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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
06 September 2019 (06.09.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/166310 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *H04L 29/08* (2006.01)

(21) International Application Number:
PCT/EP2019/054265

(22) International Filing Date:
21 February 2019 (21.02.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
18159124.9 28 February 2018 (28.02.2018) EP

(71) Applicant: **DEUTSCHE TELEKOM AG** [DE/DE]; Friedrich-Ebert-Allee 140, 53113 Bonn (DE).

(72) Inventors: **AMEND, Markus**; Elbestrasse 12, 63667 Nidda (DE). **BOGENFELD, Eckard**; Wattenheimer Str. 3a, 67316 Carlsberg (DE).

(74) Agent: **BRAUN-DULLAEUS PANNEN EMMERLING PATENT- & RECHTSANWALTSPARTNERSCHAFT MBB**; Platz der Ideen 2, 40476 Düsseldorf (DE).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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,

(54) **Title:** TECHNIQUES FOR PACKET DATA CONVERSION

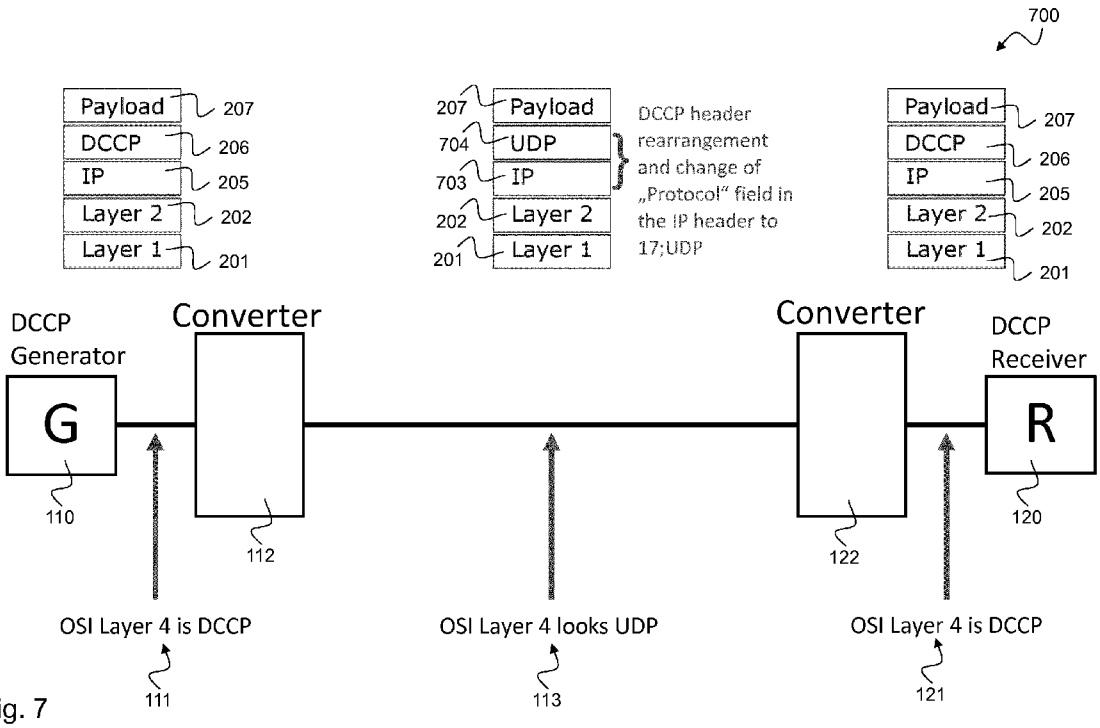

Fig. 7

(57) **Abstract:** The disclosure relates to techniques for network protocol conversion, in particular to a first packet data conversion device (810) for a sender, the packet data conversion device (810) comprising: a data interface (811) configured to provide first packet data (813) according to a first network protocol (814), in particular according to a Datagram Congestion Control Protocol, DCCP, wherein each packet of the first packet data (813) comprises a first packet header (815); and a converter (812) configured to convert the first packet data (813) into second packet data (817), wherein the conversion is based on rearranging contents of the first packet header (815), wherein the rearranged first packet header (819) indicates that the second packet data (817) is generated according to a second network protocol (818), in particular according to a User Datagram Protocol,UDP or according to a UDP-Lite protocol.

WO 2019/166310 A1

[Continued on next page]

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))*

Techniques for packet data conversion

TECHNICAL FIELD

The invention relates to techniques for packet data conversion, in particular for conversion of DCCP (Datagram Congestion Control Protocol) packet data such that it looks like UDP (User Datagram Protocol) packet data, in order to transmit the converted DCCP packet data over a middle box that supports only UDP packet data. The invention further relates to DCCP header converter devices for UDP like appearance.

BACKGROUND

The Transmission Control Protocol (TCP) according to *"University of Southern California, "TRANSMISSION CONTROL PROTOCOL" RFC no. 793, September 1981"* is widely used in today's computer networks to ensure reliable data transfer on OSI (Open System Interconnection) Layer 4. Therefore it uses congestion control to adapt to path characteristic and flow control to enable reliable in-order delivery, which can cause head-of-line blocking. Especially the head-of-line blocking is an unwanted feature from an application perspective, e.g. for real-time communication. Most of such applications use therefore the OSI Layer 4 User Datagram Protocol (UDP) according to *"J. Postel, "User Datagram Protocol RFC no. 768, 28 August 1980"*, which is stateless and does not include any congestion- or flow-control. Alternatively, UDP-Lite according to RFC3828 may be used which has the protocol number 136 instead of 17 for UDP. To ensure path adaption, the applications mostly implement an own congestion control on top which is time consuming and error-prone. Another OSI Layer 4 protocol exists which solves this issue: The plain old Datagram Congestion Control Protocol (DCCP) according to *"E. Kohler, M. Handley, S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC no. 4340, March 2006"* is a protocol, which inherits TCP like congestion control, without ensuring (in-order) delivery. Applications, which use it, can leave congestion control to Layer 4 without taking care.

In principle with DCCP it seems it might be a good alternative for many applications, which use UDP today and implement their own congestion control. However, DCCP has

the classic chicken-and-egg problem. Because it is not widely deployed, middle boxes often block it on the way to the destination and that is again the reason for not using DCCP as application developer. TCP and UDP are the only protocols from which a reasonable acceptance by the middle boxes can be expected.

SUMMARY OF THE INVENTION

It is the object of the present invention to provide a concept for solving the above-mentioned problem, i.e. how to efficiently distribute DCCP data traffic over middle boxes that are designed for UDP or TCP data traffic.

A further object of the present invention is to introduce a concept for an efficient transmission of data traffic over network devices that support only specific network protocols.

The foregoing and other objects are achieved by the subject matter of the independent claims. Further implementation forms are apparent from the dependent claims, the description and the figures.

To overcome the challenge of traversing middle boxes like firewalls and NAT (network address translation) the only reasonable solution is, without believing DCCP will be allowed to pass them in future, to convert it to common protocols like TCP or UDP. Because DCCP wants to avoid TCP behavior, the solution space is henceforth limited to UDP.

This disclosure focuses on two different types of converter devices, which transform DCCP datagrams that they look like UDP. A DCCP connection needs at least two converter devices, one for transforming it to UDP and another one to undone it as shown in Fig. 1. A first solution is to encapsulate first each DCCP datagram on the send converter into an UDP datagram and second strip it again in the receiver converter, which can be also described as DCCP over UDP transmission as shown in Fig. 2, and is rather simply. However, beside the simplicity it has the impact to change the size proportion by injecting at least an additional UDP header and eventually a header of the OSI network layer, e.g. IP. Thereby, overhead is introduced which needs at least to be transferred and

may lead in addition to fragmentation and/or payload reduction. A second solution, that is the more graceful solution, bases mainly on converting the DCCP header (see Fig. 3) to a UDP looking like as exemplarily shown in Figures 6 and 7. A general representation of this second solution is shown in Fig. 8.

The methods and systems presented below may be of various types. The individual elements described may be realized by hardware or software components, for example electronic components that can be manufactured by various technologies and include, for example, semiconductor chips, ASICs, microprocessors, digital signal processors, integrated electrical circuits, electro-optical circuits and/or passive components.

The devices, systems and methods presented below are capable of transmitting information over a communication network. The term communication network refers to the technical infrastructure on which the transmission of signals takes place. The communication network essentially comprises the switching network in which the transmission and switching of the signals takes place between the stationary devices and platforms of the mobile radio network or fixed network, and the access network in which the transmission of the signals takes place between a network access device and the communication terminal. The communication network can comprise both components of a mobile radio network as well as components of a fixed network. In the mobile network, the access network is also referred to as an air interface and includes, for example, a base station (NodeB, eNodeB, radio cell) with mobile antenna to establish the communication to a communication terminal as described above, for example, a mobile phone or a mobile device with mobile adapter or a machine terminal. In the fixed network, the access network includes, for example, a DSLAM (digital subscriber line access multiplexer) to connect the communication terminals of multiple participants based on wires. Via the switching network the communication can be transferred to other networks, for example other network operators, e.g. foreign networks.

In the following, network protocols, also denoted as communication protocols are described. A network protocol is a system of rules that allow two or more entities of a communications system to transmit information via a communication channel or transmission medium. The network protocol defines the rules “Syntax”, “Semantic” and “Synchronization” of communication and possible error detection and correction. Network

protocols may be implemented by computer hardware, software, or a combination of both. Communicating systems use well-defined formats for exchanging various messages. Each message has an exact meaning intended to elicit a response from a range of possible responses pre-determined for that particular situation. The specified behavior is typically independent of how it is to be implemented. Communication protocols have to be agreed upon by the parties involved. To reach agreement, a network protocol may be developed into a technical standard. Multiple protocols often describe different aspects of a single communication. A group of (network) protocols designed to work together are known as a (network) protocol suite; when implemented in software they are a (network) protocol stack. Internet communication protocols are published by the Internet Engineering Task Force (IETF). The IEEE handles wired and wireless networking, and the International Organization for Standardization (ISO) handles other types.

In communications and computing systems, the Open Systems Interconnection model (OSI model) defines a conceptual model that characterizes and standardizes the communication functions without regard to its underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard protocols. The model partitions a communication system into abstraction layers. The original version of the model defined seven layers: Physical layer (Layer 1), Data Link layer (Layer 2), Network layer (Layer 3), Transport layer (Layer 4), Session layer (Layer 5), Presentation layer (Layer 6) and Application layer (Layer 7).

