages having the predetermined data format between the first and second message handlers. The message forwarder may be a layer-2 switch or layer 3-router.

The first and second communication protocols may be different. The first and second communication protocols may be the same.

The control unit may further comprise a third message configured to handle data packages formatted according to a third communications protocol (which may be the same or different to the first communications protocol and/or the second communications protocol).

The control unit may further comprise a central processing unit system wherein the message forwarder is configured to exchange between the central processing unit sub-system and the first and/or second message handlers.

The control unit may be an integrated circuit, such as a microcontroller or a system-on-a-chip.

According to a seventh aspect of the present invention there is provided a communications system comprising at least two sets of buses lines and at least one control unit according to the second aspect of the present invention which is (are) connected to the buses. The buses may include buses of different types (such as CAN, CAN-FD and Ethernet).

According to an eighth aspect of the present invention there is provided a vehicle comprising a communications system.

The vehicle may be a motor vehicle. The motor vehicle may be a motorcycle, an automobile (sometimes referred to as a "car"), a minibus, a bus, a truck or lorry. The motor vehicle may be powered by an internal combustion engine and/or one or more electric motors. The vehicle may be a train or a part of a train, such as a locomotive, railway car or multiple unit.

The vehicle may be an aircraft. The communication system may be an Avionics Full-Duplex Switched Ethernet (AFDX) system.

According to a ninth aspect of the present invention there is provided industrial system, such as plant or machinery, comprising a communications system. The plant or machinery may comprise an industrial system for use in manufacturing or processing.

According to a tenth aspect of the present invention there is provided a medical system comprising a communications system.

Brief Description of the Drawings

Certain embodiments of the present invention will now be described, by way of example, with reference to Figures 2 to 8 of the accompanying drawings, in which:

Figure 1 is a schematic block diagram of a communications system in which data can be exchanged between buses in software;

Figure 2 is a schematic block diagram of a communications system in which data can be exchanged between buses in hardware;

Figure 3 is a schematic block diagram of a control unit;

Figure 4 illustrates conversion of a communications protocol-dependent package into a common-format package;

Figure 5 illustrates conversion of a common-format package into a communications protocol-dependent package;

Figure 6 illustrates a CAN frame including a CAN-dependent package;

Figure 7 illustrates an AVTP common message;

Figure 8 is a schematic diagram of a vehicular communications network and a vehicle; and

Figure 9 is a schematic diagram of an industrial communications network and robots.

Detailed Description of Certain Embodiments

Communications system 1

Figure 2 shows a communications system 1 which includes first and second sections 2_1, 2_2 for implementing the lowest protocol layers for first and second communications protocols, such as controller area network (CAN) and Ethernet, which are connected to first and second sets of bus lines 3_1, 3_2.

The first section 2_1 includes a physical layer module 4_1, a protocol engine module 5_1 and a message handler 6_1. The protocol engine module 5_1 and message handler 6_1 are implemented in hardware in a network interface module 7_1 (Figure 3).

The second section 2_2 includes a physical layer module 4_2, a protocol engine module 5_2 and a message handler 6_2. The protocol engine module 5_2 and message handler 6_2 are implemented in hardware in a network interface module 7_2 (Figure 3).

The message handlers 6_1, 6_2 are interconnected by a package or message forwarder 8 which can take the form of a layer-2 switch or a layer-3 router. The message forwarder 8 is preferably implemented in hardware comprising appropriate hardware logic,

hardware registers *etc.* The message forwarder 8 does not carry out any protocol conversion. The message forwarder 8 is also connected, via a unified network stack 9, to higher layers 10. The unified network stack 9 and higher layers 10 are implemented in software.

The physical layer module 4_1, 4_2 passes incoming frames 12_1, 12_2 to the protocol engine 5_1, 5_2 which can remove frame components, such as the frame check sequence (FCS), and generates packages 13_1, 13_2. The protocol engine 5_1, 5_2 may add information, such as a timestamp, to the packages 13_1, 13_2.

The protocol engine 5_1, 5_2 passes the packages 13_1, 13_2 to the message handler 6_1, 6_2 which generates common-format packages 14. As will be explained in more detail later, the message handler 6_1, 6_2 can pass the common-format packages 14 to the message forwarder 8 for switching or routing the common-format packages 14 to another message handler 6_1, 6_2 and/or to the unified network stack 9. Thus, packages can be forwarded not only between message handlers 6_1, 6_2, but also to higher layers 10.

If there are more than two message handlers 6_1, 6_2, then the message forwarder 8 can switch or route packages to more than one message handler 6_1, 6_2, as well as to the higher layers 10, if necessary. The message forwarder 8 may unicast or multicast messages.

Referring also to Figure 3, the physical layer modules 4_1, 4_2 take the form of PHY transceivers 4_1, 4_2. The protocol layers 5_1, 5_2, message handlers 6_1, 6_2, message forwarder 8, unified stack 9 and higher layers 10 are implemented in a control unit 16 in the form of a microcontroller or system-on-a-chip. The control unit 16 comprises network interface modules 7_1, 7_2 (or "communication controllers") which provide the protocol engines 5_1, 5_2 and message handlers 6_1, 6_2. The control unit 16 comprises a central processing unit subsystem 17 which includes at least one central processing unit (CPU) 18, memory 19 and an on-chip interconnect (not shown). The physical layer modules 4_1, 4_2 can also be implemented in the microcontroller or system-on-a-chip.

The CPU 18 is able to configure the network interface modules 7_1, 7_2 and the message forwarder 8. The CPU 18 may also be able to access message handler 6_1, 6_2, bypassing the message forwarder 8, so as to be able to read, write and process frames more directly. Thus, the message forwarding mechanism need not be used for all frames. For

example, message forwarding can be used to route frames automatically for certain, predetermined types of frames. Notwithstanding the fact that the message forwarder 8 can be bypassed, the CPU 18 may still use the common format, although it may also be able to handle data in a specific media format, such as CAN.

The CPU 18 may run a software-based message handler 20. This can allow the CPU 18 to prepare common-format messages 14 which can then be sent either directly, or via the message router 8, to one or more message handlers 6_1, 6_2.

Message handlers 6_1, 6_2

As mentioned earlier, a message handler 6_1, 6_2 receives data packages 13_1, 13_2 from a respective protocol engine 5_1, 5_2, such as a CAN protocol engine, and generates common-format packages 14.

Referring also to Figure 4, each data package 13_1, 13_2 includes a protocol-dependent address 22 and payload 23.

The format of the address 22 and how the address 22 is used depends on the communications protocols. For example, Ethernet uses 48-bit media access control (MAC) addresses to identify destination nodes for unicasting or multicasting packets. CAN uses 11- or 29-bit message identifiers to identify message content for multicasting messages. FlexRay uses a temporary relation to identify message content for multicasting messages.

Priority control can be included in the address, in the payload 23 or signalled in other ways. For example, in Ethernet, priority information is included in the payload, whereas in CAN, message priority is included in the message identity. In FlexRay, message priority is based on repetition in pre-defined cycles.

