



(51) International Patent Classification:

H04L 29/08 (2006.01) H04W 28/06 (2009.01)

(21) International Application Number:

PCT/EP2016/067650

(22) International Filing Date:

25 July 2016 (25.07.2016)

(25) Filing Language:

English

(26) Publication Language:

English

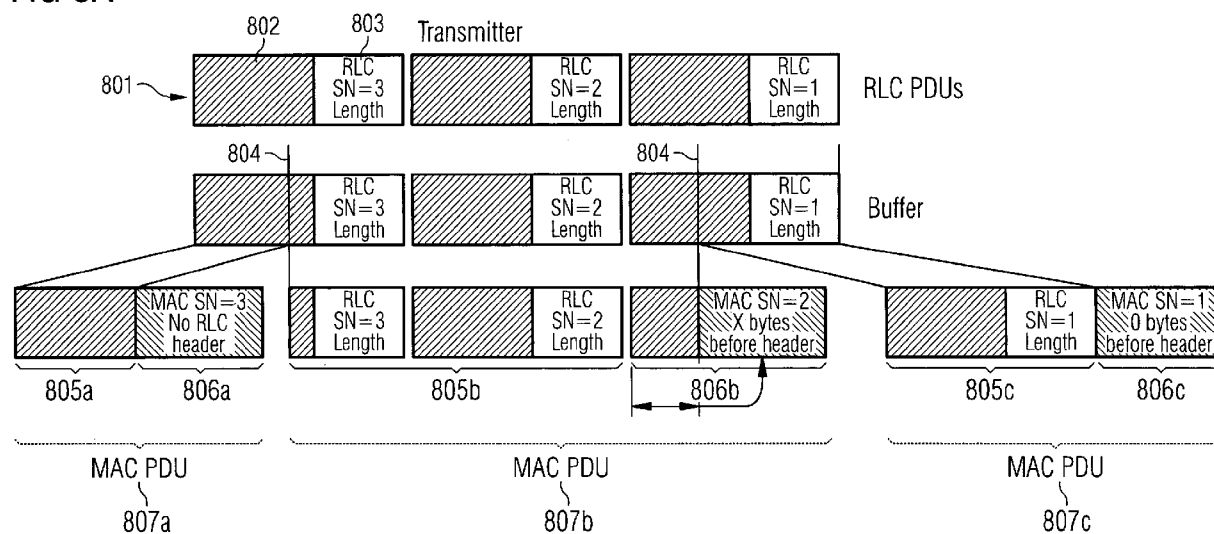
(71) Applicant: NOKIA SOLUTIONS AND NETWORKS
OY [FI/FI]; Karaportti 3, 02610 Espoo (FI).(72) Inventors: DECARREAU, Guillaume; Sebastian Bauer
Str. 16e, 81737 Munich (DE). MAEDER, Andreas; Erthal-
str. 46, 97074 Würzburg (DE). SHARMA, Amit Kumar;
Legnicka 46/48, 53-674 Wrocław (PL).(81) Designated States (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: EFFICIENT LAYER 2 FOR 5G NETWORK

FIG 8A



(57) **Abstract:** A method comprising: providing at least one data packet according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack; segmenting, at the second protocol layer, the at least one data packet according to the first protocol into at least one data portion; processing the at least one data portion at the second protocol layer to produce at least one data packet according to the second protocol; and providing the at least one data packet according to the second protocol from the second protocol layer to a third protocol layer of the protocol stack.

Title

EFFICIENT LAYER 2 FOR 5G NETWORK

Field

The present application relates to a method, apparatus, system and computer program and in particular but not exclusively to a method and apparatus for use in a layer 2 of a communication network.

Background

A communication system can be seen as a facility that enables communication sessions between two or more entities such as user terminals, base stations/access points and/or other nodes by providing carriers between the various entities involved in the communications path. A communication system can be provided for example by means of a communication network and one or more compatible communication devices. The communication sessions may comprise, for example, communication of data for carrying communications such as voice, electronic mail (email), text message, multimedia and/or content data and so on. Non-limiting examples of services provided comprise two-way or multi-way calls, data communication or multimedia services and access to a data network system, such as the Internet.

In a wireless communication system at least a part of a communication session between at least two stations occurs over a wireless link.

A user can access the communication system by means of an appropriate communication device or terminal. A communication device of a user is often referred to as user equipment (UE). A communication device is provided with an appropriate signal receiving and transmitting apparatus for enabling communications, for example enabling access to a communication network or communications directly with other users. The communication device may access a carrier provided by a station or access point, and transmit and/or receive communications on the carrier.

The communication system and associated devices typically operate in accordance with a given standard or specification which sets out what the various entities associated with the system are permitted to do and how that should be achieved. Communication protocols and/or parameters which shall be used for the connection are also typically defined. One

example of a communications system is UTRAN (3G radio). Other examples of communication systems are the long-term evolution (LTE) of the Universal Mobile Telecommunications System (UMTS) radio-access technology and so-called 5G or New Radio (the term used by 3GPP) networks. Standardization of 5G or New Radio networks is currently under discussion. LTE is being standardized by the 3rd Generation Partnership Project (3GPP).

One of the requirements that must be taken into consideration in the development of the new radio networks is the need to support the very high throughput required for 5G.

Summary of Invention

According to a first aspect, there is provided a method comprising: providing at least one data packet according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack; segmenting, at the second protocol layer, the at least one data packet according to the first protocol into at least one data portion; processing the at least one data portion at the second protocol layer to produce at least one data packet according to the second protocol; and providing the at least one data packet according to the second protocol from the second protocol layer to a third protocol layer of the protocol stack.

In one embodiment, the step of processing the at least one data portion at the second protocol layer comprises affixing a header according to the second protocol to each data portion.

In one embodiment, the step of processing at the second protocol layer further comprises for each data portion: determining that the data portion comprises one or more headers according to the first protocol; and in response to the determination, indicating in the header according to the second protocol, the number of bits of the data packet that precede a first header according to the first protocol in the data portion.

In one embodiment, at least one of the data portions comprises partial data packets according to the first protocol.

In one embodiment, at least one of the data portions comprises the concatenation of at least one of data packets and partial data packets according to the first protocol.

In one embodiment, each data portion forms a payload of a data packet according to the second protocol.

In one embodiment, the at least one data packets according to the first protocol include one or more data packets comprising data of a first logical channel and one or more data packets comprising data of a second logical channel.

In one embodiment, at least one of the data packets according to the second protocol comprises data of both the first logical channel and data of the second logical channel.

In one embodiment, at least one of the data packets according to the second protocol comprise data of only one of: the first logical channel and the second logical channel.

In one embodiment, the step of providing at least one of the data packets according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack comprises: storing the at least one data packets according to a first protocol in one or more buffers that are accessible to the second protocol layer.

In one embodiment, the at least one data packets according to the first protocol include one or more data packets comprising data of a first logical channel and one or more data packets comprising data of a second logical channel, and the step of storing the plurality of data packets according to the first protocol in one or more buffers comprises: storing the one or more data packets comprising data of the first logical channel in a first buffer of the one or more buffers; and storing the one or more data packets comprising data of the second logical channel in a second buffer of the one or more buffers.

In one embodiment, the method further comprises: providing copies of the at least one data portions from the second protocol layer to the first protocol layer of the protocol stack; and transmitting a data packet according to the second protocol comprising the copies of the at least one data portions if a message indicating that the transmission of that data packet has failed is received.

In one embodiment, the method further comprises: providing an indication that the transmission of a data portion has failed from the second protocol layer to the first protocol layer; determining a data packet according to the first protocol that was at least partially included in that data portion; and transmitting that data packet to a receiving entity.

In one embodiment, the method further comprises: performing ARQ error correction procedures to each data packet according to the first protocol at the first protocol layer of a protocol stack.

According to a second aspect, there is provided a computer program product for a computer, comprising software code portions for performing the steps of the first aspect when the product is run on the computer.

According to a third aspect, there is provided an apparatus comprising: at least one processor and at least one memory including a computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to: provide at least one data packet according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack; segment, at the second protocol layer, the at least one data packet according to the first protocol into at least one data portion; process the at least one data portion at the second protocol layer to produce at least one data packet according to the second protocol; and provide the at least one data packet according to the second protocol from the second protocol layer to a third protocol layer of the protocol stack.

According to a fourth aspect, there is provided a method comprising: receiving at least one data packet according to the second protocol at a second layer of a protocol stack from a third layer of the protocol stack; processing the at least one data packet according to the second protocol at the second protocol layer to produce at least one data portion; and concatenating, at the second protocol layer, the at least one data portion to produce at least one data packet according to a first protocol; and providing the at least one data packet according to the first protocol from the second protocol layer to a first protocol layer of the protocol stack.