According to a first aspect the invention relates to a first packet data conversion device for a sender, the packet data conversion device comprising: a data interface configured to provide first packet data according to a first network protocol, in particular according to a Datagram Congestion Control Protocol, DCCP, wherein each packet of the first packet data comprises a first packet header; and a converter configured to convert the first packet data into second packet data, wherein the conversion is based on rearranging contents of the first packet header, wherein the rearranged first packet header indicates that the second packet data is generated according to a second network protocol, in particular according to a User Datagram Protocol, UDP or to a UDP-Lite Protocol.

Such a first packet data conversion device provides efficient distribution of DCCP data traffic over middle boxes that are designed for UDP (or UDP-Lite) or TCP data traffic since

the rearranged first packet header indicates that the second packet data is generated according to a second network protocol, in particular according to UDP or according to UDP-Lite. In particular, such a first packet data conversion device provides efficient transmission of data traffic over network devices that support only specific network protocols, namely the second network protocol but blocks data traffic according to the first network protocol. UDP-Lite is even easier with respect to re-arrangement, because it already comprises data fields for ports, CsCov and Chksum (checksum).

In an exemplary implementation form of the first packet data conversion device, a length of the rearranged first packet header is equal to a length of the first packet header.

This provides the advantage that the converter does not change the size of the first packet header and avoids introducing overhead. Hence no overhead has to be additionally transmitted which may result in fragmentation and/or payload reduction.

In an exemplary implementation form of the first packet data conversion device, a length of a packet of the second packet data is equal to a length of a corresponding packet of the first packet data.

This provides the advantage that the converter does not change the size of the whole first packet data and avoids introducing overhead. Hence no overhead has to be additionally transmitted which may result in fragmentation and/or payload reduction.

In an exemplary implementation form of the first packet data conversion device, the conversion is further based on changing a protocol field of the first packet header from indicating the first network protocol to indicating the second network protocol.

This provides the advantage that network devices within the transmission path between sender and receiver, so called middle boxes, can detect the second network protocol in the protocol field and hence will pass the first packet data if they are designed to transmit data traffic according to the second network protocol, e.g. UDP data traffic or UDP-Lite data traffic.

In an exemplary implementation form of the first packet data conversion device, the first packet header comprises an Internet Protocol, IP, header, and the protocol field is a protocol field of the IP header.

This provides the advantage that Internet routers and/or gateways can detect the protocol field of the IP header and may forward the first packet data.

In an exemplary implementation form of the first packet data conversion device, the converter is configured to change the protocol field from a value of 33 indicating the DCCP protocol to a value of 17 indicating the UDP protocol or to a value of 136 indicating the UDP-Lite protocol.

This provides the advantage that typical middle boxes that are designed to transfer UDP (or UDP-Lite) or TCP data traffic are able to detect the first packet data as being UDP (or UDP-Lite) data traffic. Hence, first packet data is not blocked by such middle boxes.

In an exemplary implementation form of the first packet data conversion device, the conversion is further based on extending a checksum coverage field, CsCov, of a respective packet of the first packet data, wherein the CsCov indicates a range of the respective packet which is covered by a checksum.

This provides the advantage that the checksum can be extended over a broader range of the first packet data resulting in more reliable transmission.

In an exemplary implementation form of the first packet data conversion device, the CsCov is extended to a size of a length field comprised in a packet header of the second network protocol, in particular a packet header of the UDP protocol or of the UDP-Lite protocol.

This provides the advantage that the CsCov can be interpreted as the UDP (or UDP-Lite) length field and thus, first packet data is interpreted as UDP (or UDP-Lite) traffic by usual middle boxes which are designed to forward UDP (or UDP-Lite) traffic but to block DCCP or other traffic.

In an exemplary implementation form of the first packet data conversion device, the conversion is further based on rearranging at least one of a type field, a CCVal field and a data offset field of the first packet header to another position in the rearranged first packet header.

This provides the advantage that these type, CCVal and data offset fields can be shifted further back in the first packet header, e.g. at positions that are not checked by middle boxes to identify UDP (or UDP-Lite) data traffic.

In an exemplary implementation form of the first packet data conversion device, the rearranged first packet header comprises the following data fields, in particular in the following sequence: a source port, a destination port, a checksum coverage, CsCov, a checksum, a type, a CCVal, a data offset and a sequence number.

This provides the advantage that the first part of the rearranged first packet header which includes the source port, the destination port, the checksum coverage, CsCov, and the checksum looks like a UDP (or UDP-Lite) header. When checking this first part by typical middle boxes, the first packet data is identified as UDP data traffic.

In an exemplary implementation form of the first packet data conversion device, the first packet data and the second packet data are segmented data streams according to an OSI layer 4 representation.

This provides the advantage that the services of the transport layer (Layer 4) according to the OSI specification can be provided, e.g. connection-oriented communication, same order delivery, reliability, flow control, congestion control and multiplexing.

According to a second aspect, the invention relates to a second packet data conversion device for a receiver, the second packet data conversion device comprising: a data interface configured to receive second packet data according to a second network protocol, in particular according to a User Datagram Protocol, UDP, or according to a UDP-Lite protocol, wherein each packet of the second packet data comprises a second packet header; and a converter configured to convert the second packet data into first packet data, wherein the conversion is based on rearranging contents of the second

packet header, wherein the rearranged second packet header indicates that the first packet data is generated according to a first network protocol, in particular according to a Datagram Congestion Control Protocol, DCCP.

Such a second packet data conversion device provides efficient distribution of DCCP data traffic over middle boxes that are designed for UDP (or UDP-Lite) or TCP data traffic since the second packet header indicates that the second packet data is generated according to a second network protocol, in particular according to UDP (or UDP-Lite) and hence can pass these middle boxes. The converter of the second packet data conversion device reconverts the transmitted data traffic back to its original network protocol, namely to the first network protocol, in particular DCCP. Such a second packet data conversion device provides (together with the above-described first packet data conversion device) efficient transmission of data traffic over network devices that support only specific network protocols, namely the second network protocol but block data traffic according to the first network protocol.

According to a third aspect, the invention relates to a transmission system, comprising: a sender configured to send first packet data over a transmission channel, wherein the sender comprises a first packet data conversion device according to the first aspect, which is configured to convert the first packet data into second packet data before sending; and a receiver configured to receive second packet data via the transmission channel, wherein the receiver comprises a second packet data conversion device according to the second aspect, which is configured to convert the second packet data into first packet data after reception.

Such a transmission system provides efficient distribution of first packet data that is generated according to a first network protocol over a transmission channel having transmission paths supporting only transmission of data traffic generated according to a second network protocol since the first packet header looks like a packet header of the second network protocol.

In an exemplary implementation form, the transmission system comprises a middle box between the sender and the receiver, wherein the middle box is configured to transfer data according to the second network protocol, in particular according to the UDP (or

UDP-Lite) protocol and block data according to the first network protocol 814, in particular according to the DCCP protocol.

Such a transmission system provides efficient distribution of DCCP data traffic over middle boxes that support only UDP (or UDP-Lite) or TCP data traffic, since the DCCP header is rearranged resembling a UDP header.

According to a fourth aspect, the invention relates to a method for packet data conversion, the method comprising: providing first packet data according to a first network protocol, in particular according to a Datagram Congestion Control Protocol, DCCP, wherein each packet of the first packet data comprises a first packet header; and converting the first packet data into second packet data, wherein the converting is based on rearranging contents of the first packet header, wherein the rearranged first packet header indicates that the second packet data is generated according to a second network protocol, in particular according to a User Datagram Protocol, UDP or according to a UDP-Lite protocol.

Such a method provides efficient distribution of DCCP data traffic over middle boxes that are designed for UDP (or UDP-Lite) or TCP data traffic since the rearranged first packet header indicates that the second packet data is generated according to a second network protocol, in particular according to UDP (or UDP-Lite). In particular, such a method provides efficient transmission of data traffic over network devices that support only specific network protocols, namely the second network protocol but blocks data traffic according to the first network protocol.

According to a fifth aspect the invention relates to a computer program product comprising program code for performing the method according to the fourth aspect of the invention, when executed on a computer or a processor.

Embodiments of the invention can be implemented in hardware and/or software.

The following acronyms are applied in this disclosure:

API	Application Interface
DCCP	Datagram Congestion Control Protocol

IP	Internet Protocol
OSI	Open System Interconnection
TCP	Transport Control Protocol
UDP	User Datagram Protocol

BRIEF DESCRIPTION OF THE DRAWINGS

Further embodiments of the invention will be described with respect to the following figures, wherein:

Fig. 1 shows a block diagram illustrating an exemplary DCCP UDP converter architecture 100;

Fig. 2 shows a block diagram illustrating an OSI layer view of a DCCP over UDP transmission system 200;

Fig. 3 shows a bit diagram illustrating a DCCP header structure 300 according to RFC 4340;

Fig. 4 shows a bit diagram illustrating a UDP header structure 400 according to RFC 768;

Fig. 5 shows a bit diagram illustrating an IP header structure 400 according to RFC 791;

Fig. 6 shows a bit diagram illustrating an exemplary DCCP to UDP header conversion 600 according to the disclosure;

Fig. 7 shows a block diagram illustrating an OSI layer view of a transmission system 700 with an exemplary DCCP to UDP header conversion according to the disclosure;

Fig. 8 shows a block diagram illustrating a general transmission system 800 with first 810 and second 820 packet data conversion devices applying header conversion according to the disclosure; and

Fig. 9 shows a schematic diagram illustrating a method 900 for packet data conversion according to the disclosure.

DETAILED DESCRIPTION OF EMBODIMENTS

In the following detailed description, reference is made to the accompanying drawings, which form part of the disclosure, and in which are shown, by way of illustration, specific aspects in which the present invention may be placed. It is understood that other aspects may be utilized and structural or logical changes may be made without departing from the scope of the present invention. The following detailed description, therefore, is not to be taken in a limiting sense, as the scope of the present invention is defined by the appended claims.

For instance, it is understood that a disclosure in connection with a described method may also hold true for a corresponding device or system configured to perform the method and vice versa. For example, if a specific method step is described, a corresponding device may include a unit to perform the described method step, even if such unit is not explicitly described or illustrated in the figures. Further, it is understood that the features of the various exemplary aspects described herein may be combined with each other, unless specifically noted otherwise.