The message handler 6_1, 6_2 extracts the protocol-dependent address 22 from a data package 13_1, 13_2 and prepares a common-format header 24 and payload 25. The common-format header 24 includes a common-format address 26.

The message handler 6_1, 6_2 maps the protocol-dependent address 22 into a common-format address 26. Preferably, address and message identification are split into different fields. This can help to simplify routing of the common-format package 14.

The message handler 6_1, 6_2 encapsulates the data package 13_1, 13_2 received from the protocol engine 5_1, 5_2 into the payload 25 of the common-format message 14. Thus, the common-format message 14 can carry information for all supported data link layers, such as CAN message ID as well as a common-format address. If the common-format message 14 uses a layer-2 or layer-3 protocol which does not allow tunnelling of all fields (such as baud rate bit of CAN) such that it results in missing information, then the message handler 6_1, 6_2 may be configured to fill-in the missing information, for example, configured by software running on the CPU 18.

Referring also to Figure 5, when a message handler 6_1, 6_2 receives a common-format package 14 from the message forwarder 10, it discards the header 26 and extracts the data package 13_1, 13_2 from the payload 25. The message handler 7_1, 7_2 then forwards the package 13_1, 13_2 to the protocol engine 5_1, 5_2.

As explained earlier, the message forwarder 8 may forward a common-format package 14 to more than one target. The one or more targets may include more than one message handler 6_1, 6_2. The one or more targets may include the CPU 18.

The common-format package 14 provides a common addressing format which allow identification of a target node (not shown) outside the control unit 16 (Figure 3) or a target service (not shown) running on the CPU 18. The common addressing format preferably also provides a quality of service (QoS) level. The common addressing format provides a minimum set of information which allows switching or routing to be performed.

The common-format package 14 provides a container which allows a communication protocol, such as Ethernet, CAN or FlexRay, to map its message back into its protocol-specific message format.

The common-format package 14 can be any suitable type of layer-2 or layer-3 data container. Preferably, the common-format package 14 take the form of Audio Video Transport Protocol (AVTP) Control Format (ACF) messages defined according to IEEE 1722 using the 64-bit stream ID as the common-format address 26 and 8-level QoS according to IEEE 802.1Q for signalling priority levels. This allows a layer-2 Ethernet switch to be used a message forwarder 8.

ACF messages using stream ID as the common-format address need not be used. Other types of data container can be used, such as a layer-3 datagram (i.e. an "IP packet" or simply "packet") using the IP address as the common-format address (e.g. based on IPv4 or IPv6) or a layer-2 datagram (i.e. "frame") which uses the MAC address as the common-format address.

Figure 6 shows a data package 13_1, taken from a CAN data frame, which is passed from the protocol engine 5_1 to the message handler 6_1.

Figure 7 shows how the data package 13_1 shown in Figure 6 is converted into a common-format package 14.

As shown in Figure 6, data in some of the fields, such as RTR, IDE and DATA (i.e. payload data), are obtained by copying data from the data package 13_1. Data are added to other fields according to pre-defined schemes which may be based on acceptance filter lists (AFLs) or look-up tables. Data are added to other fields dependent on data found in fields in the data package 13_1. Although the message handler 6_1 is implemented in hardware, it can be configured by the CPU 18.

The IEEE 802.1 and 802.1Q headers are not required, but are helpful because IEEE 1722 messages are typically encapsulated in Ethernet packages.

Referring to Figure 8, a vehicular communications network 31 is shown. The network 31 includes a plurality of different busses 3_1, 3_2, 3_3, such as CAN and Ethernet. The network includes a plurality of control units 32 which may take the form of microcontrollers connected to the busses 3_1, 3_2, 3_3. At least some of the control units 32 are control units 16 which include network interface modules 7_1, 7_2 (Figure 3) and message forwarder 8. The vehicular communications network 31 is deployed in a vehicle 33.

Referring to Figure 9, an industrial communications network 41 is shown. The network 41 includes a plurality of different busses 3_1, 3_2, 3_3, such as CAN and Ethernet. The network includes a plurality of control units 42 which may take the form of microcontrollers connected to the busses 3_1, 3_2, 3_3. At least some of the control units 42 are control units 16 which include network interface modules 7_1, 7_2 (Figure 3) and

message forwarder 8. The industrial communications network 41 is deployed in a plant, machinery or system 43, such as a manufacturing or processing system.

The message handlers 6_1, 6_2 can have one or more benefits.

The message handlers 6_1, 6_2 can help to provide uniformity in the way messages are forwarded regardless of bus type and whether the message arrived on a communications bus or is generated by the CPU. It can also provide configurable flexibility, for example, allowing messages to be forwarded to multiple targets. Furthermore, the use of a common network stack reduces the degree of layer-3 software stack adaption when adding a new protocol. Moreover, a common set of functions are available, regardless of protocol, such as an abstract QoS mechanism (for example, mapped to IDs on CAN, to 802.1Q PCP on Ethernet and to scheduling on FlexRay), an abstract node addressing mechanism (for example mapped to IDs on CAN, to MAC, VLAN, AVTP or IPv4 on Ethernet) and a unified timestamping mechanism (for example, based on 802.1AS). Additionally, it can simplify software maintenance when changing protocol (for example, porting a CAN-based control unit to Ethernet).

Modifications

It will be appreciated that many modifications may be made to the embodiments hereinbefore described.

The control units may include more than two network interface modules. Two or more of the network interface modules may be of the same type, for example, CAN controllers. Thus, the control unit may be connected to more than two types of bus. The message forwarder may be connected to more than two network interface modules. Thus, the control unit may be connected to more than two buses of the same type. The message forwarder may be connected to more than two network interface modules. Thus, the message forwarder may switch or route common-format messages between three or more message handlers.

Claims

1. A message handler (6_1, 6_2; 20) configured, in response to receiving a data package (13_1, 13_2) which is formatted according to a given communications protocol and which comprises package-directing data (22) and payload data (23), to generate a package (14) having a predetermined data format which comprises a header (24) and payload data (25), wherein the header comprises an address generated in dependence upon the package-directing data and wherein the payload comprises the data package.

2. A message handler according to claim 1, wherein the package (14) having the predetermined data format is a layer-2 or layer-3 package.

3. A message handler according to claim 1 or 2, which is configured to copy data from a first field in the package-directing data into a corresponding field in the header.

4. A message handler according to any one of claims 1 to 3, which is configured to add predetermined data to a second field in the header.

5. A message handler according to any one of claims 1 to 4, which is configured to add calculated data to the header in dependence upon data from the first field and/or from a second field to a second field in the header.

6. A message handler according to any one of claims 1 to 5, which is configured, in response to receiving a package (14) having a predetermined data format which comprises a header (24) and payload data (25), to extract a data package (13_1, 13_2) which is formatted according to a given communications protocol and which comprises package-directing data (22) and payload data (23).

7. A message handler (6_1, 6_2, 20) configured, in response to receiving a package (14) having a predetermined data format which comprises a header (24) and payload data (25), to extract a data package (13_1, 13_2) which is formatted according to a given communications protocol and which comprises package-directing data (22) and payload data (23).