In one embodiment, one of the data packets according to the second protocol comprises one or more data packets of a first protocol, wherein a data packet of the first protocol located at a start of the one of the data packets of the second protocol is a first incomplete data packet.

In one embodiment, the method further comprises: loading from a memory, a second incomplete data packet of the first protocol; and combining the first incomplete data packet with the second incomplete data packet to form a complete data packet according to the first protocol.

In one embodiment, the method further comprises: determining that a memory does not contain a second incomplete data packet of the first protocol that is configured to combine with the first incomplete data packet to form a complete data packet; and in response to the determination, deleting the first incomplete data packet.

In one embodiment, the method further comprises, prior to receiving the at least one data packet according to the second protocol at the second layer of the protocol stack: receiving a second data packet according to the second protocol at the second layer of a protocol stack from the third layer of the protocol stack, wherein the second data packet according to the second protocol comprises the second incomplete data packet, wherein the second incomplete data packet is located at an end of the second data packet according to the second protocol; and storing the second incomplete data packet in a memory.

In one embodiment, the data packet according to the second protocol comprises one of the at least one data packets according to the second protocol comprises one or more data packets of a first protocol, wherein a data packet of the first protocol located at an end of the one of the at least one data packets of the second protocol is a second incomplete data packet, the method comprising: storing the second incomplete data packet in a memory; and if a predetermined time period elapses, without a first incomplete data packet being received, deleting the second incomplete data packet, wherein the first incomplete data packet is configured to combine with the first incomplete data packet to form a complete data packet.

According to a fifth aspect, there is provided a computer program product for a computer, comprising software code portions for performing the steps of any of claims 17 to 22 when the product is run on the computer.

According to sixth aspect, there is provided an apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to: receive at least one data packet according to the second protocol at a second layer of a protocol stack from a third layer of the protocol stack; process the at least one data packet according to the second protocol at the second protocol layer to produce at least one data portion; concatenate, at the second protocol layer, the at least one data portion to produce at least one data packet according to a first protocol; and provide the at least one data packet according to the first protocol from the second protocol layer to a first protocol layer of the protocol stack.

Description of Figures

Embodiments will now be described, by way of example only, with reference to the accompanying Figures in which:

Figure 1 shows a schematic diagram of an example communication system comprising a plurality of base stations and a plurality of communication devices;

Figure 2 shows a schematic diagram of an example mobile communication device;

Figure 3 shows a schematic diagram of the LTE protocol stack;

Figure 4 illustrates a method typically performed to produce data packets for transmission;

Figure 5 illustrates a method performed to generate data packets for transmission according to embodiments;

Figure 6 illustrates a method performed to generate data packets for transmission according to embodiments;

Figure 7 illustrates a method performed at a receiving entity, following receipt of a data packet;

Figure 8a shows the production of data packets belonging to a second protocol from data packets belonging to a first protocol;

Figure 8b shows data packets of a second protocol;

Figure 9a shows an example of two complete data packets of a first protocol being received at a receiving entity;

Figure 9b shows an example of a complete and an incomplete data packet being received at a receiving entity;

Figure 10 shows an example of data packets being received at the receiving entity;

Figure 11 shows an example of data packets being received and stored at a receiving entity;

Figure 12 shows an example of incomplete data packets stored at the receiving entity being deleted;

Figure 13a illustrates an example of organising data packets belonging to different logical channels;

Figure 13b illustrates an example of organising data packets belonging to different logical channels;

Figure 13c illustrates an example of organising data packets belonging to different logical channels;

Figure 14 shows an example of receiving data packets of different logical channels at the receiving entity; and

Figure 15 shows a schematic diagram of an example control apparatus.

Detailed description

Before explaining in detail the examples, certain general principles of a wireless communication system and mobile communication devices are briefly explained with reference to Figures 1 to 2 to assist in understanding the technology underlying the described examples.

In a wireless communication system 100, such as that shown in figure 1, mobile communication devices or user equipment (UE) 102, 104, 105 are provided wireless access via at least one base station or similar wireless transmitting and/or receiving node or point. A base station is referred to as an eNodeB (eNB) in LTE. Base stations are typically controlled by at least one appropriate controller apparatus, so as to enable operation thereof and management of mobile communication devices in communication with the base stations. The controller apparatus may be located in a radio access network (e.g. wireless communication system 100) or in a core network (CN) (not shown) and may be implemented as one central apparatus or its functionality may be distributed over several apparatus. The controller apparatus may be part of the base station and/or provided by a separate entity such as a Radio Network Controller. In Figure 1 control apparatus 108 and 109 are shown to control the respective macro level base stations 106 and 107. In some systems, the control apparatus may additionally or alternatively be provided in a radio network controller.

LTE systems may however be considered to have a so-called “flat” architecture, without the provision of RNCs; rather the eNB is in communication with a system architecture evolution gateway (SAE-GW) and a mobility management entity (MME), which entities may also be pooled meaning that a plurality of these nodes may serve a plurality (set) of eNBs. Each UE is served by only one MME and/or S-GW at a time and the (e) NB keeps track of current association. SAE-GW is a “high-level” user plane core network element in LTE, which may consist of the S-GW and the P-GW (serving gateway and packet data network gateway, respectively). The functionalities of the S-GW and P-GW are separated and they are not required to be co-located.

In Figure 1 base stations 106 and 107 are shown as connected to a wider communications network 113 via gateway 112. A further gateway function may be provided to connect to another network.

The smaller base stations 116, 118 and 120 may also be connected to the network 113, for example by a separate gateway function and/or via the controllers of the macro level stations. The base stations 116, 118 and 120 may be pico or femto level base stations or the like. In the example, stations 116 and 118 are connected via a gateway 111 whilst station 120 connects via the controller apparatus 108. In some embodiments, the smaller stations may not be provided.

A possible mobile communication device will now be described in more detail with reference to Figure 2 showing a schematic, partially sectioned view of a communication device 200. Such a communication device is often referred to as user equipment (UE) or terminal. An appropriate mobile communication device may be provided by any device capable of sending and receiving radio signals. Non-limiting examples comprise a mobile station (MS) or mobile device such as a mobile phone or what is known as a ‘smart phone’, a computer provided with a wireless interface card or other wireless interface facility (e.g., USB dongle), personal data assistant (PDA) or a tablet provided with wireless communication capabilities, or any combinations of these or the like. A mobile communication device may provide, for example, communication of data for carrying communications such as voice, electronic mail (email), text message, multimedia, and so on. Users may thus be offered and provided numerous services via their communication devices. Non-limiting examples of these services comprise two-way or multi-way calls, data communication or multimedia services or simply an access to a data communications network system, such as the Internet. Users may also be provided broadcast or multicast data. Non-limiting examples of the content comprise downloads, television and radio programs, videos, advertisements, various alerts and other information.

The mobile device 200 may receive signals over an air or radio interface 207 via appropriate apparatus for receiving and may transmit signals via appropriate apparatus for transmitting radio signals. In Figure 2 transceiver apparatus is designated schematically by block 206. The transceiver apparatus 206 may be provided for example by means of a radio part and associated antenna arrangement. The antenna arrangement may be arranged internally or externally to the mobile device.

A mobile device is typically provided with at least one data processing entity 201, at least one memory 202 and other possible components 203 for use in software and hardware aided execution of tasks it is designed to perform, including control of access to and communications with access systems and other communication devices. The data processing, storage and other relevant control apparatus can be provided on an appropriate circuit board and/or in chipsets. This feature is denoted by reference 204. The user may control the operation of the mobile device by means of a suitable user interface such as key pad 205, voice commands, touch sensitive screen or pad, combinations thereof or the like. A display 208, a speaker and a microphone can be also provided. Furthermore, a mobile communication device may comprise appropriate connectors (either wired or wireless) to other devices and/or for connecting external accessories, for example hands-free equipment, thereto. The communication devices 102, 104, 105 may access the communication system based on various access techniques.

Reference is now made to figure 3, which provides an illustration of the UE protocol stack that is part of the LTE standard. As may be seen in the figure, the stack includes 3 different layers through which a data packet passes on the uplink, when being prepared for transmission over the network, and on the downlink, when being received from the network for delivery to the host. Embodiments of the application are directed to a method of optimising the performance of layer 2 of the protocol stack. Such optimisation may help to improve the throughput and processing efficiency of network communications, thus helping to support the requirements of the 5G systems.

Layer 2 of the protocol stack comprises, a first protocol layer 301, a second protocol layer 302, and a zeroth protocol layer 303. In this example, the first protocol layer 301 is a Radio Link Control (RLC) layer 301, the second protocol layer 302 is a Medium Access Control (MAC) layer 302, and the zeroth protocol layer 303 is a Packet Data Convergence Protocol (PDCP) Layer 303. However, it would be understood that these are example only, and that the present

application is not limited in its scope only to the LTE protocol stack, but may find application to other protocol stacks.