Fig. 1 shows a block diagram illustrating an exemplary DCCP UDP converter architecture 100. A DCCP connection needs at least two converter devices 112, 122, one 112 (left-hand side of Fig. 1) for transforming it to UDP and another one 122 (right-hand side of Fig. 1) to undo it. At sender side, i.e. left side of Fig. 1, a DCCP generator 110 generates a DCCP data stream 111 at OSI layer 4 according to the Open System Interconnection (OSI) reference model. The converter 112 on sender side converts this DCCP data stream 111 in a converted data stream (on OSI layer 4) 113 that looks like UDP. At receiver side, i.e. right-hand side of Fig. 1, the converter 122 at receiver side (re-)converts this converted data stream looking like UDP into a re-converted DCCP data stream 121 (on OSI layer 4) which is then forwarded to DCCP receiver 120.

The converter devices 112, 122 can be designed to support two different kinds of conversion techniques for transforming DCCP datagrams that they look like UDP. A first one (shown in Fig. 2) is based on encapsulating each DCCP datagram on the send converter into a UDP datagram, according to DCCP over UDP transmission, while the second one (shown in Figures 6 and 7) is based on rearranging the contents of the DCCP header in such a manner that it looks like a UDP header. Both techniques allow DCCP packet data to pass middle boxes in between sender and receiver which are designed to support UDP data traffic but to block DCCP data traffic.

Fig. 2 shows a block diagram illustrating an OSI layer view of a DCCP over UDP transmission system 200. The transmission system 200 corresponds to the system described above with respect to Fig. 1 in which the converters 112, 122 are designed to first encapsulate each DCCP datagram on the send converter 112 into a UDP datagram, according to DCCP over UDP transmission and second strip it again in the receiver converter 122.

In the example of Fig. 2, the DCCP data stream 111 includes Layer 1, 201, Layer 2, 202, an IP layer 205 as Layer 3, a DCCP layer 206 as Layer 4 and the payload 207 as higher Layers. After the send converter 112 the converted data stream 113 additionally includes an encapsulated IP layer 203 and an encapsulated UDP layer 204 as shown in the middle left section of Fig. 2 or alternatively an encapsulated UDP layer 204 between IP layer 205 and DCCP layer 206 as shown in the middle right section of Fig. 2. After the receive converter 112 these additional layers 203, 204 (first alternative) or this single additional layer 204 (second alternative) are stripped again resulting in the receive DCCP data stream 121 that has the same structure than the original DCCP data stream 111, i.e. including Layer 1, 201, Layer 2, 202, IP layer 205 as Layer 3, DCCP layer 206 as Layer 4 and the payload 207 as higher Layers.

The solution according to Fig. 2 is rather simple. However, beside the simplicity it has the impact to change the size proportion by injecting at least an additional UDP header 204 and eventually a header of the OSI network layer, e.g. IP 203. Thereby, overhead is introduced which needs at least to be transferred and may lead in addition to fragmentation and/or payload reduction.

Fig. 3 shows a bit diagram illustrating a DCCP header structure 300 of a DCCP network protocol according to RFC 4340. The Datagram Congestion Control Protocol (DCCP) is a message-oriented transport layer protocol. DCCP implements reliable connection setup, teardown, Explicit Congestion Notification (ECN), congestion control, and feature negotiation. DCCP provides a way to gain access to congestion-control mechanisms without having to implement them on higher layers than layer 4. It allows for flow-based semantics like in Transmission Control Protocol (TCP), but does not provide reliable in-order delivery. A DCCP connection contains acknowledgment traffic as well as data traffic.

The DCCP header 300 includes a source port (16 bit), a destination port (16 bit), a data offset (8 bit) 307, a CCVal (4 bit) 306, a checksum coverage, CsCov, (4 bit) 301, the checksum (16 bit), a reserved bit field (3 bit), a type field (4 bit), an X=1 field (1 bit), a further reserved bit field (8 bit), a sequence number (high bits) (16 bit) and a sequence number (low bits) (32 bit).

According to the second solution as described above with respect to Fig. 1, the contents of the DCCP header, in particular the data offset 307, the CCVal 306, the CsCov 301 and the type 305 bit fields is rearranged in such a manner that it looks like a UDP header as described below. In some embodiment, the CsCov 301 bit field has to be additionally converted to a respective length and/or value field of the UDP header. Checksum Coverage, CsCov, determines the parts of the packet that are covered by the Checksum field. This second solution has not the drawback to inject extra information as in the first solution described above with respect to Fig. 2 rather exploiting the already used space of the DCCP header and rearrange the header content.

Fig. 4 shows a bit diagram illustrating a UDP header structure 400 of a UDP network protocol according to RFC 768. With the User Datagram Protocol (UDP), computer applications can send messages, in this case referred to as *datagrams*, to other hosts on an Internet Protocol (IP) network. Prior communications are not required in order to set up communication channels or data paths. UDP uses a simple connectionless communication model with a minimum of protocol mechanism. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. It has no handshaking dialogues, and thus exposes the

user's program to any unreliability of the underlying network. There is no guarantee of delivery, ordering, or duplicate protection.

The UDP header 400 includes a source port (16 bit), a destination port (16 bit), a length field 401 and a checksum. In some embodiment, source port, destination port and checksum, depending on CsCov, can be the same as in the DCCP header 300 shown in Fig. 3. In some embodiment, the checksum may have to be recalculated due to the rearrangement. If the CsCov field 301 of the DCCP header 300 is enlarged to a 16 bit value and the CCVal 306 and Data Offset 307 are shifted to some other position in the rearranged DCCP header, this rearranged DCCP header looks like the UDP header 400 shown in Fig. 4. Additionally it behaves like a valid UDP packet in case if a middle box checks length and checksum.

Fig. 5 shows a bit diagram illustrating an IP header structure 500 of an IP network protocol according to RFC 791. The Internet Protocol (IP) is responsible for addressing hosts, encapsulating data into datagrams (including fragmentation and reassembly) and routing datagrams from a source host to a destination host across one or more IP networks. For these purposes, the Internet Protocol defines the format of packets and provides an addressing system. Each datagram has two components: a header and a payload. The IP header includes source IP address, destination IP address, and other metadata needed to route and deliver the datagram. The payload is the data that is transported. This method of nesting the data payload in a packet with a header is called encapsulation.

The IP header includes different bit fields such as Version, IHL, Type of Service, Total Length, Identification, Flags, Fragment Offset, Time to Live, Protocol, Header Checksum, Source Address, Destination Address, Options and Padding. For the second solution as described above with respect to Fig. 1, together with the rearrangement of the header contents, the converter devices have to change the “Protocol” field 501 in the IP header 500 to either 17;UDP (send converter) or 33;DCCP (receiver converter). In an alternative embodiment using the UDP-Lite protocol according to RFC 3828, the converter devices have to change the “Protocol” field 501 in the IP header 500 to either 136;UDP-Lite (send converter) or 33;DCCP (receiver converter).

Fig. 5 shows the original IP header (IPv4) according to RFC 791. The solution can also be applied to IP headers of IPv6 according to RFC 2460. The protocol field has to be correspondingly changed.

Fig. 6 shows a bit diagram illustrating an exemplary DCCP to UDP header conversion 600 according to the disclosure. The converted header 600 includes the following bit fields: Source Port 601, Destination Port 602, CsCov 603, Checksum 604, Type 605, CCVal 606, Data Offset 607, Sequence Number (high bits) 608 and Sequence Number (low bits) 609. Compared to the DCCP header 300 shown in Fig. 3, Source Port, 601, Destination Port 602 and Checksum 604 are at the same location while CsCov 603 is enlarged to a 16 bit value. Data Offset 307, CCVal 306 and Type 605 are shifted to some other position and Sequence Number (high bits) 608 and Sequence Number (low bits) 609 are at the same location. Reserved bit fields and X=1 bit field of DCCP header 300 are removed. The converted header 600 now matches 610 the UDP header 400 as shown in Fig. 4 since the CsCov bit field 603 of the converted header 600 resembles the length bit field 401 of the UDP header 400.

With this, each datagram looks between the converters like UDP without losing DCCP relevant information. Together with the rearrangement, the converter device has to change the “Protocol” field in the IP header as described above with respect to Fig. 5 to either 17;UDP (send converter) or 33;DCCP (receiver converter). In an alternative embodiment using the UDP-Lite protocol according to RFC 3828, the converter device has to change the “Protocol” field in the IP header as described above with respect to Fig. 5 to either 136;UDP-Lite (send converter) or 33;DCCP (receiver converter). Additionally it behaves like a valid UDP (or UDP-Lite) packet in case if a middle box checks length and checksum. The resulting architecture and relevant OSI layer change is shown in Fig. 7.

Fig. 7 shows a block diagram illustrating an OSI layer view of a transmission system 700 with an exemplary DCCP to UDP header conversion according to the disclosure.

The transmission system 700 corresponds to the system described above with respect to Fig. 1 in which the converters 112, 122 are designed to rearrange each DCCP header as described above with respect to Fig. 6 and change the protocol field in the IP header to

17, UDP (or alternatively 136, UDP-Lite) for the send converter 112 or 33, DCCP for the receive converter 122.

In the example of Fig. 7, the DCCP data stream 111 includes Layer 1, 201, Layer 2, 202, an IP layer 205 as Layer 3, a DCCP layer 206 as Layer 4 and the payload 207 as higher Layers. After the send converter 112 the converted data stream 113 includes a rearranged DCCP layer 206 that resembles a UDP layer 704 and a modified IP layer 703 in which the protocol field is changed to 17, UDP (or alternatively 136, UDP-Lite). After the receive converter 112 these rearranged layers 703, 704 are brought back to their original structure, resulting in the receive DCCP data stream 121 that has the same structure than the original DCCP data stream 111, i.e. including Layer 1, 201, Layer 2, 202, IP layer 205 as Layer 3, DCCP layer 206 as Layer 4 and the payload 207 as higher Layers.

Fig. 8 shows a block diagram illustrating a general transmission system 800 with first 810 and second 820 packet data conversion devices applying header conversion according to the disclosure. The transmission system 800 includes a first packet data conversion device 810 on sender side (left-hand side) and a second packet data conversion device 820 on receiver side (right-hand side). Both packet data conversion devices 810, 820 can be implemented at different network entities, e.g. a sender and a receiver, or at the same network entity, e.g. a transceiver providing both sender and receiver functionalities.