8. A message handler according to any one of claims 1 to 7, wherein the package (14) having the predetermined data format is an IEEE 1722 frame.

9. A message handler according to any one of claims 1 to 7, wherein the package (14) having the predetermined data format is a layer 2 frame.

10. A message handler according to any one of claims 1 to 7, wherein the package (14) having the predetermined data format is an IP packet.

11. A message handler according to any one of claims 1 to 10, wherein the given communications protocol is ISO 11898-1.

12. A message handler according to any one of claims 1 to 10, wherein the given communications protocol is the controller area network flexible data rate protocol.

13. A message handler according to any one of claims 1 to 10, wherein the given communications protocol is the FlexRay protocol.

14. A message handler according to any one of claims 1 to 10, wherein the given communications protocol is the media oriented systems transport protocol.

15. A message handler according to any one of claims 1 to 10, wherein the given communications protocol is Ethernet based.

16. A message handler according to any one of claims 1 to 10, wherein the given communications protocol is EtherCAT, CC-Link IE Field, PROFINET, EtherNet/IP, Modbus TCP or FL-net.

17. A network interface module (7_1, 7_2) which is hardware-implemented and which comprises:

a protocol engine (5_1, 5_2); and

a message handler (6_1, 6_2) according to any one of claims 1 to 16 and which is configured to exchange data packages (13_1, 13_2) formatted according to a given communications protocol with the protocol engine.

18. A central processing unit sub-system comprising:

a central processing unit (18);

memory (19);

a computer program stored in memory or other storage which, when executed by the central processing unit, causes the central processing unit to execute a message handler according to any one of claims 1 to 16.

19. A message forwarder (8) configured to exchange packages having a predetermined data format with first and second message handlers (6_1, 6_2) for handling messages according to first and second respective communications protocols.

20. A control unit (16) comprising:

a first message handler (6_1) according to any one of claims 1 to 16 or a network interface module (7_1) according to claim 17 configured to handle data packages (13_1, 13_2) formatted according to a first communications protocol;

a second message handler (6_2) any one of claims 1 to 16 or a network interface module (7_2) according to claim 17 configured to handle data packages (13_1, 13_2) formatted according to a second communications protocol; and

a message forwarder (8) configured to exchange packages having the predetermined data format between the first and second message handlers.

21. A control unit according to claim 20, wherein the message forwarder is a layer-2 switch and/or a layer 3-router.

22. A control unit according to claim 20 or 21, wherein the first and second communication protocols are different.

23. A control unit according to claim 20, 21, or 22, further comprising:

a third message handler according to any one of claims 1 to 16, configured to handle data packages (13_1, 13_2) formatted according to a third communications protocol.

24. A control unit according to any one of claims 20 to 23, further comprising:

a central processing unit sub-system (17); and

wherein the message forwarder is configured to exchange between the central processing unit sub-system and the first and/or second message handlers.

25. A control unit according to any one of claims 20 to 24 which is an integrated circuit.

26. A control unit according to any one of claims 20 to 25 which is a microcontroller.

27. A communications system (31) comprising:

at least two sets of buses lines (3_1, 3_2); and

at least one control unit (16) according to any one of claims 20 to 26 connected to the buses.

28. A vehicle (32) comprising a communications system (31) according to claim 27.

29. An industrial system (33) comprising a communications system (31) according to claim 27.

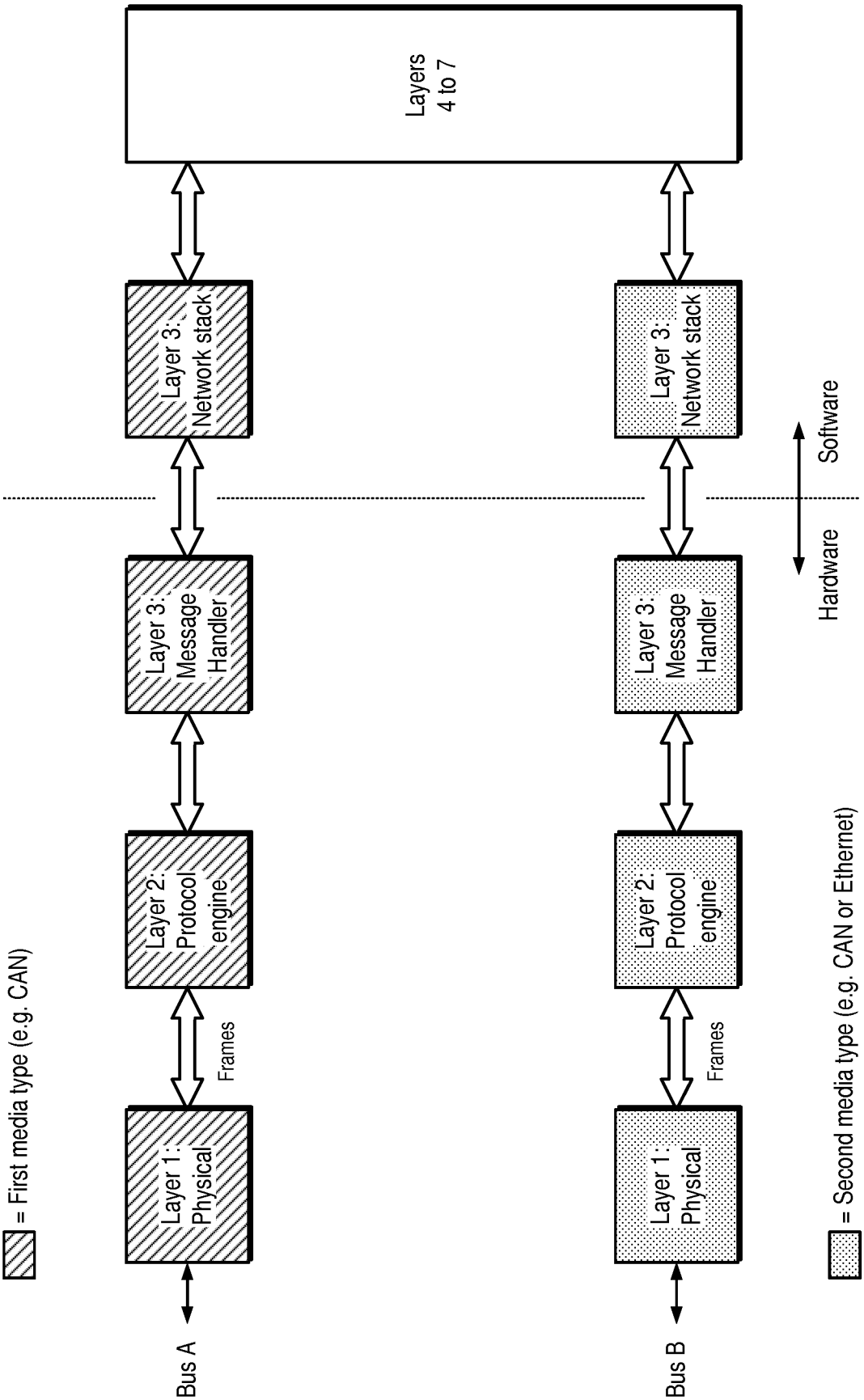

Layers
4 to 7
Layer 3:
Network stack
Layer 3:
Network stack
Software
Hardware
Layer 3:
Message
Handler
Layer 3:
Message
Handler
Layer 2:
Protocol
engine
Layer 2:
Protocol
engine
Frames
Frames
Layer 1:
Physical
Layer 1:
Physical
Bus A
Bus B
= First media type (e.g. CAN)
= Second media type (e.g. CAN or Ethernet)