During processing in the uplink direction, i.e. from the host to the network, the RLC layer 301, receives data packets from the PDCP 303. The RLC 301 then performs the necessary processing in order to produce data packets which are passed to the MAC layer 302. The data packet passed by the RLC layer 301 to the MAC layer 302 may be referred to as RLC protocol data units (PDU) or MAC service data units (SDU). The term “service data units” refers to packets received by a layer of the stack, whilst packets output by a layer of the stack are referred to as “protocol data units”. Therefore, when the term “RLC PDUs” is used, this may also be understood to refer to a “MAC SDUs”, and vice versa.

The protocol stack may include a third protocol layer 304, which in this example is a physical layer 304. However, it would be understood that the physical layer is just an example, and that the third protocol layer may take forms other than the physical layer 304. When the MAC layer 302 has processed the RLC PDUs 301 to produce the MAC PDUs 302, the MAC layer 302 may then pass the MAC PDUs to the physical layer 304. The physical layer is configured to prepare and transmit the data packets over the network to a receiving entity.

The processing performed at the RLC layer 301 to produce the RLC PDUs / MAC SDUs includes concatenation, wherein an RLC SDU is packed into a size appropriate for transport blocks, which are to be transmitted over the network. The RLC PDU is chosen based on the transport block size for the radio bearer. If the RLC SDU is large, or the available radio data rate is low, the RLC SDU may be split into several RLC PDUs. If the RLC SDU is small, or the available radio data rate is high, several RLC SDUs may be packed into a single PDU.

Therefore, prior to the processing used to produce the MAC SDUs, the RLC layer must be informed of the transport block size, so that the RLC SDU can be packed into a size appropriate for the transport blocks. Typically, the MAC layer 302 informs the RLC layer 301 of the transport block size and then the RLC layer 301 performs the processing functions including segmentation and concatenation to produce the MAC SDUs in accordance with the informed transport block size. The RLC layer 301 additionally performs other processing functions, such as error correction, reordering, and duplicate detection.

In addition to the creation of the MAC SDUs, the RLC layer 301 may be responsible for the Automatic Repeat Query (ARQ) function that uses acknowledgements and timeouts to ensure reliable data transmission. Specifically, if the host sending the data packet receives a negative

acknowledgement (NACK), the data packet is re-transmitted. In the LTE protocol stack, the RLC layer 301, may be configured to store in memory an indication of the RLC PDUs which have been transmitted over the network to a destination. The indication may be used to determine for which RLC PDUs a corresponding ACK is expected to be received in response. If the NACK is received, the RLC PDU is then be retransmitted over the network to the destination.

Therefore, the RLC layer may be responsible for both the RLC PDU / MAC SDU creation and the ARQ function. These function operate on separate time scales, i.e. the RLC PDU / MAC SDU creation is a synchronous function, operating on a Transmission Time Interval (TTI) scale. On the other hand, the ARQ function is an asynchronous function, which operates on a non-TTI scale. A bottleneck may arise in the processing performed by the protocol stack due to both of these functions being configured to operate in sequence. Specifically, the RLC layer 301 is configured to firstly create, based on the transport block size information passed to it from the MAC layer, the packet to be transmitted over the network. The RLC layer is configured to then perform the processing heavy operations to provide the ARQ function. During this operation to provide the ARQ function, the RLC layer 301 must keep a copy stored in memory of the packet to be transmitted over the network. The requirement to maintain this copy in memory, whilst performing the operations to provide the ARQ function may result in a delay in the completion of the packet for transmission and therefore a delay in its transmission over the network. This may produce a bottleneck in the LTE protocol stack. It may be desirable to reduce the processing delay in the different protocol stacks so as to enable the protocol stack to support the requirements of the 5G system, such as high throughput.

Figure 4 illustrates an example of the steps which may be performed in the creation of the data packets, which are passed to the MAC layer.

At S401, one or more RLC SDUs are received from the Packet Data Convergence Protocol (PDCP) layer of the protocol stack, which precedes the RLC layer 301.

At S402, the one or more PDCP PDUs that have been received are added to a transmission buffer.

At S403, an indication of the size to be used for the transport blocks in which the data is to be transmitted over the network is received from the MAC layer 302 at the RLC layer 310. The segmentation and concatenation of the RLC SDUs is then performed using the received transport block size information.

At S404, the operations necessary to provide the ARQ function are performed. This involves making and storing a copy of the data packets in the buffer. Such copies may be used for re-transmission of the data packets, if an ACK is not received within the timeout period after sending the data packet

At S405, an RLC header is added to the data packets in the transmission buffer to complete the processing to produce the RLC PDUs / MAC SDUs.

At S406, the RLC PDUs/ MAC SDUs are then passed to the MAC layer 302. The MAC layer will perform the function such as random access channel scheduling, building headers and transport format selection.

By performing S404 prior to completion of the MAC SDUs, the partially completed MAC SDU data packets must be retained in the transmission buffer, hence delaying the real time operation of MAC SDU creation. This may result in a significant bottleneck in the protocol stack.

Embodiments of the present application will now be described. These may reduce the bottleneck, enabling faster creation and efficient processing of data packets for transmission over the network. As in the example of the typical LTE stack given above in figure 4, the following description describes embodiments involving the LTE stack. In this case, the first protocol layer is an RLC layer, and the second protocol layer is a MAC layer. However, it would be understood by the person of ordinary skill in the art that the invention is not limited to implementation in an LTE stack, using the RLC layer, MAC layer and PDCP layer, but may be applied to different protocol stacks using different protocol layers. Additionally, in the following description the data packets of a first protocol, which are produced by the first protocol layer, are RLC PDUs / MAC SDUs, and the data packets of a second protocol, which are produced by the second protocol layer are MAC PDUs. However, it would be understood by the person skilled in the art that the data packets of the first and second protocols are not limited as such, but could be take other forms. The first and second protocols need not be RLC and MAC protocols but could be any suitable protocols.

Reference is made to figure 5, which shows the steps performed in accordance with embodiments of the application at the transmitting entity configured to process and transmit the data packets over the network. The entity may comprise a user equipment (UE), such as that described above with reference to figure 2.

At S501, the RLC layer receives one or more RLC SDUs from the PDCP layer. These SDU data packets may then be added to a buffer.

At S502, the RLC layer performs protocol processing on the data packets stored in the buffer. This may include but is not limited to error correction, reordering, and duplicate detection of the data packets. The processing includes adding RLC headers to the data packets in the buffer to form RLC PDUs. The RLC layer may then perform ARQ error correction procedures to each RLC PDU.

At S503, the RLC PDUs are passed, unsegmented, from the RLC layer to the MAC layer. This may be performed by the RLC layer passing control of the buffer containing the RLC PDUs to the MAC layer.

At S504, a value for the MAC PDU size is obtained by the MAC layer. Once the MAC PDU size has been determined, the MAC segments the RLC PDUs into a plurality of data portions. Each portion of data comprises at least one part of an RLC PDU. Each portion of data forms a payload for a MAC PDU.

At S506, one or more MAC PDUs are constructed from the data portions. This step comprises adding MAC PDU headers to the data packets.

Reference is made to figure 6, which shows the step of S506 in more detail. This figure shows the steps performed in order to construct the MAC PDU header, which is added to the data packets so as to form the MAC PDU in more detail.

At S601, for each portion of data it is determined whether or not the portion of data contains at least one RLC header. If it is determined that the portion of data does not begin with a RLC PDU header, the method proceeds to S602. Otherwise, the method proceeds to S604.

At S602, it is determined if the data portion begins with an RLC header. If not, the method proceeds to S603. If so, the method proceeds to S604.

At S603, an indication of the number of bits located prior to the first RLC PDU header is added to the MAC PDU header. The header may then be affixed to the portion of data to form the MAC PDU. The method then proceeds to step S604.

At S604, a sequence number may be added to each MAC PDU header in each portion of data. The sequence number provides an indication of the order of the MAC PDUs.

At S605, the MAC PDU header is affixed to the data portion to produce the MAC PDU data packet.

At S606, the MAC layer passes the MAC PDU to the physical layer of the protocol stack. The MAC PDU is then transmitted over the interface to its destination. The MAC layer informs the RLC layer that the transmission of the MAC PDU has taken place. The MAC layer may also provide to the RLC layer, a copy of the portion of data that was used to produce the MAC PDU. In one embodiment, the MAC layer may provide the entire MAC PDU to the RLC layer. By providing the portion of data to the RLC layer, MAC PDU may, if the RLC layer determines that retransmission of the MAC PDU is required, be retransmitted. The RLC layer may determine that the MAC PDU is to be retransmitted if a message indicating that the transmission of the data packet has failed is received from the receiving entity at the transmitting entity. This message may take the form of a negative ACK (NACK), which is received from the destination to which the MAC PDU was sent.