The first packet data conversion device 810 comprises a data interface 811 configured to provide first packet data 813 according to a first network protocol 814, in particular according to a Datagram Congestion Control Protocol, DCCP, wherein each packet of the first packet data (813) comprises a first packet header (815). The first packet data 813 can be generated internally or received from an external device. The first packet data conversion device 810 comprises a converter 812 configured to convert the first packet data 813 into second packet data 817. The conversion is based on rearranging contents of the first packet header 815, wherein the rearranged first packet header 819 indicates that the second packet data 817 is generated according to a second network protocol 818, in particular according to a User Datagram Protocol, UDP, e.g. as described above with respect to Figures 3 to 7 referred to as second solution based on packet header content rearrangement.

The conversion is done such that the length of first packet header 815 and rearranged first packet header 817 that is referred to as second packet header stays the same. I.e., a length of the rearranged first packet header 819 is equal to a length of the first packet header 815. Besides, a length of a packet of the second packet data 817 may be equal to a length of a corresponding packet of the first packet data 813, which means that the length of the whole packet is not changed by the conversion of the converter 812.

The conversion may further be based on changing a protocol field 501 of the first packet header 815 from indicating the first network protocol 814 to indicating the second network protocol 818, e.g. as described above with respect to Figures 5 to 7. The first packet header 815 may comprise an Internet Protocol (IP) header, e.g. an IP header 703 as described above with respect to Fig. 7. The protocol field can be a protocol field 501 of the IP header 703 as described above with respect to Fig. 5. The IP header may be an IPv4 header or an IPv6 header.

The converter 812 may be configured to change the protocol field 501 from a value of 33 indicating the DCCP protocol to a value of 17 indicating the UDP protocol (or alternatively 136 indicating the UDP-Lite protocol), e.g. as described above with respect to Figures 5 to 7.

The conversion may further be based on extending a checksum coverage field, CsCov, e.g. a CsCov 301 as described above with respect to Fig. 3, of a respective packet of the first packet data 813. The CsCov 301 indicates a range of the respective packet which is covered by a checksum 302, e.g. as shown in Fig. 3. The CsCov 301 may be extended to a size of a length field 401 comprised in a packet header 400 of the second network protocol, in particular a packet header of the UDP protocol, e.g. as shown in Fig. 4. For example, the CsCov 301 may be extended from 4 bit to 16 bit as shown in Fig. 6.

The conversion may further be based on rearranging at least one of a type field 305, 605, a CCVal field 306, 606 and a data offset field 307, 607 of the first packet header 815 to another position in the rearranged first packet header 819, e.g. as shown in Figures 3 and 6.

The rearranged first packet header 819 may include the following data fields, in particular in the following sequence: a source port 601, a destination port 602, a checksum coverage, CsCov 603, a checksum 604, a type 605, a CCVal 606, a data offset 607 and a sequence number 608, 609, e.g. as shown in Fig. 6.

The first packet data 813 and the second packet data 817 can be segmented data streams 111, 113 according to an OSI layer 4 representation, e.g. as shown in Figures 1 and 7.

At receiver side (right-hand side), a second packet data conversion device 820 is used to reconvert the received packet data, referred to as second packet data 817 to its original format. The second packet data conversion device 820 includes a data interface 821 configured to receive second packet data 817 according to a second network protocol 818, in particular according to a User Datagram Protocol, UDP. Each packet of the second packet data 817 comprises a second packet header 819 and data 816. The second packet data conversion device 820 further includes a converter 822 (also denoted as a reconverter since it performs the reverse operation of the converter 812 on sender side). The converter 822 is configured to convert the second packet data 817 into first packet data 813. The conversion is based on rearranging contents of the second packet header 819, wherein the rearranged second packet header 815 indicates that the first packet data 813 is generated according to a first network protocol 814, in particular according to a Datagram Congestion Control Protocol, DCCP.

The transmission system 800 may further comprise a sender (not shown, but located at the left-hand side) which is configured to send first packet data 813 over a transmission channel. The sender comprises the first packet data conversion device 810 as described above, which is configured to convert the first packet data 813 into second packet data 817 before sending. The transmission system 800 may further comprise a receiver (not shown, but located at the right-hand side) which is configured to receive second packet data 817 via the transmission channel. The receiver comprises the second packet data conversion device 820 as described above, which is configured to convert the second packet data 817 into first packet data 813 after reception.

The sender may for example be implemented in a mobile device such as a Smartphone, etc., e.g. a Smartphone application, and the receiver may for example be implemented in a server, e.g. a Download server. The sender may trigger the Download server to download data. In an exemplary implementation, both, sender and receiver, can be implemented in the Smartphone and both, sender and receiver, can be implemented in the Download server to implement duplex transmission.

In another example, the sender may be implemented in a first network node and the receiver may be implemented in a second network node, e.g. gateways or routers in the Internet. In this example data traffic according to a first network protocol can be converted into data traffic according to a second network protocol, in order to pass a transmission path between the first network node and the second network node that is designed to transmit data traffic according to the second network protocol but to block data traffic according to the first network protocol. In this example, the first network node and the second network node may include both, sender and receiver as well for duplex transmission.

In an exemplary implementation, the transmission system 800 includes a middle box between the sender and the receiver, wherein the middle box is configured to transfer data according to the second network protocol 818, in particular according to the UDP protocol and block data according to the first network protocol 814, in particular according to a DCCP protocol. In this implementation, data traffic can pass the middle box due to the data traffic conversion.

Fig. 9 shows a schematic diagram illustrating a method 900 for packet data conversion according to the disclosure.

The method 900 includes: providing 901 first packet data 813 according to a first network protocol 814, in particular according to a Datagram Congestion Control Protocol, DCCP, e.g. as described above with respect to Fig. 8, wherein each packet of the first packet data 813 comprises a first packet header 815.

The method 900 further includes converting 902 the first packet data 813 into second packet data 817, e.g. as described above with respect to Fig. 8, wherein the converting is

based on rearranging contents of the first packet header 815, wherein the rearranged first packet header 819 indicates that the second packet data is generated according to a second network protocol 818, in particular according to a User Datagram Protocol, UDP.

The method 900 may include further steps, such as, for example, according to the computing blocks described above with reference to Figures 6 to 8, in particular as described above with respect to the converter device 810 of Fig. 8.

Another aspect of the invention is related to a computer program product comprising program code for performing the method 900 or the functionalities described above, when executed on a computer or a processor. The method 900 may be implemented as program code that may be stored on a non-transitory computer medium. The computer program product may implement the techniques described above with respect to Figures 6 to 9.

While a particular feature or aspect of the disclosure may have been disclosed with respect to only one of several implementations or embodiments, such feature or aspect may be combined with one or more other features or aspects of the other implementations or embodiments as may be desired and advantageous for any given or particular application. Furthermore, to the extent that the terms "include", "have", "with", or other variants thereof are used in either the detailed description or the claims, such terms are intended to be inclusive in a manner similar to the term "comprise". Also, the terms "exemplary", "for example" and "e.g." are merely meant as an example, rather than the best or optimal. The terms “coupled” and “connected”, along with derivatives may have been used. It should be understood that these terms may have been used to indicate that two elements cooperate or interact with each other regardless whether they are in direct physical or electrical contact, or they are not in direct contact with each other.

Although specific aspects have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that a variety of alternate and/or equivalent implementations may be substituted for the specific aspects shown and described without departing from the scope of the present disclosure. This application is intended to cover any adaptations or variations of the specific aspects discussed herein.

Although the elements in the following claims are recited in a particular sequence, unless the claim recitations otherwise imply a particular sequence for implementing some or all of those elements, those elements are not necessarily intended to be limited to being implemented in that particular sequence.

Many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the above teachings. Of course, those skilled in the art readily recognize that there are numerous applications of the invention beyond those described herein. While the present invention has been described with reference to one or more particular embodiments, those skilled in the art recognize that many changes may be made thereto without departing from the scope of the present invention. It is therefore to be understood that within the scope of the appended claims and their equivalents, the invention may be practiced otherwise than as specifically described herein.

CLAIMS

1. A first packet data conversion device (810) for a sender, the packet data conversion device (810) comprising:

a data interface (811) configured to provide first packet data (813) according to a first network protocol (814), in particular according to a Datagram Congestion Control Protocol, DCCP, wherein each packet of the first packet data (813) comprises a first packet header (815); and

a converter (812) configured to convert the first packet data (813) into second packet data (817),

wherein the conversion is based on rearranging contents of the first packet header (815), wherein the rearranged first packet header (819) indicates that the second packet data (817) is generated according to a second network protocol (818), in particular according to a User Datagram Protocol, UDP or a UDP-Lite protocol.

2. The first packet data conversion device (810) of claim 1,

wherein a length of the rearranged first packet header (819) is equal to a length of the first packet header (815).

3. The first packet data conversion device (810) of claim 1 or 2,

wherein a length of a packet of the second packet data (817) is equal to a length of a corresponding packet of the first packet data (813).

4. The first packet data conversion device (810) of one of the preceding claims,

wherein the conversion is further based on changing a protocol field (501) of the first packet header (815, 703) from indicating the first network protocol (814) to indicating the second network protocol (818).

5. The first packet data conversion device (810) of claim 4,

wherein the first packet header (815) comprises an Internet Protocol, IP, header (703), and

wherein the protocol field (501) is a protocol field of the IP header (703).

6. The first packet data conversion device (810) of claim 4 or 5,
wherein the converter (812) is configured to change the protocol field (501) from a value of 33 indicating the DCCP protocol to a value of 17 indicating the UDP protocol or to a value of 136 indicating the UDP-Lite protocol.

7. The first packet data conversion device (810) of one of the preceding claims,
wherein the conversion is further based on extending a checksum coverage field, CsCov (301), of a respective packet of the first packet data (813), wherein the CsCov (301) indicates a range of the respective packet which is covered by a checksum (302).

8. The first packet data conversion device (810) of claim 7,
wherein the CsCov (301) is extended to a size of a length field (401) comprised in a packet header (400) of the second network protocol, in particular a packet header of the UDP protocol or a packet header of the UDP-Lite protocol.

9. The first packet data conversion device (810) of one of the preceding claims,
wherein the conversion is further based on rearranging at least one of a type field (305, 605) a CCVal field (306, 606) and a data offset field (307, 607) of the first packet header (815) to another position in the rearranged first packet header (819).