Fig. 1

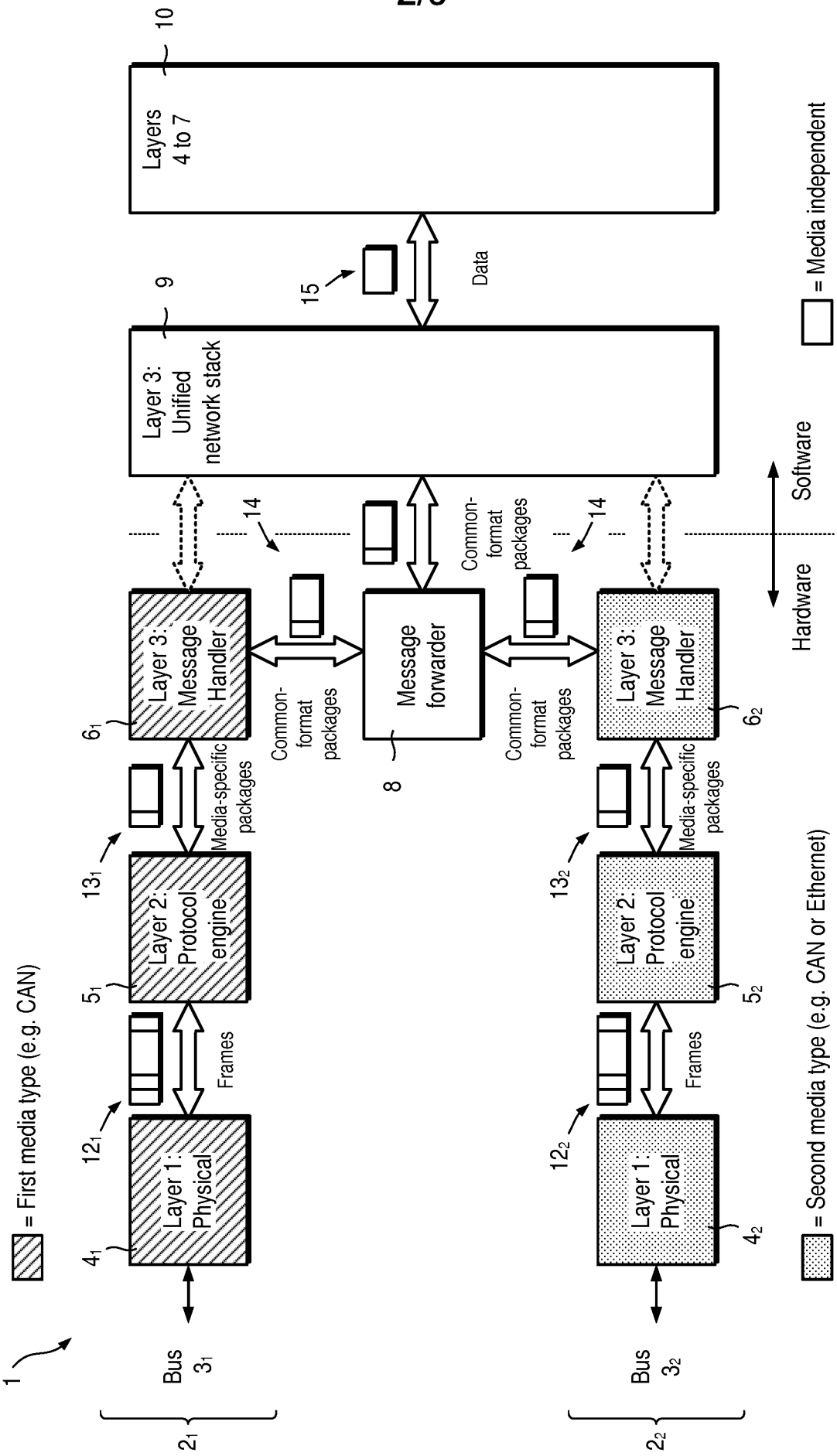

1
= First media type (e.g. CAN)
= Second media type (e.g. CAN or Ethernet)
= Media independent
Software
Hardware
Layers 4 to 7
10
Layer 3: Unified network stack
9
Data
15
14
Common-format packages
Message forwarder
8
Layer 3: Message Handler
6₁
Layer 3: Message Handler
6₂
Media-specific packages
Layer 2: Protocol engine
5₁
Layer 2: Protocol engine
5₂
Frames
Layer 1: Physical
4₁
Layer 1: Physical
4₂
Bus
3₁
Bus
3₂
2₁
2₂
12₁
12₂
13₁
13₂