In one embodiment, the MAC layer may provide to the RLC layer an indication that a portion of data was not successfully transmitted. As noted above, the MAC layer may also provide a copy of the portion of data itself. In response to receiving the indication, the RLC layer may determine which RLC PDU/s were used to create the portion of data that was unsuccessfully transmitted. The copy of the data portion that was provided from the MAC layer may be used to perform this determination. In response to the determination, the RLC layer may then be configured to resend the RLC PDU from which the portion of data was created to the MAC layer. The MAC layer may then be configured to process the RLC PDU/s, as previously described, and pass the resulting MAC PDU/s to the physical layer for transmission.

In one embodiment, the RLC layer may store the copy of the portion of data that was transmitted from the MAC layer, until it receives an indication as to whether or not the transmission was successful. If it receives an indication that the transmission was successful, then the RLC layer may be configured to delete the copy of the portion of data.

A receiving entity is provided at a location in the network and is configured to receive the MAC PDU that was sent by the transmitting entity. The receiving entity may be a UE or may be another type of apparatus.

Reference is now made to figure 7, which shows a method performed at the receiving entity, which is performed after the receiving entity successfully receives the MAC PDU from the transmitting entity, and the physical layer has passed the MAC PDU to the MAC layer. It would be understood by the person skilled in the art that not all of these steps are essential to the invention, and that some are optional and may be omitted in embodiments. It should, furthermore be understood that the order of the steps of the method may modified in different embodiments.

At S701, the receiving entity is configured to sort the MAC PDUs into the order in which they were produced at the transmitting entity. In order to achieve this, the receiving entity is configured to inspect the sequence number in each MAC PDU header and order the MAC PDU packets in dependence upon the sequence numbers in the packets. In some embodiments, this step may only be performed in the case that the receiving entity is configured to perform hybrid automatic repeat request (HARQ). In this case, the reordering may be needed if there are multiple parallel HARQ processes.

At S702, the MAC layer extracts from the MAC PDUs the payloads (i.e. the data portions) of the MAC PDUs.

At S703, it is determined whether or not the payload of the received MAC PDU begins with an RLC header. If so, the method proceeds to S704. If not, the method proceeds to S705.

At S704, the MAC layer processes the data portions to obtain the RLC PDUs. This process may reverse the segmentation process performed by the MAC layer of the transmitting entity to obtain the data portions. In this case, the one or more data portions located in the buffer at the receiving entity may be concatenated by the MAC layer to produce at least one RLC PDU. The RLC PDUs may then be passed by the MAC layer to the RLC layer. RLC PDUs are processed by the RLC layer. This processing includes but may not be limited to segmentation in which an RLC PDU is unpacked to form several RLC SDUs, which are then passed to the higher layer (in this case PDCP layer) in the stack.

At S705, it is determined whether or not the receiving entity has any received partially complete RLC PDU stored, which correspond to the partially complete RLC PDU contained in the received MAC PDU. If so, the method proceeds to S705. If not, the method proceeds to S706.

At S706, the partially complete RLC PDU from the received MAC PDU are processed using the stored partially complete RLC PDU to produce complete RLC PDU. The complete RLC PDU are then processed by the RLC layer as in S703.

At S707, the receiving entity deletes the incomplete part of the RLC PDU contained in the received MAC PDU.

At S708, if there is an incomplete RLC PDU located at the end of the MAC PDU, it is stored in the receiving entity.

If, on the other hand, the MAC PDU is not successfully received at the receiving entity, for example if there are errors in the packet, the receiving entity deletes the unsuccessfully received MAC PDU.

In embodiments, segmentation and concatenation functions may be performed in the MAC layer 302. The RLC layer may continue to perform other processing functions, such as error correction, reordering, and duplicate detection. The processes of producing and transmitting the resulting MAC PDUs may be carried out by any suitable transmitting entity, which may, for example, take the form of the UE described above with reference to figure 2. Similarly, the receipt of receive processing of the MAC PDUs, which is also described below, may be performed by any suitable receiving entity, which may, for example, take the form of the UE described above with reference to figure 2.

Reference is made to figures 8a and 8b, which illustrate the creation of the MAC PDUs from the RLC PDUs. This processing is performed by the MAC layer of the transmitting entity. One or more RLC PDUs 801 are provided by the RLC layer to the MAC layer. Each RLC PDU comprises a payload 802 and a header 803. As shown, each header 803 may comprise a sequence number that may be used by the receiving entity to which the RLC PDUs are transmitted to, to determine the correct order of the RLC PDU packets.

The receiving entity may be configured to store the RLC PDUs in a buffer, which is accessible to the MAC layer. The MAC layer may then access the RLC PDUs stored in the buffer and perform a segmentation process to divide the three RLC packets along the lines 804 shown in figure 8. The resulting portions of data each provide a payload for a MAC PDU. MAC PDU headers are added to each of the portions of data to produce the MAC PDUs. As shown in figure 8b, if the portion of data to which the MAC PDU header 809 is added has RLC PDU payload data 808 placed in front of the first RLC PDU header 810, the RLC PDU payload data

808 that is placed in front of the first PDU header 810 is added to the MAC PDU header 809. In other words, if the portion of data comprises an RLC PDU header, but the portion of data does not begin with an RLC PDU header, an indication of the number of bits 808 of the portion of data prior to the first RLC PDU header 810 of the portion of data is added to the MAC PDU header 809.

Figure 8a shows three example MAC PDUs 807a, 807b, and 807c, which are formed by affixing MAC PDU headers to each of the three portions of data. A first MAC PDU 807a is formed having a payload 805a, and a header 806a. In this case, the payload 805a does not include an RLC header. A second MAC PDU 807b is formed having a payload 805b and a header 806b. The portion of data used to form this MAC PDU 807b has data 808 prior to the first RLC PDU header 810, and therefore an indication of the number of bits of this data 808 forms part of the header 809, when the MAC PDU packet is complete. A third MAC PDU 807c is formed having payload 805c and MAC PDU header 806c. In this case, the payload 805c includes an RLC PDU header, but one that is located at the start of the RLC PDU.

Figure 8b shows how the complete RLC PDUs have been segmented and multiplexed into different MAC PDUs. The first MAC PDU includes a partial RLC PDU 811a. The second MAC PDU includes another partial RLC PDU 811b. Since these two partial PDUs are segments of the same RLC PDU, together they form a complete RLC PDU 813. Therefore, one MAC PDU includes one part of an RLC PDU and a second MAC PDU includes a second part of an RLC PDU.

Figure 8b additionally shows that a single MAC PDU may include more than one complete or partial RLC PDUs. As shown, in figure 8b, the second MAC PDU includes a complete RLC PDU 813, a partial RLC PDU 811b, and another partial RLC PDU 814.

When the MAC PDUs are complete, the transmitting entity transmits them over the network to the receiving entity. Following the transmission of a MAC PDU, the MAC layer provides an indication that the transmission has taken place to the RLC layer together with the one or more data portions that were transmitted in the MAC PDU. Using the ARQ mechanism, the RLC layer is configured to determine whether or not the transmission to the receiving entity was successful or not. This can be achieved by waiting for an ACK from the receiving entity indicating successful receipt of the MAC PDU containing the portion of data. If the ACK is not received, it may be determined that the transmission was not successful. If it is determined that the transmission was not successfully, the MAC PDU containing the portion of data is then retransmitted to the receiving entity.

By performing the segmentation and concatenation functions at the MAC layer, the bottleneck at the RLC layer may be substantially reduced in some embodiments.

The receiving entity may be a UE or may another form of device. In the embodiments described, the receiving entity also uses the same LTE protocol stack as the transmitting entity. However, it would be understood by the person skilled in the art that the RLC protocol layer and MAC protocol layer are given as examples of a first and second protocol layer, and that the invention is not limited to implantation using the LTE protocol stack.

At the receiving entity, the payloads 805a, 805b, and 806c are extracted from the MAC PDUs and used by the MAC layer to reconstruct the RLC PDUs, which are then passed to the RLC layer. In performing this operation, the MAC layer may first extract the MAC PDU payloads (i.e. data portions) from the MAC PDUs. The MAC layer may then process the MAC PDU payloads to produce at least one RLC PDU. The at least one RLC PDU may then be passed by the MAC layer to the RLC layer.

Upon receipt of the data packet / MAC PDU at the receiving entity, the receiving entity is configured to transmit an ACK to the transmitting entity indicating successful transmission of the data packet. If an expected MAC PDU is not successfully received at the receiving entity, the receiving entity may instead transmit a NACK. The NACK may be configured to cause the transmitting entity to retransmit the data packet which was not received

Reference is made to figure 9, which shows examples of how the RLC PDUs are obtained from the received MAC PDUs. If the payload of the received MAC PDU begins with an RLC PDU header, then the complete RLC PDUs of the MAC PDU are processed by the RLC layer. This case is shown in figure 9a, wherein a MAC PDU payload 910 having two complete RLC PDUs is received at the receiving entity. Both of these RLC PDUs can be extracted from the payload 910 and transferred to the RLC layer for processing.