10. The first packet data conversion device (810) of one of the preceding claims,
wherein the rearranged first packet header (819) comprises the following data fields, in particular in the following sequence: a source port (601), a destination port (602), a checksum coverage, CsCov (603), a checksum (604), a type (605), a CCVal (606), a data offset (607) and a sequence number (608, 609).

11. The first packet data conversion device (810) of one of the preceding claims,
wherein the first packet data (813) and the second packet data (817) are segmented data streams according to an OSI layer 4 representation.

12. A second packet data conversion device (820) for a receiver, the second packet data conversion device (820) comprising:
a data interface (821) configured to receive second packet data (817) according to a second network protocol (818), in particular according to a User Datagram Protocol,

UDP, or according to a UDP-Lite protocol, wherein each packet of the second packet data (817) comprises a second packet header (819); and

a converter (822) configured to convert the second packet data (817) into first packet data (813),

wherein the conversion is based on rearranging contents of the second packet header (819), wherein the rearranged second packet header (815) indicates that the first packet data (813) is generated according to a first network protocol (814), in particular according to a Datagram Congestion Control Protocol, DCCP.

13. A transmission system (800), comprising:

a sender configured to send first packet data (813) over a transmission channel, wherein the sender comprises a first packet data conversion device (810) according to one of claims 1 to 11, which is configured to convert the first packet data (813) into second packet data (817) before sending; and

a receiver configured to receive second packet data (817) via the transmission channel, wherein the receiver comprises a second packet data conversion device (820) according to claim 12, which is configured to convert the second packet data (817) into first packet data (813) after reception.

14. The transmission system (800) of claim 13, comprising:

a middle box between the sender and the receiver, wherein the middle box is configured to transfer data according to the second network protocol (818), in particular according to the UDP protocol or according to the UDP-Lite protocol and block data according to the first network protocol (814), in particular according to the DCCP protocol.

15. A method (900) for packet data conversion, the method comprising:

providing (901) first packet data according to a first network protocol, in particular according to a Datagram Congestion Control Protocol, DCCP, wherein each packet of the first packet data comprises a first packet header; and

converting (902) the first packet data into second packet data,

wherein the converting is based on rearranging contents of the first packet header, wherein the rearranged first packet header indicates that the second packet data is generated according to a second network protocol, in particular according to a User Datagram Protocol, UDP, or according to a UDP-Lite protocol.

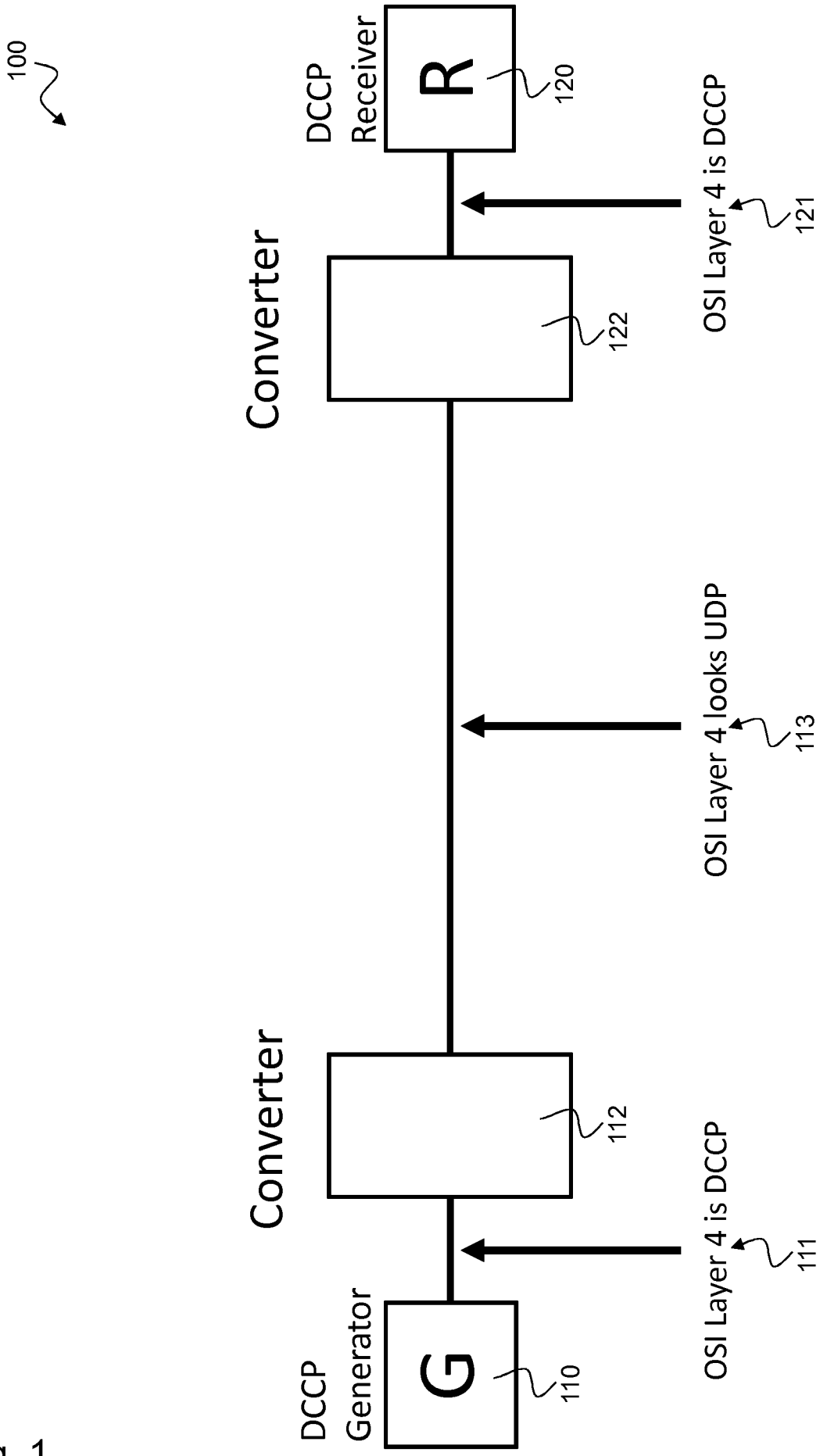
100
DCCP Receiver
R
120
Converter
122
OSI Layer 4 is DCCP
121
OSI Layer 4 looks UDP
113
Converter
112
OSI Layer 4 is DCCP
111
DCCP Generator
G
110

Fig. 1

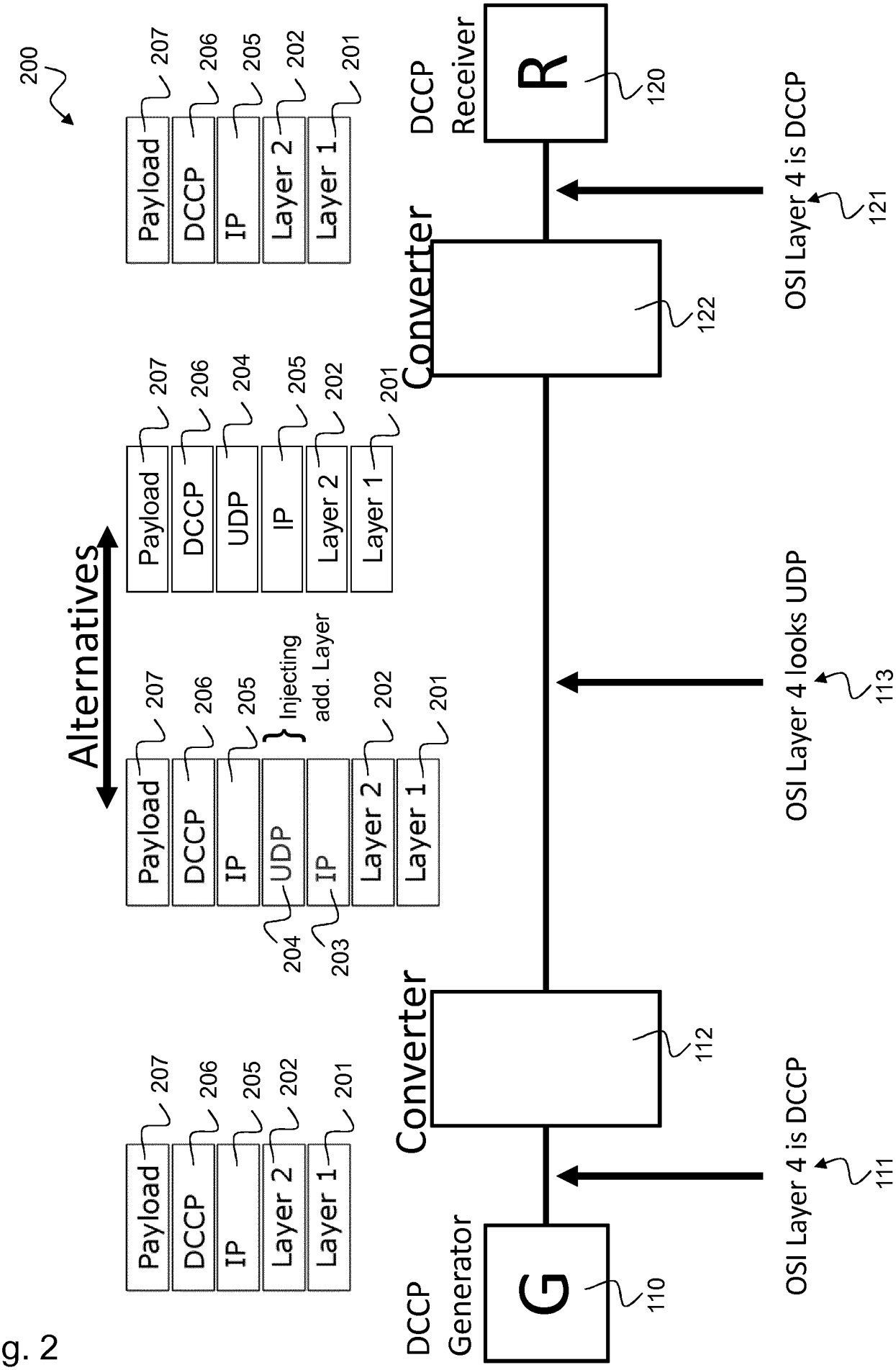
200
Payload
DCCP
IP
Layer 2
Layer 1
207
206
205
202
201
DCCP Receiver
R
120
Converter
122
OSI Layer 4 is DCCP
121
Alternatives
Payload
DCCP
UDP
IP
Layer 2
Layer 1
207
206
204
205
202
201
Payload
DCCP
IP
UDP
IP
Layer 2
Layer 1
207
206
205
Injecting add. Layer
202
201
204
203
OSI Layer 4 looks UDP
113
Converter
112
DCCP Generator
G
110
OSI Layer 4 is DCCP
111