Fig. 2

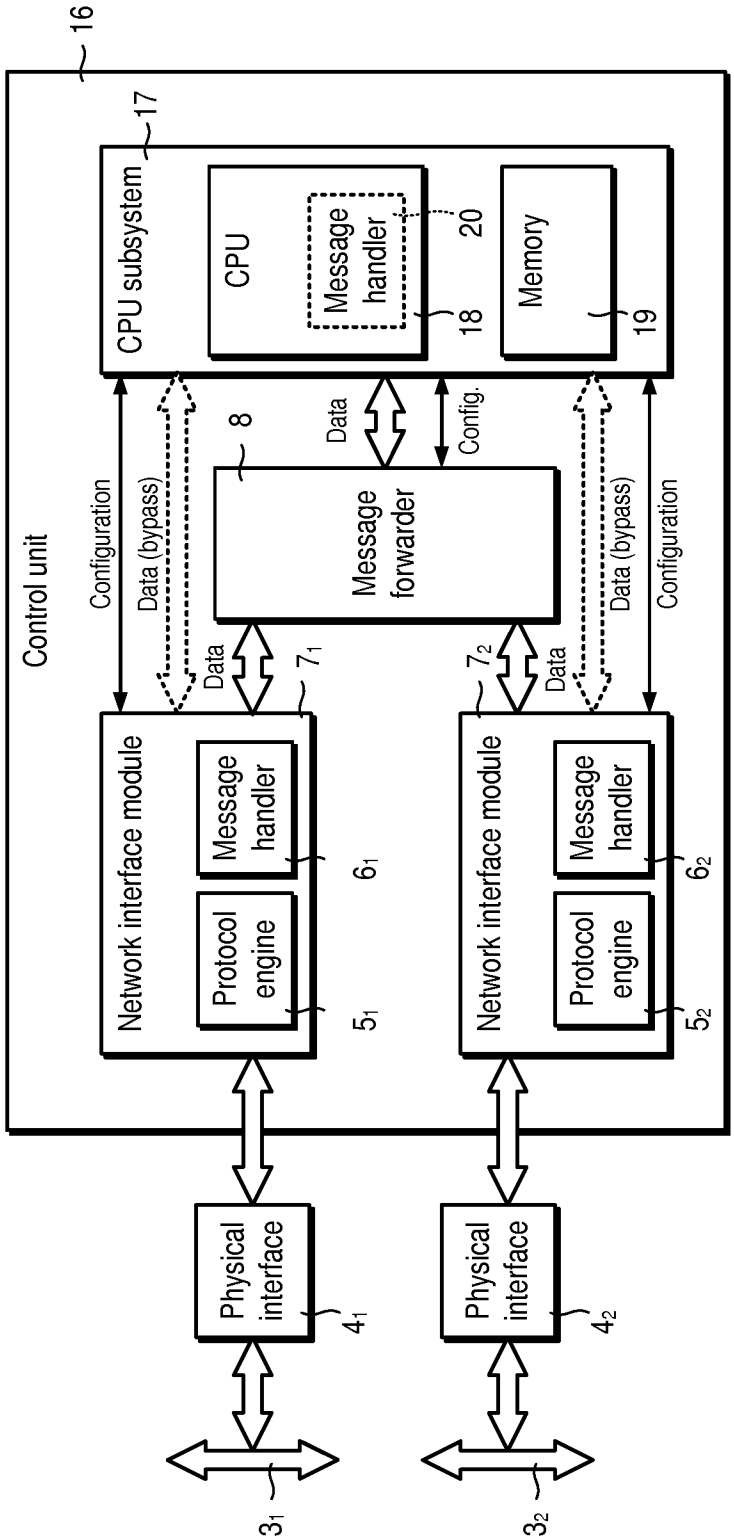

16
Control unit
17
CPU subsystem
CPU
Message handler
20
18
Memory
19
8
Message forwarder
Data
Config.
Configuration
Data (bypass)
Data
Configuration
Data (bypass)
Data
7_1
7_2
Network interface module
Message handler
Protocol engine
6_1
5_1
Network interface module
Message handler
Protocol engine
6_2
5_2
Physical interface
4_1
Physical interface
4_2
3_1
3_2

Fig. 3

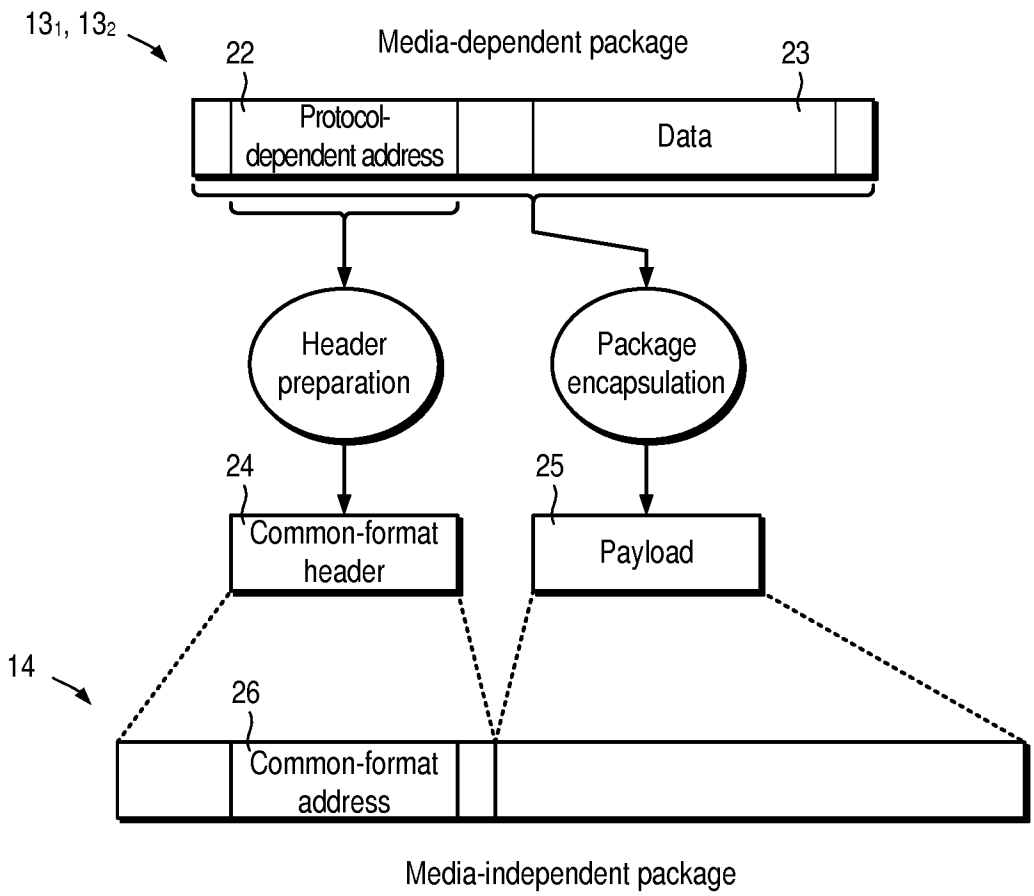

Conversion into common-format package
13₁, 13₂
22
Media-dependent package
23
Protocol-dependent address
Data
Header preparation
Package encapsulation
24
25
Common-format header
Payload
14
26
Common-format address
Media-independent package

Fig. 4

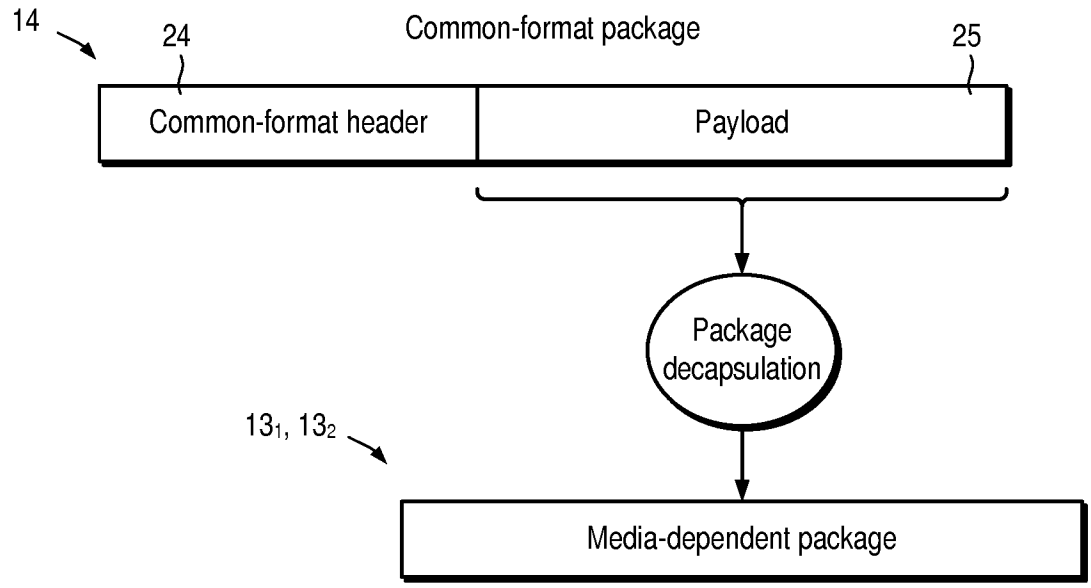

Conversion from common-format package
14
24
Common-format package
25
Common-format header
Payload
Package decapsulation
13₁, 13₂
Media-dependent package