The receiving entity may be configured to store in its memory, partial RLC PDUs that it receives from the transmitting entity, so that it is able to complete the RLC PDUs if corresponding partial PDUs are later received. When a MAC PDU is received, if the payload of the MAC PDU does not begin with an RLC PDU header, the receiving entity checks its memory to determine if the remaining part/s of the RLC PDU are found there. If the remaining parts of the RLC PDU are found stored in the memory of the receiving entity, the receiving entity is configured to combine the stored partial RLC PDU with the received RLC PDU to form

a complete RLC PDU. This complete RLC PDU is then passed to the RLC layer for processing. Figure 9b illustrates an example of the case where a partial RLC PDU 930 is contained in the MAC PDU payload 920. As shown, in this case, the memory of the receiving entity contains a stored incomplete RLC PDU, which corresponds to the incomplete RLC PDU 930 in the received MAC PDU payload 920. The receiving entity is configured to combine these two incomplete RLC PDUs to form a complete RLC PDU, which may then be passed to the RLC layer for processing.

Reference is made to figure 10, which shows examples of actions that may take place if a MAC PDU payload that does not begin with an RLC PDU payload is received at the receiving entity and no partial/incomplete RLC PDU is stored that may complete the RLC PDU. As shown in the figure, if a MAC PDU payload 1010 or 1020 beginning with an incomplete RLC PDU is received, and no incomplete RLC PDU is stored which can complete the RLC PDU, the receiving entity is configured to delete the incomplete RLC PDU at the start of the MAC PDU. If the incomplete RLC PDU is the only RLC PDU in the MAC PDU payload – as is the case for the MAC PDU payload 1010 – then the entire MAC PDU payload is deleted. On the other hand, if the MAC PDU contains one or more RLC PDUs in addition to the incomplete RLC PDU at the start of the MAC PDU payload – as is the case for payload 1020 – then only the incomplete/partial RLC PDU at the start of the payload 1020 is deleted. The resulting payload 1030 is then shown below, with the incomplete RLC PDU deleted. The remaining complete RLC PDU in the payload is then processed by the RLC layer.

Reference is made to figure 11, which illustrates an example of actions that be performed when a MAC PDU payload 1110, which ends with an incomplete RLC PDU 1120, is received at the receiving entity. In this, case the incomplete RLC PDU 1120 is stored in the memory of the receiving entity. This incomplete RLC PDU may then be used to complete an RLC PDU if an incomplete RLC PDU of the same complete RLC PDU is received at the receiving entity at a later point in time.

Reference is made to figure 12, which illustrates examples of an action that may be performed if the MAC PDU 1210 that contains the incomplete RLC PDU for completing the incomplete RLC PDU 1220 stored in the memory of the receiving entity is not successfully received. If the MAC PDU 1210 is not successfully received, the stored incomplete RLC PDU 1220 is then deleted from the memory of the receiving entity. In one embodiment, the stored incomplete RLC PDU is deleted after the expiry of a predefined time period after being received or stored without the MAC PDU containing the incomplete RLC PDU for completing the RLC PDU.

In some embodiments, the data that is transmitted from the transmitting entity to the receiving entity may belong to more than one logical channel. For example, some RLC PDUs produced for transmission may comprise data belonging to a first logical channel, whereas other RLC PDUs may comprise data belong to a second logical channel.

Reference is made to figure 13, which provides examples of how the transmitting entity may manage the transmission of data belonging to different logical channels. In one embodiment, RLC PDUs of different logical channels are multiplexed together into the same buffer. In another embodiment, RLC PDUs belonging to different logical channels are stored in separate buffers, wherein there is a first buffer for RLC PDUs of a first logical channel and a second buffer for RLC PDUs of a second logical channel. The data of the first buffer is provided to the MAC layer separately to the data in the second buffer. Therefore, when a MAC PDU is created, the RLC PDUs which contain data of only a single logical channel.

In figure 13a, three RLC PDUs are provided by the RLC layer in the buffer. In this case, the RLC PDUs of different logical channels have been multiplexed together into the same buffer. The RLC PDUs 1310a, 1310b belongs to a first logical channel, whilst the RLC PDU 1320 belongs to a second logical channel.

In figure 13b, three RLC PDUs are provided by the RLC layer in each buffer. The RLC PDUs belonging to logical channel number 1 are stored in buffer 1, whilst the RLC PDUs belong to logical channel number 2 are stored in a buffer 2.

In figure 13c, the creation of the MAC PDU is illustrated. As shown, the first part of MAC PDU 1330 is constructed using RLCL PDUs containing data of the first channel, whilst a second part of MAC PDU 1340 is constructed using data from the second channel. The MAC PDU header may be used for both MAC PDU parts.

In one embodiment, if an incomplete/partial RLC PDU is stored at the receiving entity due to one of the MAC PDUs ending in the incomplete/partial RLC PDU (as discussed above), then if the expected MAC PDU is not received within a predefined time limit, all of the stored incomplete/partial RLC PDU from all of the logical channels are deleted.

Reference is made to figure 14 which illustrates a method performed at the receiving entity upon receiving a MAC PDU containing a first RLC PDU 1410 comprising data of first logical channel, and second RLC PDU 1420 comprising data of a second logical channel which is different to the first logical channel. Both the first and second RLC PDUs are incomplete PDUs,

which are stored in the memory of the receiving entity upon receipt at the receiving entity. As shown, if an expected MAC PDU containing incomplete RLC PDUs that combine with the stored incomplete RLC PDUs to form complete RLC PDUs is not received within a predetermined time period, both of the stored RLC PDUs are deleted from the memory of the receiving entity.

It is noted that whilst embodiments have been described in relation to one example of a standalone LTE network, similar principles may be applied in relation to other examples of standalone 3G, LTE or 5G networks. It should be noted that other embodiments may be based on other cellular technology other than LTE or on variants of LTE. Therefore, although certain embodiments were described above by way of example with reference to certain example architectures for wireless networks, technologies and standards, embodiments may be applied to any other suitable forms of communication systems than those illustrated and described herein.

It is also noted herein that while the above describes example embodiments, there are several variations and modifications which may be made to the disclosed solution without departing from the scope of the present invention.

The method may additionally be implemented in a control apparatus as shown in Figure 15. The method may be implanted in a single processor 201 or control apparatus or across more than one processor or control apparatus. Figure 15 shows an example of a control apparatus for a communication system, for example to be coupled to and/or for controlling a station of an access system, such as a RAN node, e.g. a base station, (e) node B, a central unit of a cloud architecture or a node of a core network such as an MME or S-GW, a scheduling entity such as a spectrum management entity, or a server or host. The control apparatus may be integrated with or external to a node or module of a core network or RAN. In some embodiments, base stations comprise a separate control apparatus unit or module. In other embodiments, the control apparatus can be another network element such as a radio network controller or a spectrum controller. In some embodiments, each base station may have such a control apparatus as well as a control apparatus being provided in a radio network controller. The control apparatus 1500 can be arranged to provide control on communications in the service area of the system. The control apparatus 1500 comprises at least one memory 1510, at least one data processing unit 1520, 1530 and an input/output interface 1540. Via the interface the control apparatus can be coupled to a receiver and a transmitter of the base station. The receiver and/or the transmitter may be implemented as a radio front end or a remote radio head. For example the control apparatus 1500 or processor 201 can be

configured to execute an appropriate software code to provide the control functions. Control functions may comprise providing a plurality of data packets according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack; segmenting, at the second protocol layer, the plurality of data packets according to the first protocol into a plurality of data portions; processing each data portion at the second protocol layer to produce a data packet according to the second protocol; and transmitting the data packets according to the second protocol to a user device.

Alternatively, or in addition, control functions may comprise receiving a data packet according to a second protocol from a user device, wherein the data packet according to the second protocol was generated at the user device by the steps of: providing a plurality of data packets according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack; segmenting, at the second protocol layer, the plurality of data packets according to the first protocol into a plurality of data portions; processing each data portion at the second protocol layer to produce the data packet according to the second protocol.

It should be understood that the apparatuses may comprise or be coupled to other units or modules etc., such as radio parts or radio heads, used in or for transmission and/or reception. Although the apparatuses have been described as one entity, different modules and memory may be implemented in one or more physical or logical entities.