Fig. 2

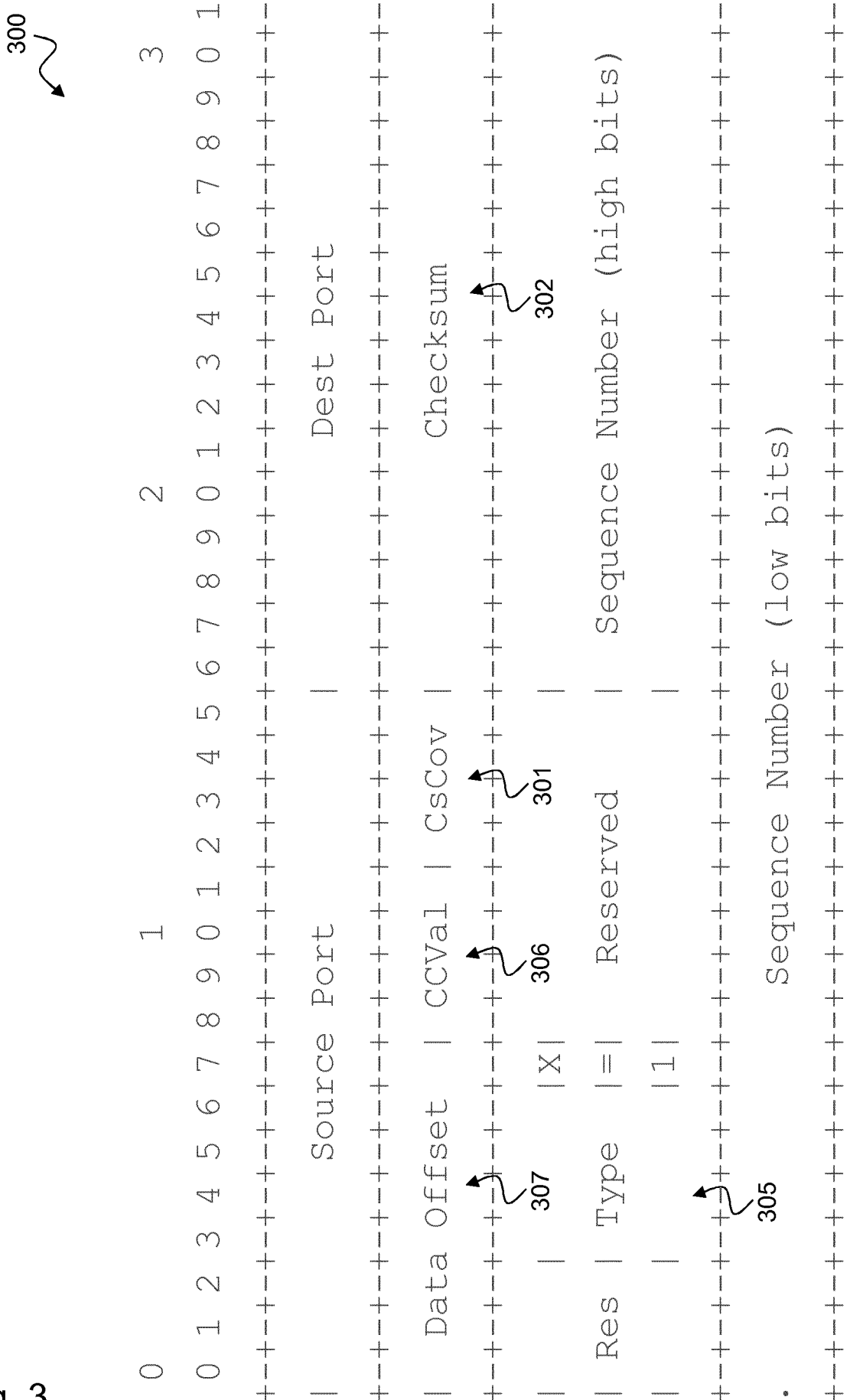
300
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Source Port
Dest Port
Data Offset
CCVal
CsCov
Checksum
302
301
306
307
Res
Type
=
Reserved
Sequence Number (high bits)
X
1
305
Sequence Number (low bits)

Fig. 3

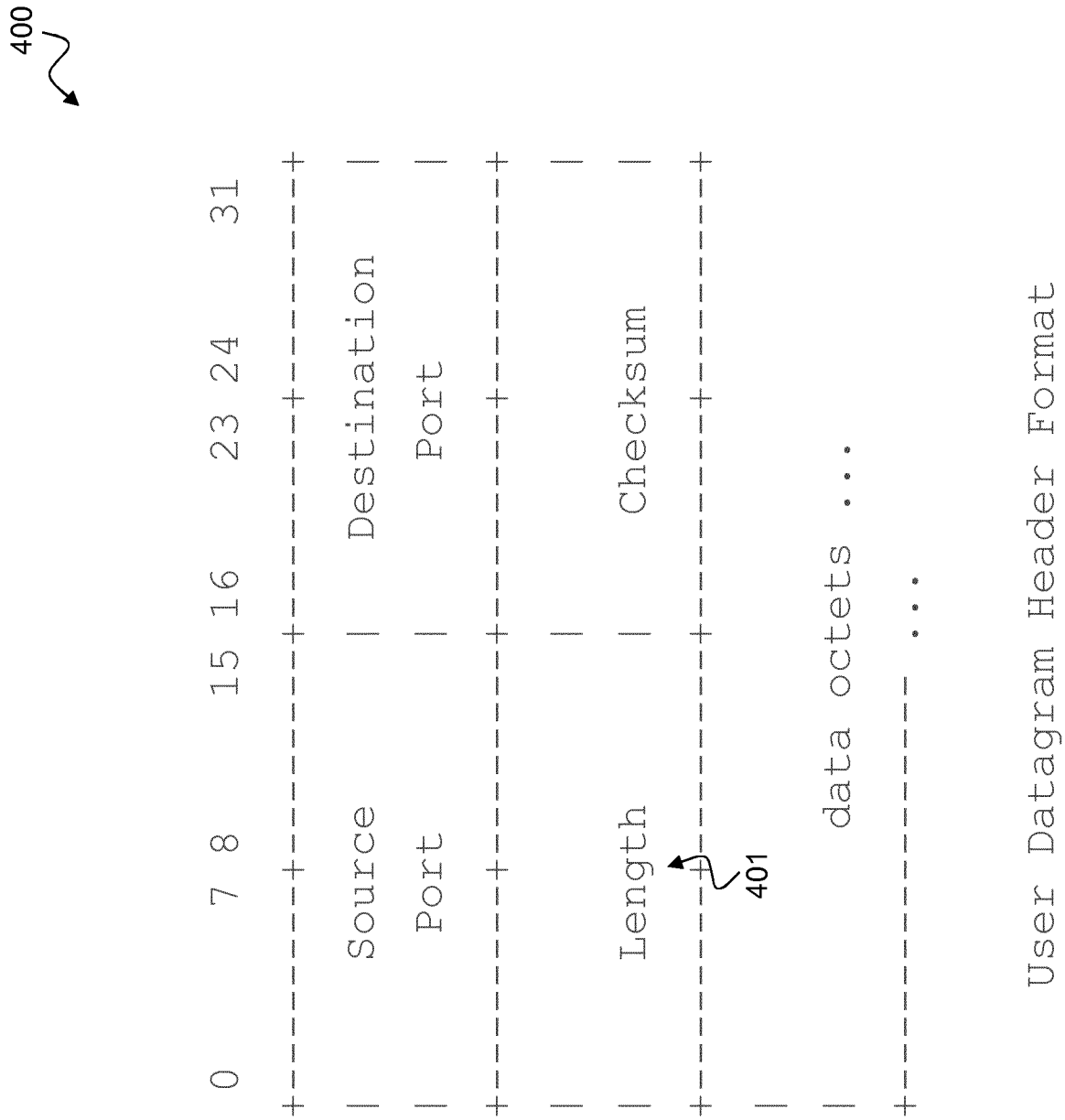
400
0 7 8 15 16 23 24 31
Source Port
Destination Port
Length
Checksum
401
data octets ...
User Datagram Header Format