Fig. 5

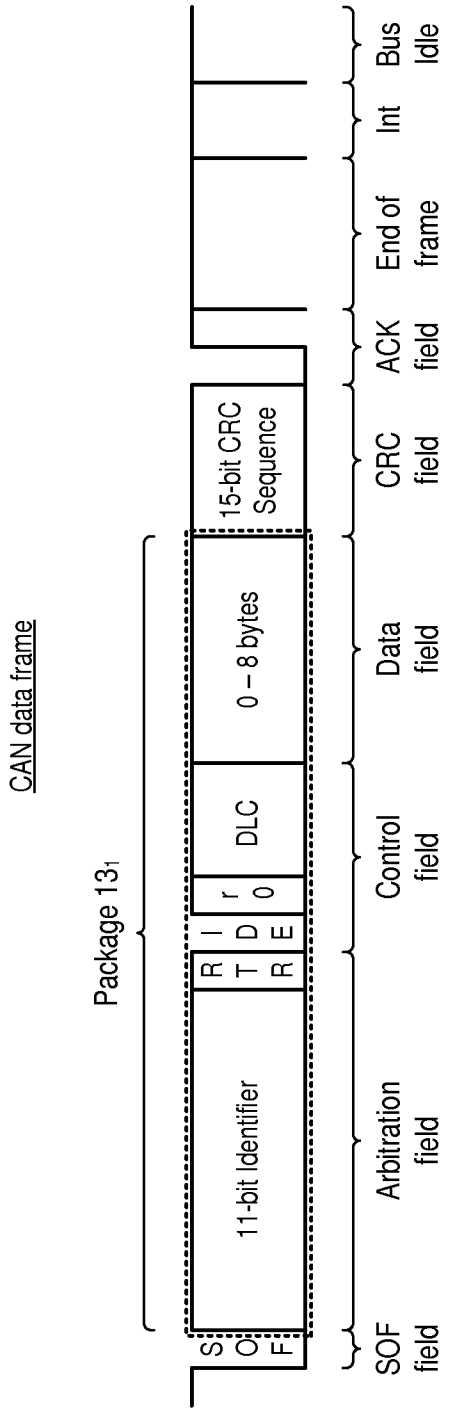
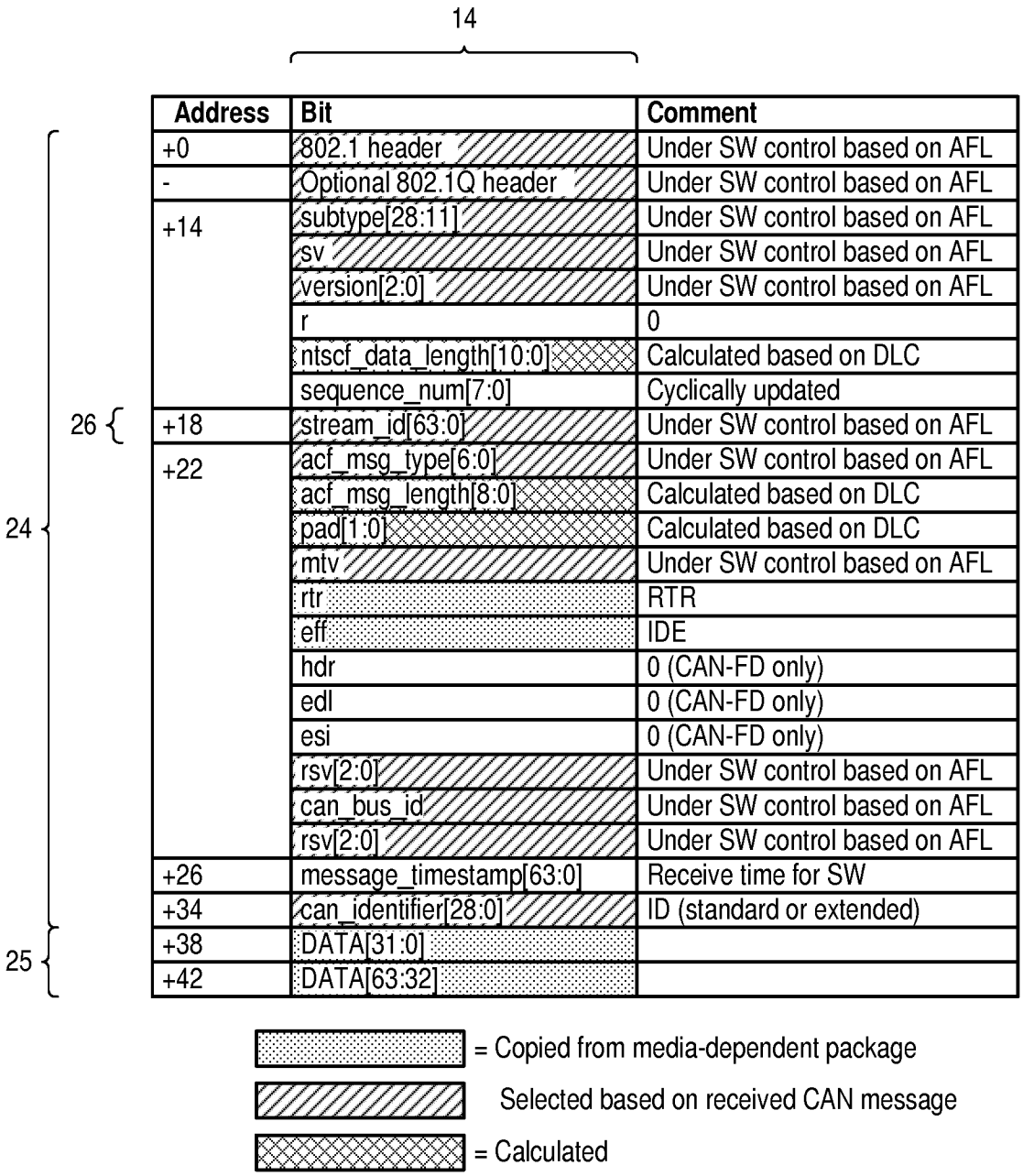
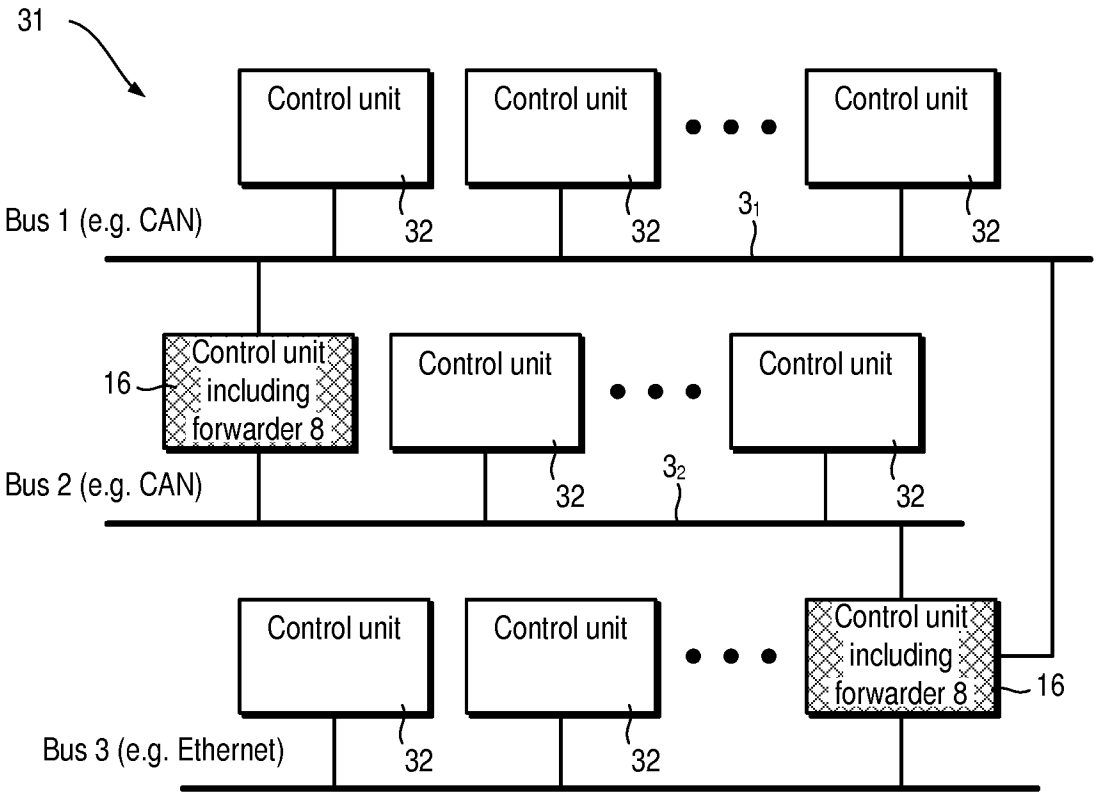