In general, the various embodiments may be implemented in hardware or special purpose circuits, software, logic or any combination thereof. Some aspects of the invention may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed by a controller, microprocessor or other computing device, although the invention is not limited thereto. While various aspects of the invention may be illustrated and described as block diagrams, flow charts, or using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

The embodiments of this invention may be implemented by computer software executable by a data processor of the mobile device, such as in the processor entity, or by hardware, or by a combination of software and hardware. Computer software or program, also called program product, including software routines, applets and/or macros, may be stored in any apparatus-

readable data storage medium and they comprise program instructions to perform particular tasks. A computer program product may comprise one or more computer-executable components which, when the program is run, are configured to carry out embodiments. The one or more computer-executable components may be at least one software code or portions of it.

Further in this regard it should be noted that any blocks of the logic flow as in the Figures may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and functions. The software may be stored on such physical media as memory chips, or memory blocks implemented within the processor, magnetic media such as hard disk or floppy disks, and optical media such as for example DVD and the data variants thereof, CD. The physical media is a non-transitory media.

The memory may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The data processors may be of any type suitable to the local technical environment, and may comprise one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASIC), FPGA, gate level circuits and processors based on multi core processor architecture, as non-limiting examples.

Embodiments of the inventions may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

The foregoing description has provided by way of non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this invention will still fall within the scope of this invention as defined in the appended claims. Indeed there is a further embodiment comprising a combination of one or more embodiments with any of the other embodiments previously discussed.

Claims

1. A method comprising:
 - providing at least one data packet according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack;
 - segmenting, at the second protocol layer, the at least one data packet according to the first protocol into at least one data portion;
 - processing the at least one data portion at the second protocol layer to produce at least one data packet according to the second protocol; and
 - providing the at least one data packet according to the second protocol from the second protocol layer to a third protocol layer of the protocol stack.
2. A method as claimed in claim 1, wherein the step of processing the at least one data portion at the second protocol layer comprises affixing a header according to the second protocol to each data portion.
3. A method as claimed in claim 2, wherein the step of processing at the second protocol layer further comprises for each data portion:
 - determining that the data portion comprises one or more headers according to the first protocol; and
 - in response to the determination, indicating in the header according to the second protocol, the number of bits of the data packet that precede a first header according to the first protocol in the data portion.
4. A method as claimed in any preceding claim, wherein at least one of the data portions comprises partial data packets according to the first protocol.
5. A method as claimed in any preceding claim, wherein at least one of the data portions comprises the concatenation of at least one of data packets and partial data packets according to the first protocol.
6. A method as claimed in any preceding claim, wherein, each data portion forms a payload of a data packet according to the second protocol.
7. A method as claimed in any preceding claim, wherein the at least one data packets according to the first protocol include one or more data packets comprising data of a first logical channel and one or more data packets comprising data of a second logical channel.

8. A method as claimed in claim 7, wherein at least one of the data packets according to the second protocol comprises data of both the first logical channel and data of the second logical channel.

9. A method as claimed in claim 7, wherein at least one of the data packets according to the second protocol comprise data of only one of: the first logical channel and the second logical channel.

10. A method as claimed in any preceding claim, wherein the step of providing at least one of the data packets according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack comprises: storing the at least one data packets according to a first protocol in one or more buffers that are accessible to the second protocol layer.

11. A method as claimed in claim 10, wherein the at least one data packets according to the first protocol include one or more data packets comprising data of a first logical channel and one or more data packets comprising data of a second logical channel, and the step of storing the plurality of data packets according to the first protocol in one or more buffers comprises:

storing the one or more data packets comprising data of the first logical channel in a first buffer of the one or more buffers; and

storing the one or more data packets comprising data of the second logical channel in a second buffer of the one or more buffers.

12. A method as claimed in any preceding claim, further comprising:

providing copies of the at least one data portions from the second protocol layer to the first protocol layer of the protocol stack; and

transmitting a data packet according to the second protocol comprising the copies of the at least one data portions if a message indicating that the transmission of that data packet has failed is received.

13. A method as claimed in any preceding claim, further comprising:

providing an indication that the transmission of a data portion has failed from the second protocol layer to the first protocol layer;

determining a data packet according to the first protocol that was at least partially included in that data portion; and

transmitting that data packet to a receiving entity.

14. A method as claimed in any preceding claim, further comprising:
performing ARQ error correction procedures to each data packet according to the first protocol at the first protocol layer of a protocol stack.
15. A computer program product for a computer, comprising software code portions for performing the steps of any of claims 1 to 14 when the product is run on the computer.
16. An apparatus comprising:
at least one processor and at least one memory including a computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:
provide at least one data packet according to a first protocol from a first protocol layer of a protocol stack to a second protocol layer of the protocol stack;
segment, at the second protocol layer, the at least one data packet according to the first protocol into at least one data portion;
process the at least one data portion at the second protocol layer to produce at least one data packet according to the second protocol; and
provide the at least one data packet according to the second protocol from the second protocol layer to a third protocol layer of the protocol stack.
17. A method comprising:
receiving at least one data packet according to the second protocol at a second layer of a protocol stack from a third layer of the protocol stack;
processing the at least one data packet according to the second protocol at the second protocol layer to produce at least one data portion; and
concatenating, at the second protocol layer, the at least one data portion to produce at least one data packet according to a first protocol; and
providing the at least one data packet according to the first protocol from the second protocol layer to a first protocol layer of the protocol stack.
18. A method as claimed in claim 17, wherein one of the data packets according to the second protocol comprises one or more data packets of a first protocol, wherein a data packet of the first protocol located at a start of the one of the data packets of the second protocol is a first incomplete data packet.

19. A method as claimed in 18, further comprising:
loading from a memory, a second incomplete data packet of the first protocol; and
combining the first incomplete data packet with the second incomplete data packet
to form a complete data packet according to the first protocol.
20. A method as claimed in 18, further comprising:
determining that a memory does not contain a second incomplete data packet of
the first protocol that is configured to combine with the first incomplete data packet to form a
complete data packet; and
in response to the determination, deleting the first incomplete data packet.
21. A method as claimed in claim 19, further comprising, prior to receiving the at least
one data packet according to the second protocol at the second layer of the protocol stack:
receiving a second data packet according to the second protocol at the second
layer of a protocol stack from the third layer of the protocol stack, wherein the second data
packet according to the second protocol comprises the second incomplete data packet,
wherein the second incomplete data packet is located at an end of the second data packet
according to the second protocol; and
storing the second incomplete data packet in a memory.
22. A method as claimed in any of claims 17 or 21, wherein one of the at least one
data packets according to the second protocol comprises one or more data packets of a first
protocol, wherein a data packet of the first protocol located at an end of the one of the at least
one data packets of the second protocol is a second incomplete data packet, the method
comprising:
storing the second incomplete data packet in a memory; and
if a predetermined time period elapses, without a first incomplete data packet
being received, deleting the second incomplete data packet, wherein the first incomplete data
packet is configured to combine with the first incomplete data packet to form a complete data
packet.
23. A computer program product for a computer, comprising software code portions for
performing the steps of any of claims 17 to 22 when the product is run on the computer.
24. An apparatus comprising:

at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:

receive at least one data packet according to the second protocol at a second layer of a protocol stack from a third layer of the protocol stack;

process the at least one data packet according to the second protocol at the second protocol layer to produce at least one data portion;

concatenate, at the second protocol layer, the at least one data portion to produce at least one data packet according to a first protocol; and

provide the at least one data packet according to the first protocol from the second protocol layer to a first protocol layer of the protocol stack.