Fig. 4

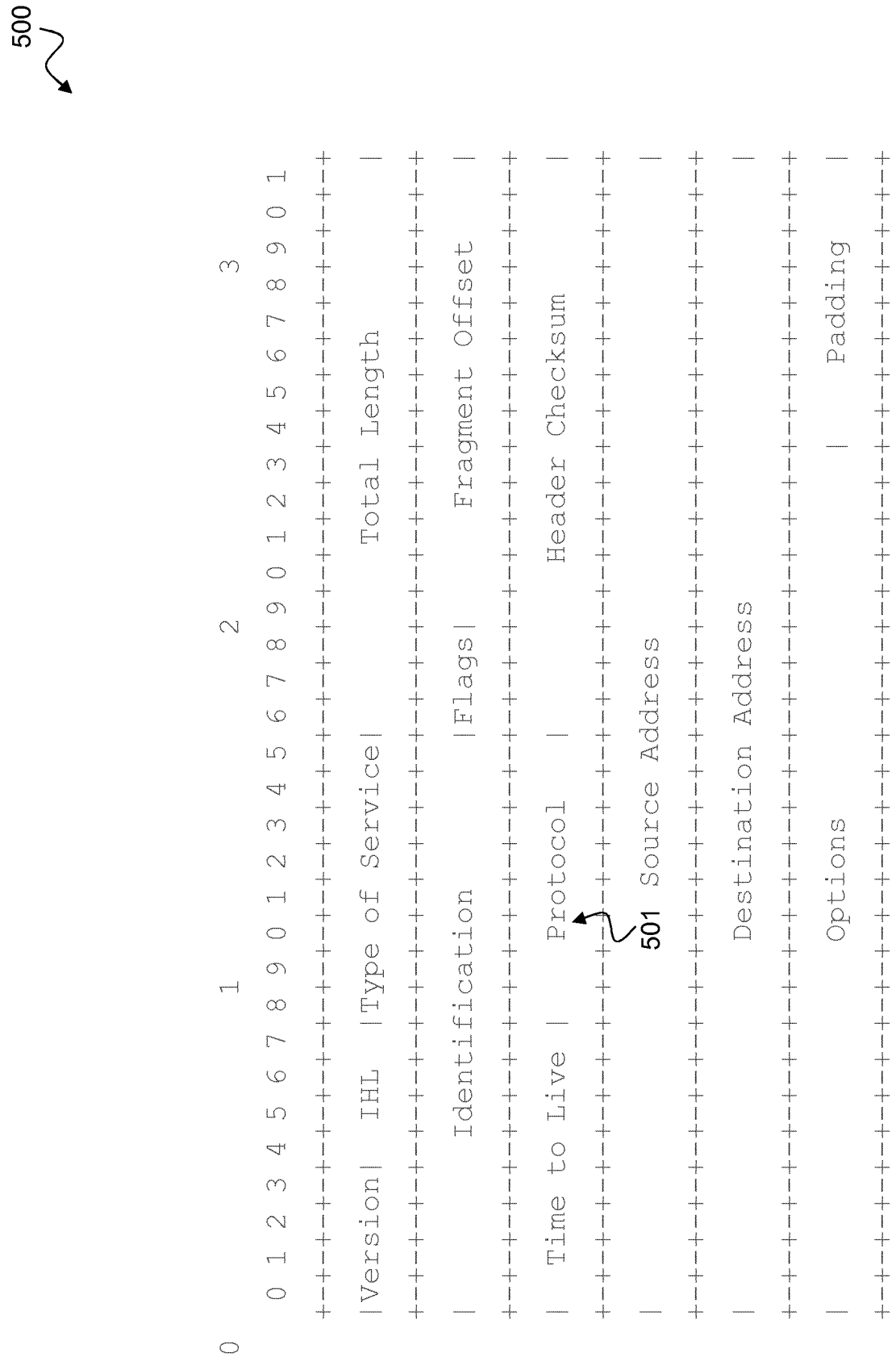
500
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version	IHL	Type of Service	Total Length
Identification	Flags	Fragment Offset
Time to Live	Protocol	Header Checksum
501
Source Address
Destination Address
Options	Padding

Fig. 5

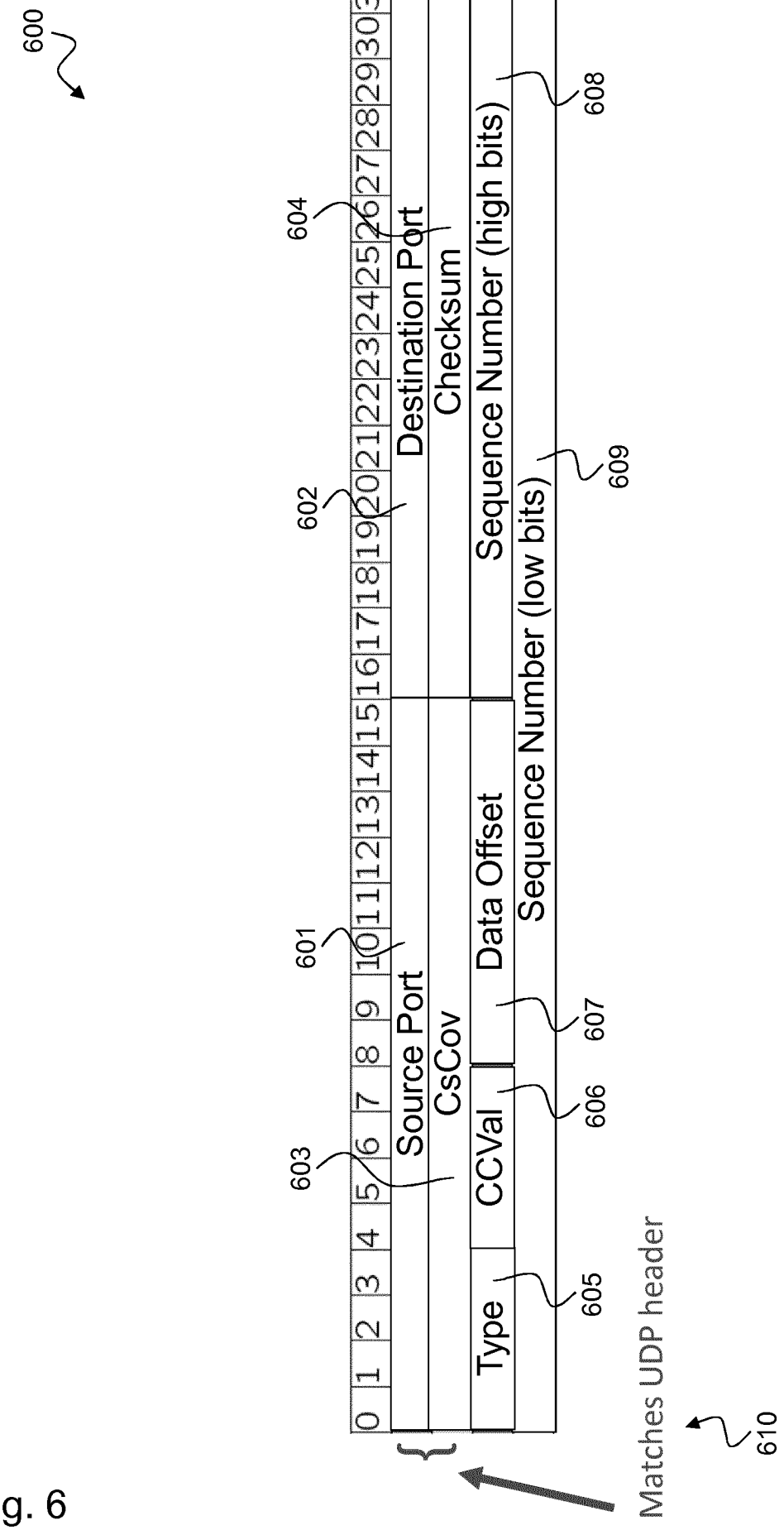
600
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Source Port
Destination Port
CsCov
Checksum
Type
CCVal
Data Offset
Sequence Number (high bits)
Sequence Number (low bits)
601
602
603
604
605
606
607
608
609
610
Matches UDP header

Fig. 6

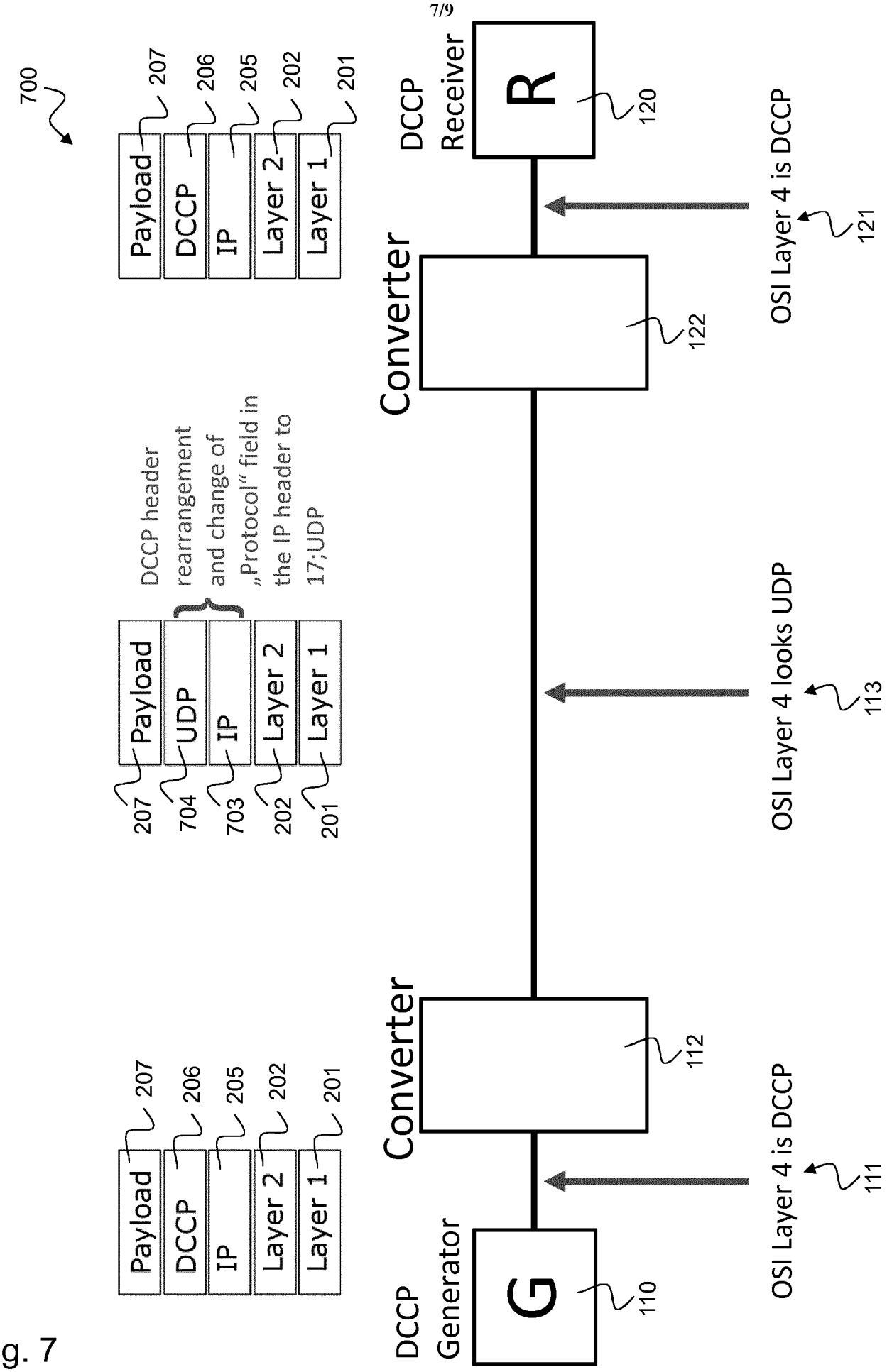
700
Payload
DCCP
IP
Layer 2
Layer 1
207
206
205
202
201
DCCP header rearrangement and change of „Protocol“ field in the IP header to 17;UDP
Payload
UDP
IP
Layer 2
Layer 1
207
704
703
202
201
DCCP Generator
G
110
Converter
112
OSI Layer 4 is DCCP
111
OSI Layer 4 looks UDP
113
Converter
122
OSI Layer 4 is DCCP
121
DCCP Receiver
R
120