CAN data frame
Package 13₁
SOF
11-bit Identifier
RTR
IDE
r0
DLC
0 – 8 bytes
15-bit CRC Sequence
SOF field
Arbitration field
Control field
Data field
CRC field
ACK field
End of frame
Int
Bus Idle

Fig. 6



Address	Bit	Comment
+0	802.1 header	Under SW control based on AFL
-	Optional 802.1Q header	Under SW control based on AFL
+14	subtype[28:11]	Under SW control based on AFL
	sv	Under SW control based on AFL
	version[2:0]	Under SW control based on AFL
	r	0
	ntscf_data_length[10:0]	Calculated based on DLC
	sequence_num[7:0]	Cyclically updated
+18	stream_id[63:0]	Under SW control based on AFL
+22	acf_msg_type[6:0]	Under SW control based on AFL
	acf_msg_length[8:0]	Calculated based on DLC
	pad[1:0]	Calculated based on DLC
	mtv	Under SW control based on AFL
	rtr	RTR
	eff	IDE
	hdr	0 (CAN-FD only)
	edl	0 (CAN-FD only)
	esi	0 (CAN-FD only)
	rsv[2:0]	Under SW control based on AFL
	can_bus_id	Under SW control based on AFL
	rsv[2:0]	Under SW control based on AFL
+26	message_timestamp[63:0]	Receive time for SW
+34	can_identifier[28:0]	ID (standard or extended)
+38	DATA[31:0]	
+42	DATA[63:32]	

Fig. 7


31
Control unit
Control unit
Control unit
Bus 1 (e.g. CAN)
32
32
3_1
32
16
Control unit including forwarder 8
Control unit
Control unit
Bus 2 (e.g. CAN)
32
3_2
32
Control unit
Control unit
Control unit including forwarder 8
16
Bus 3 (e.g. Ethernet)
32
32

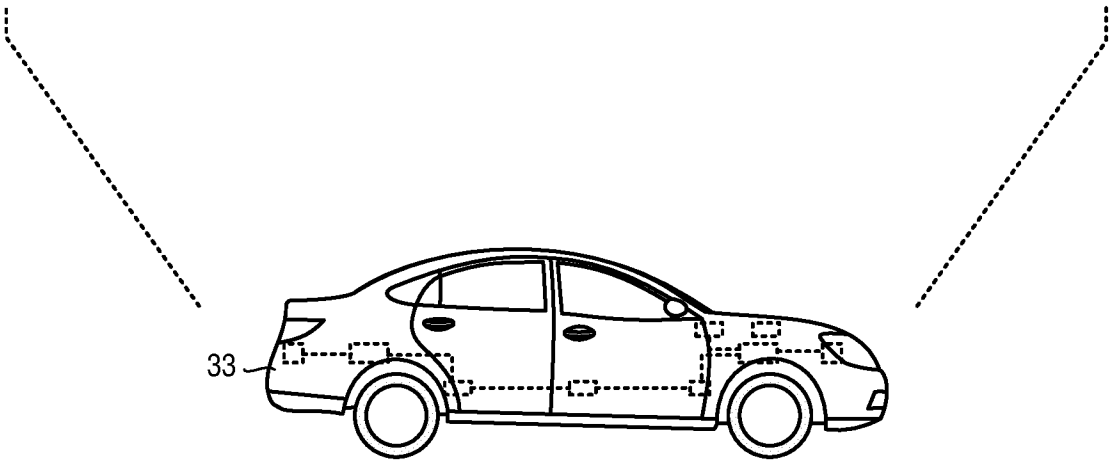
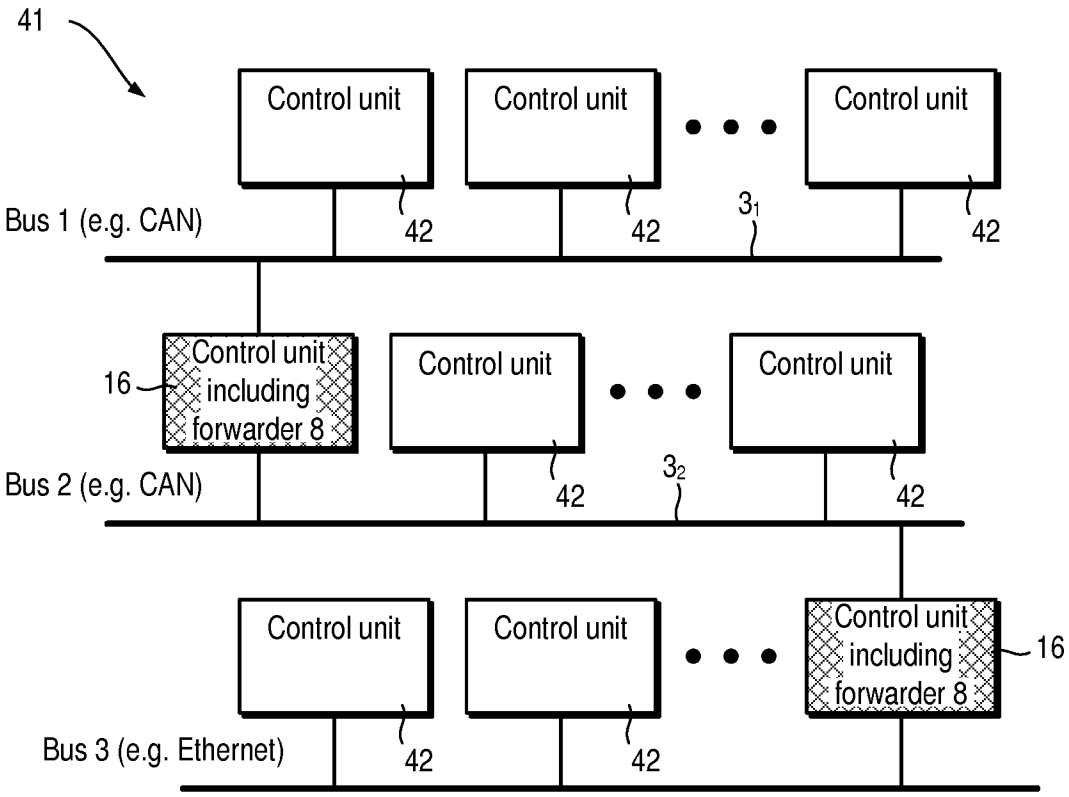