FIG 1

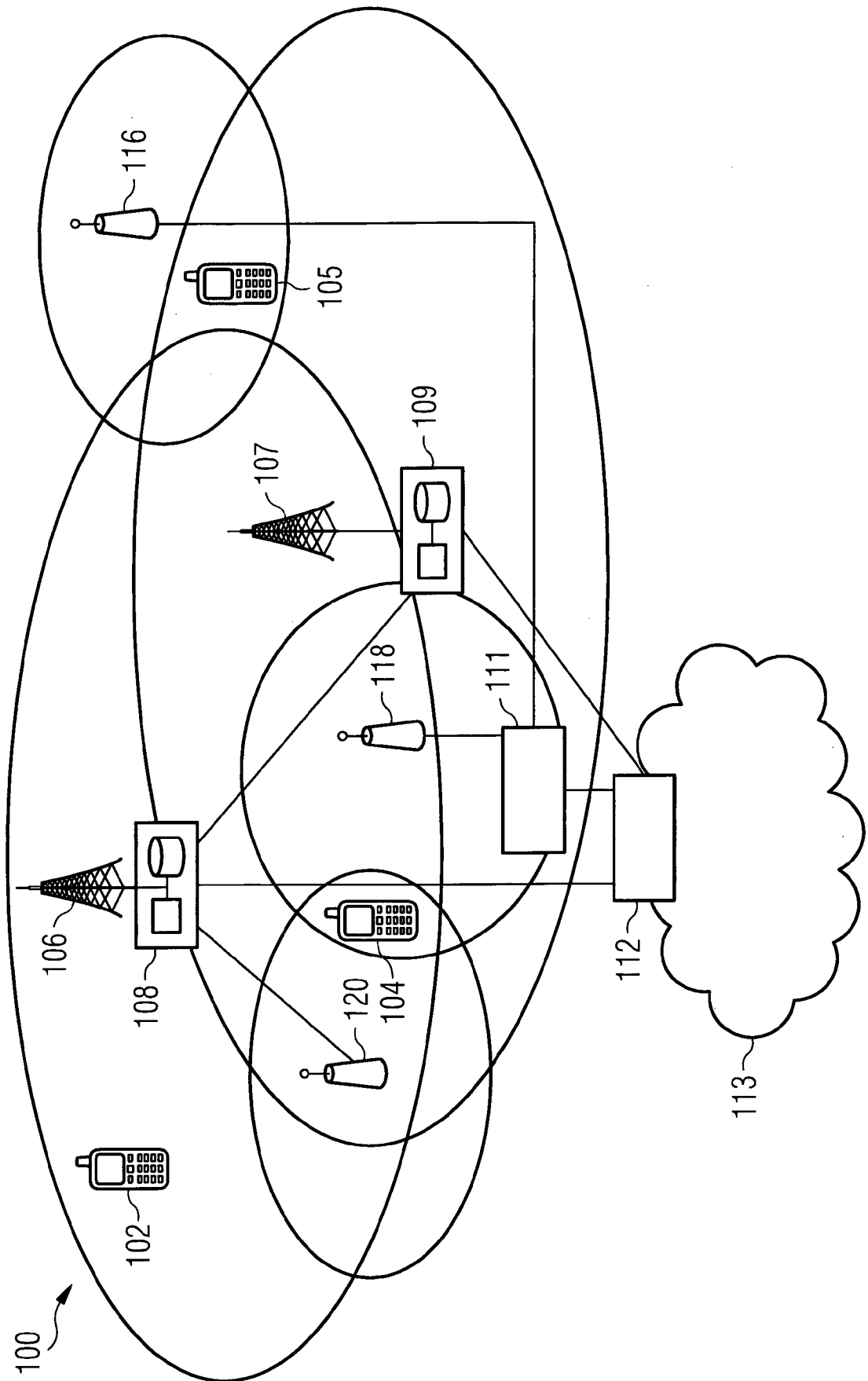


FIG 2

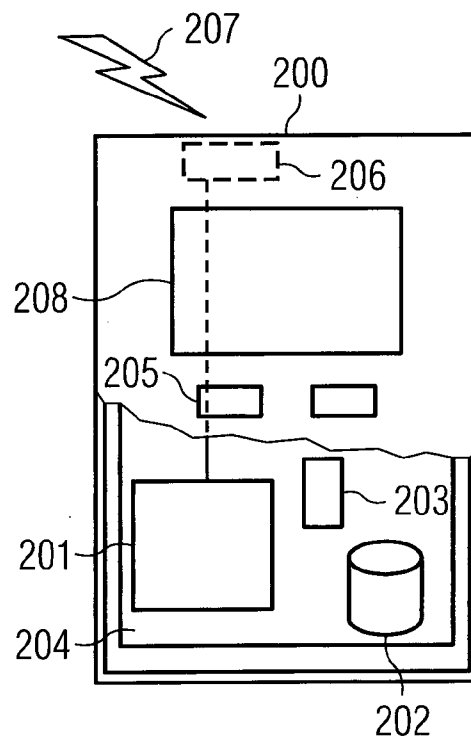


FIG 3

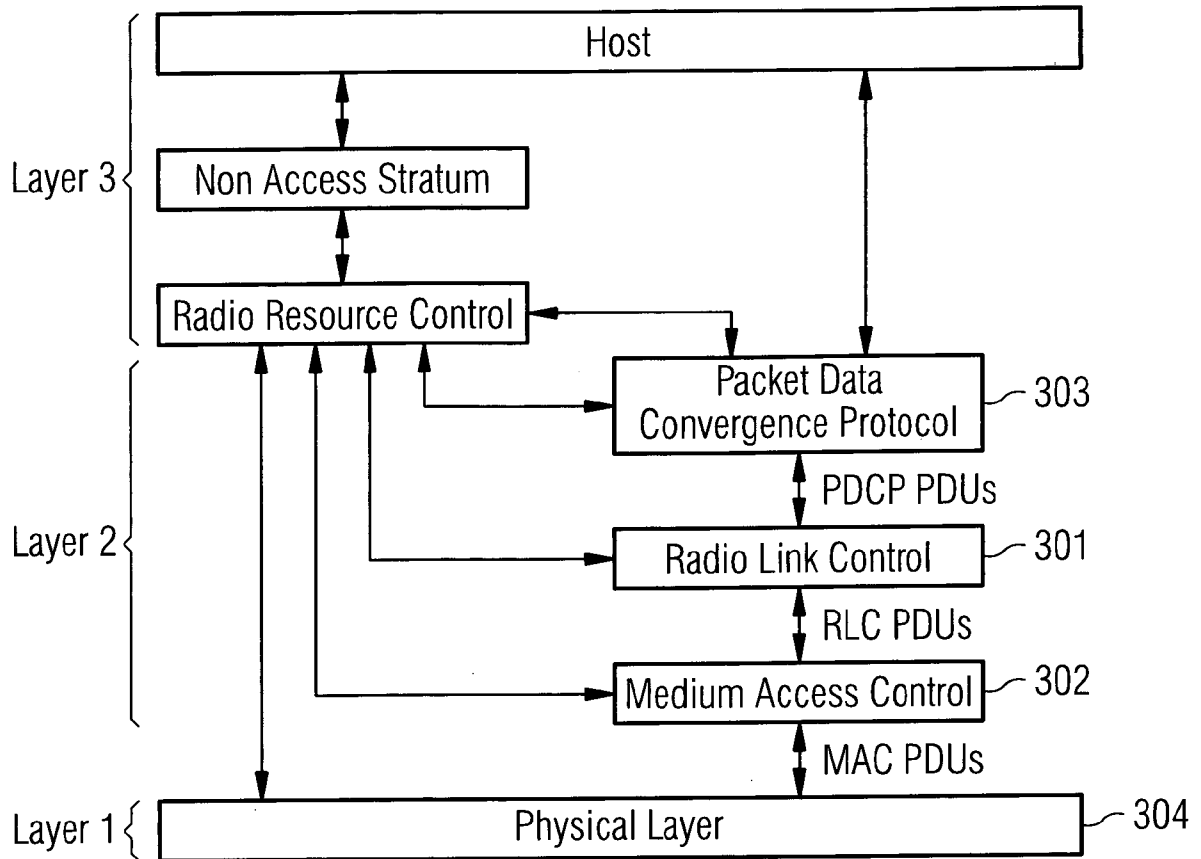


FIG 4

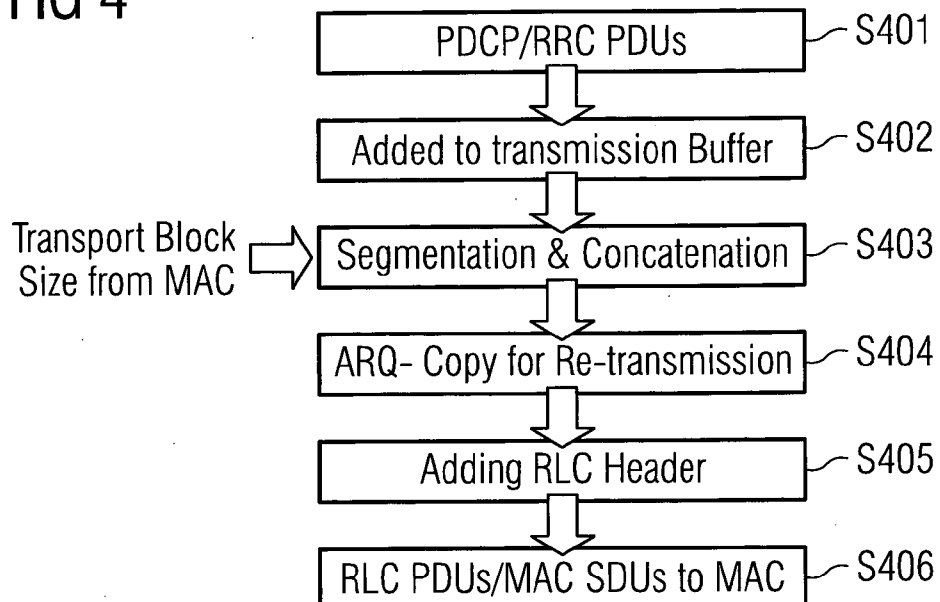