Fig. 7

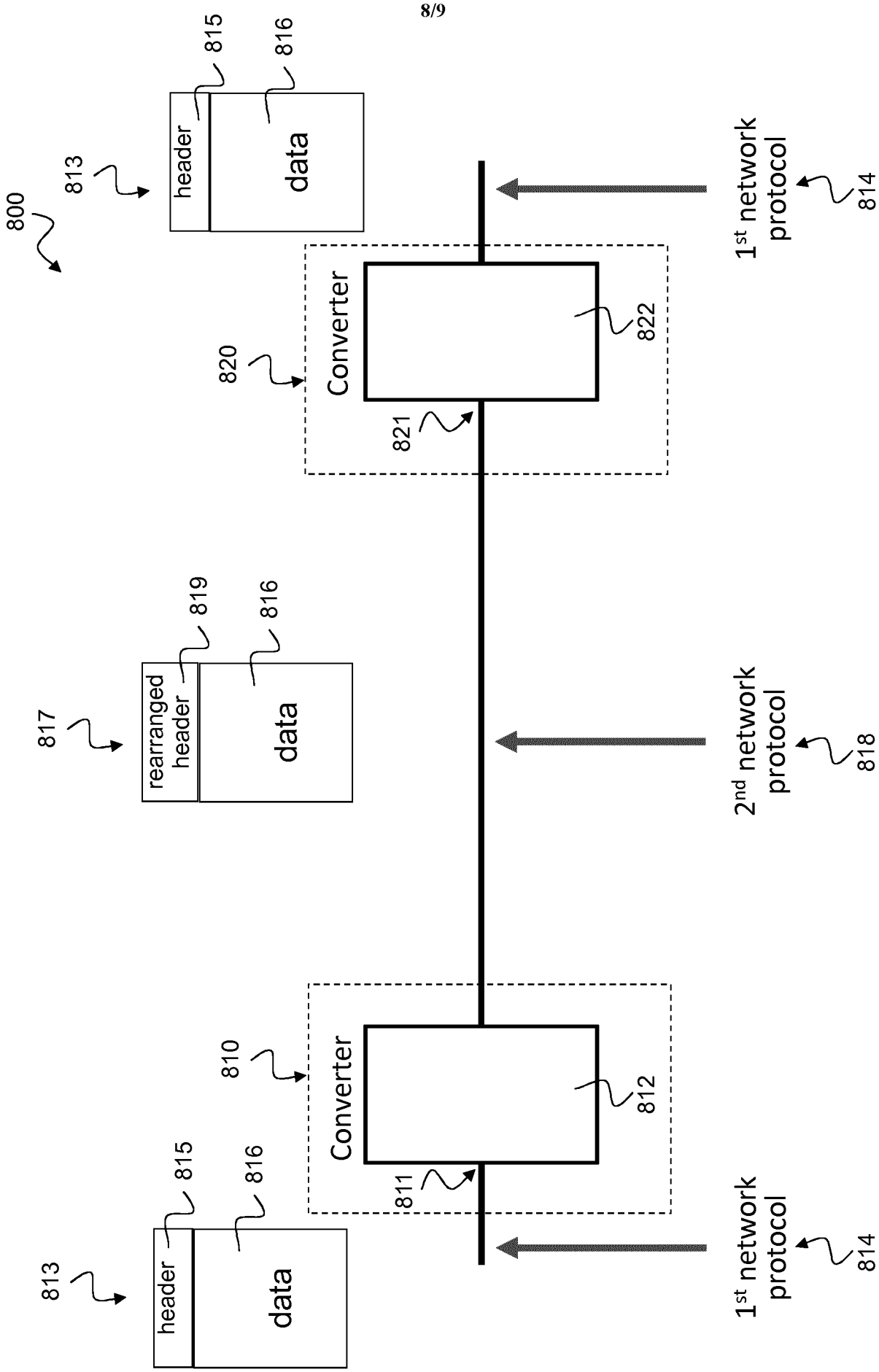
800
813
header
815
data
816
1st network protocol
814
820
Converter
822
821
817
rearranged header
819
data
816
2nd network protocol
818
810
Converter
812
811
813
header
815
data
816
1st network protocol
814

Fig. 8

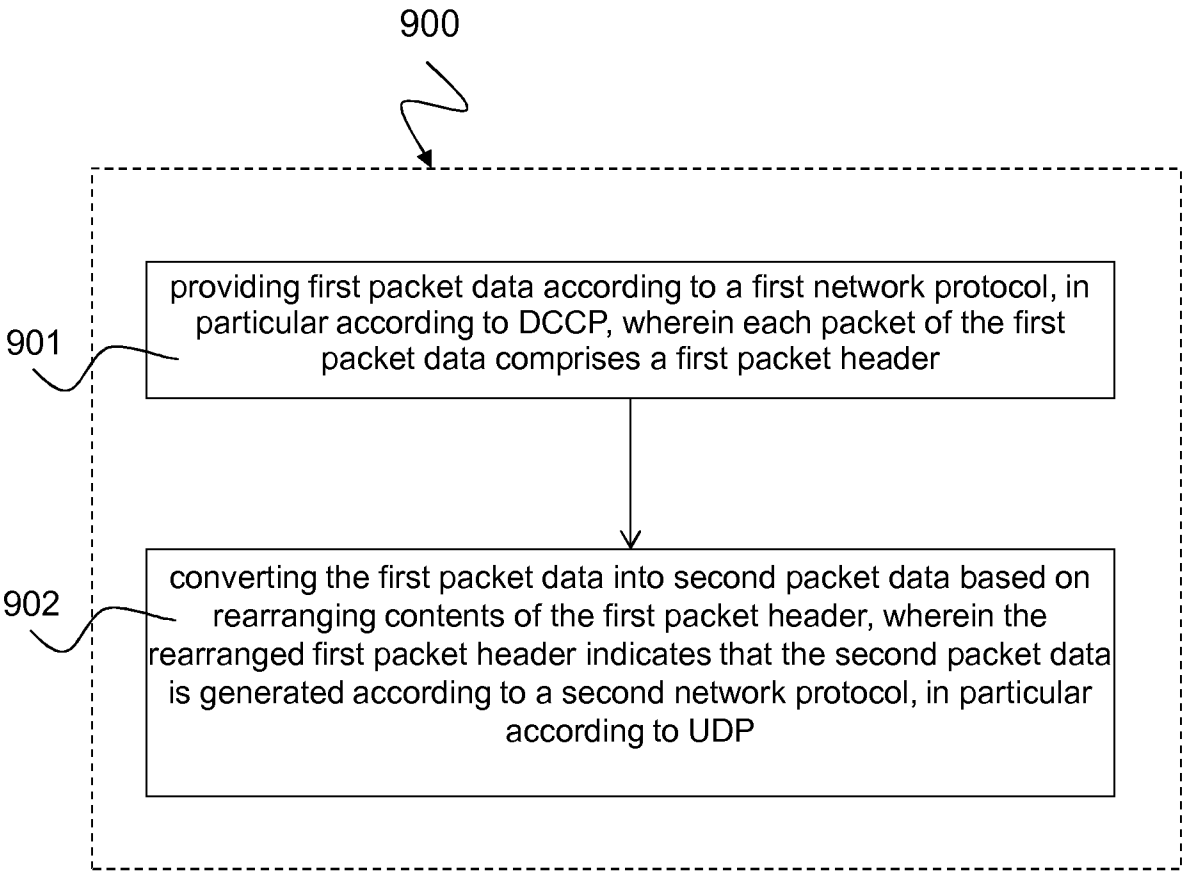
900
providing first packet data according to a first network protocol, in particular according to DCCP, wherein each packet of the first packet data comprises a first packet header
901
converting the first packet data into second packet data based on rearranging contents of the first packet header, wherein the rearranged first packet header indicates that the second packet data is generated according to a second network protocol, in particular according to UDP
902

Fig. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2019/054265

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06 H04L29/08
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, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PHELAN SONUS NETWORKS T: "Datagram Congestion Control Protocol (DCCP) Encapsulation for NAT Traversal (DCCP-NAT); draft-phelan-dccp-natencap-00.txt", DATAGRAM CONGESTION CONTROL PROTOCOL (DCCP) ENCAPSULATION FOR NAT TRAVERSAL (DCCP-NAT); DRAFT-PHELAN-DCCP-NATENCAP-00.TXT, INTERNET ENGINEERING TASK FORCE, IETF; STANDARDWORKINGDRAFT, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH- 1205 GENEVA, SWITZ, 14 February 2008 (2008-02-14), XP015054476, abstract paragraphs [003.] - [3.2.1] ----- -/--	1-14

[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
20 March 2019	27/03/2019

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	Kesting, Christiane

2

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

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PHELAN SONUS G FAIRHURST UNIVERSITY OF ABERDEEN C PERKINS UNIVERSITY OF GLASGOW T: "DCCP-UDP: A Datagram Congestion Control Protocol UDP Encapsulation for NAT Traversal; rfc6773.txt", DCCP-UDP: A DATAGRAM CONGESTION CONTROL PROTOCOL UDP ENCAPSULATION FOR NAT TRAVERSAL; RFC6773.TXT, INTERNET ENGINEERING TASK FORCE, IETF; STANDARD, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH- 1205 GENEVA, SWITZERLAND, 14 November 2012 (2012-11-14), pages 1-20, XP015086487, paragraphs [003.] - [03.3] abstract -----	1-14
X	US 2013/279409 A1 (DUBLIN III WILBUR L [US] ET AL) 24 October 2013 (2013-10-24) paragraphs [0158], [0162]; figure 10c -----	1-3,11, 14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2019/054265

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013279409 A1	24-10-2013	EP 2839612 A1	25-02-2015
		EP 2839697 A1	25-02-2015
		US 2013279409 A1	24-10-2013
		US 2013279410 A1	24-10-2013
		WO 2013158589 A1	24-10-2013
		WO 2013158591 A1	24-10-2013