33

Fig. 8


41
Control unit
Control unit
Control unit
Bus 1 (e.g. CAN)
42
42
3_1
42
16
Control unit including forwarder 8
Control unit
Control unit
Bus 2 (e.g. CAN)
42
3_2
42
Control unit
Control unit
Control unit including forwarder 8
16
Bus 3 (e.g. Ethernet)
42
42

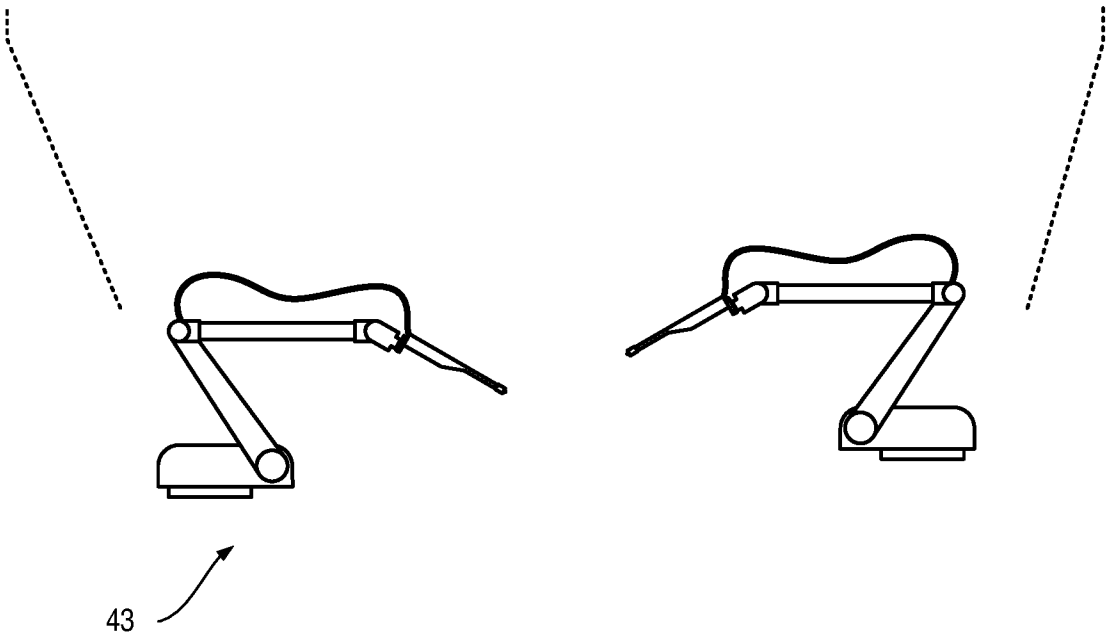

43

Fig. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/053459

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/725 H04L12/66
ADD.

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007/014278 A1 (EBBESEN BJORN [DE] ET AL) 18 January 2007 (2007-01-18) figures 3, 5 page 1, paragraphs 8, 9 page 3, paragraph 35 page 7, paragraphs 80, 81 page 8, paragraphs 81, 82 page 10, paragraph 102 -----	1-18, 20-29
A	EP 2 938 046 A1 (ALCATEL LUCENT [FR]) 28 October 2015 (2015-10-28) figure 1 column 2, paragraph 10 column 4, paragraph 26 column 6, paragraph 38 ----- -/--	1-18, 20-29

[X] Further documents are listed in the continuation of Box C.

[X] 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

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 July 2016	27/09/2016

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

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/053459

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 2 833 585 A1 (NEC CORP [JP]) 4 February 2015 (2015-02-04) figures 1, 5 page 3, paragraph 15 page 5, paragraph 22 page 6, paragraph 38 -----	1-18, 20-29
A	US 2012/099589 A1 (KATO TAISUKE [JP]) 26 April 2012 (2012-04-26) figures 1, 4, 9 page 4, paragraph 62 page 5, paragraph 69 page 17; claim 1 -----	1-18, 20-29

1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2016/053459

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-18, 20-29

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-18, 20-29

Content dependent routing

---

2. claim: 19

Gateway for two different communication protocols

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2016/053459

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007014278 A1	18-01-2007	CN 101317416 A	03-12-2008
		EP 1908256 A2	09-04-2008
		JP 4881380 B2	22-02-2012
		JP 2009510811 A	12-03-2009
		KR 20080032183 A	14-04-2008
		US 2007014278 A1	18-01-2007
		US 2007014303 A1	18-01-2007
		US 2007028000 A1	01-02-2007
		WO 2007011533 A2	25-01-2007
EP 2938046 A1	28-10-2015	EP 2938046 A1	28-10-2015
		WO 2015177642 A1	26-11-2015
EP 2833585 A1	04-02-2015	CN 104205749 A	10-12-2014
		EP 2833585 A1	04-02-2015
		JP 5858147 B2	10-02-2016
		JP WO2013146885 A1	14-12-2015
		US 2015043588 A1	12-02-2015
		WO 2013146885 A1	03-10-2013
US 2012099589 A1	26-04-2012	CN 102498479 A	13-06-2012
		CN 104811486 A	29-07-2015
		HK 1172105 A1	24-03-2016
		HK 1212520 A1	10-06-2016
		JP 5670897 B2	18-02-2015
		JP 5913548 B2	27-04-2016
		JP 2015062142 A	02-04-2015
		US 2012099589 A1	26-04-2012
		WO 2010147221 A1	23-12-2010