FIG 5

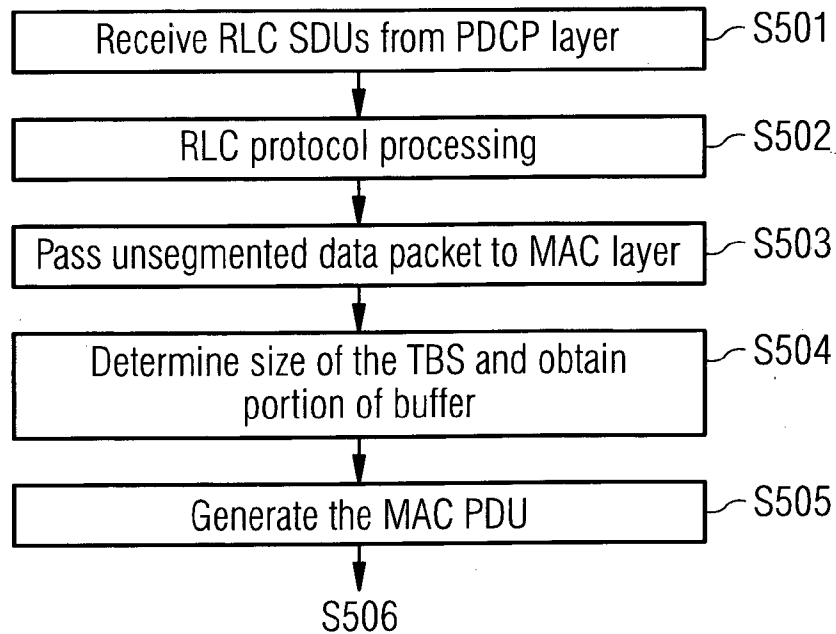


FIG 6

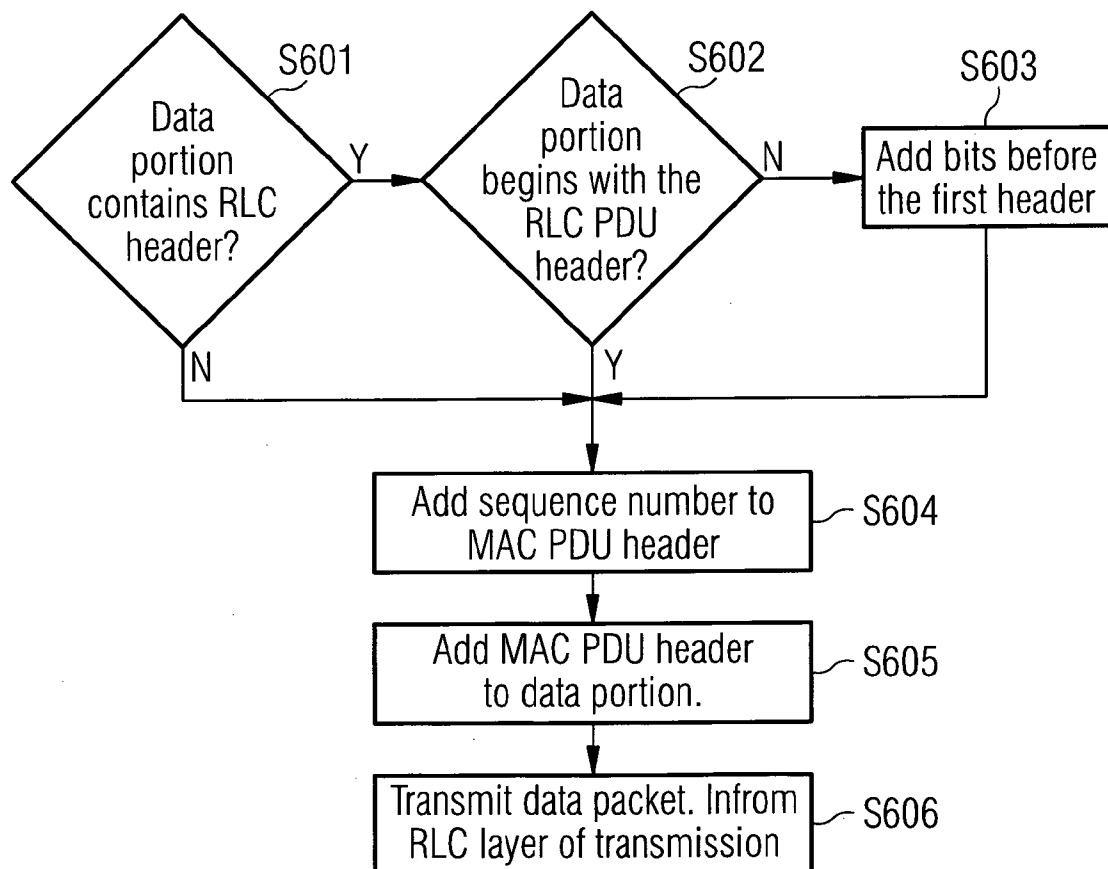


FIG 7

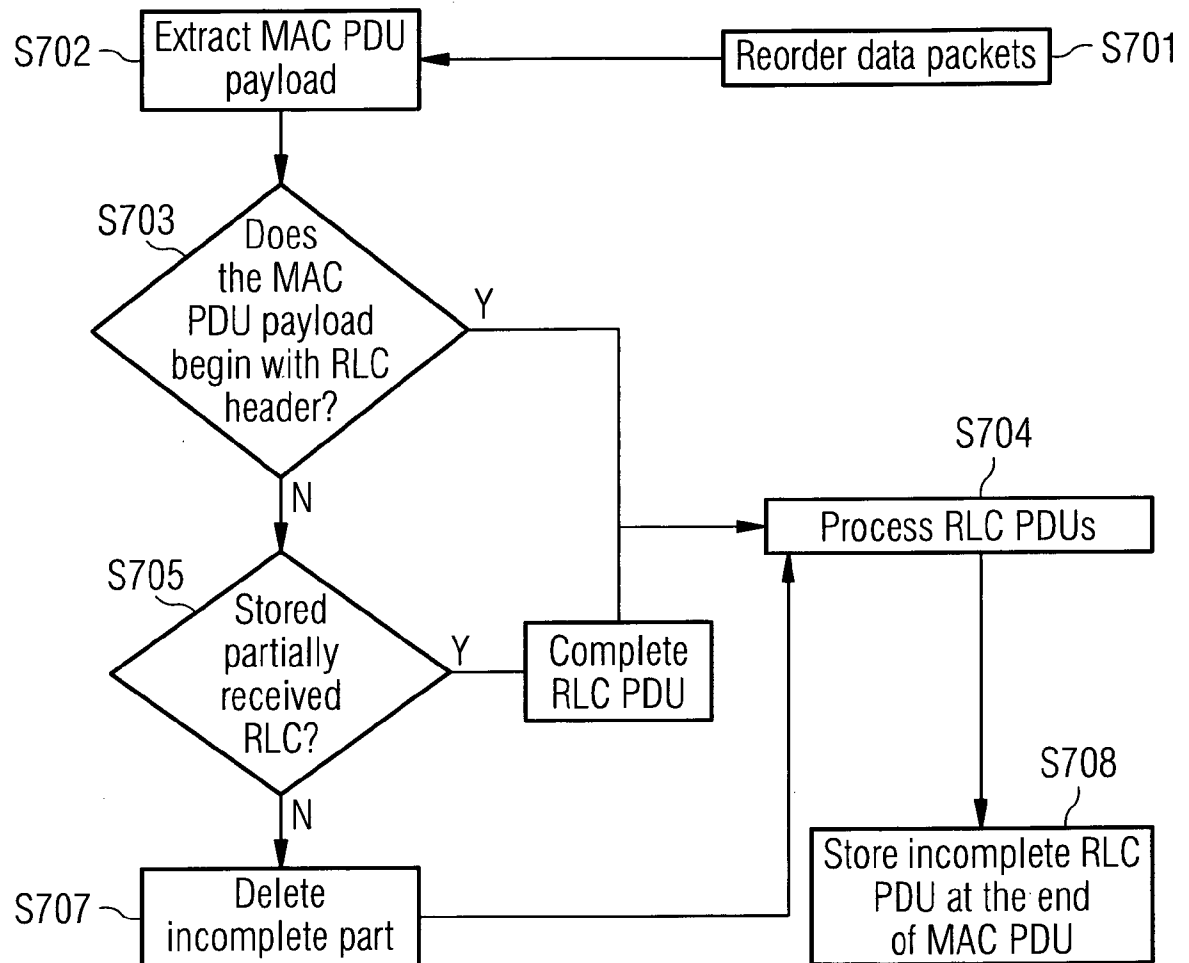


FIG 8A

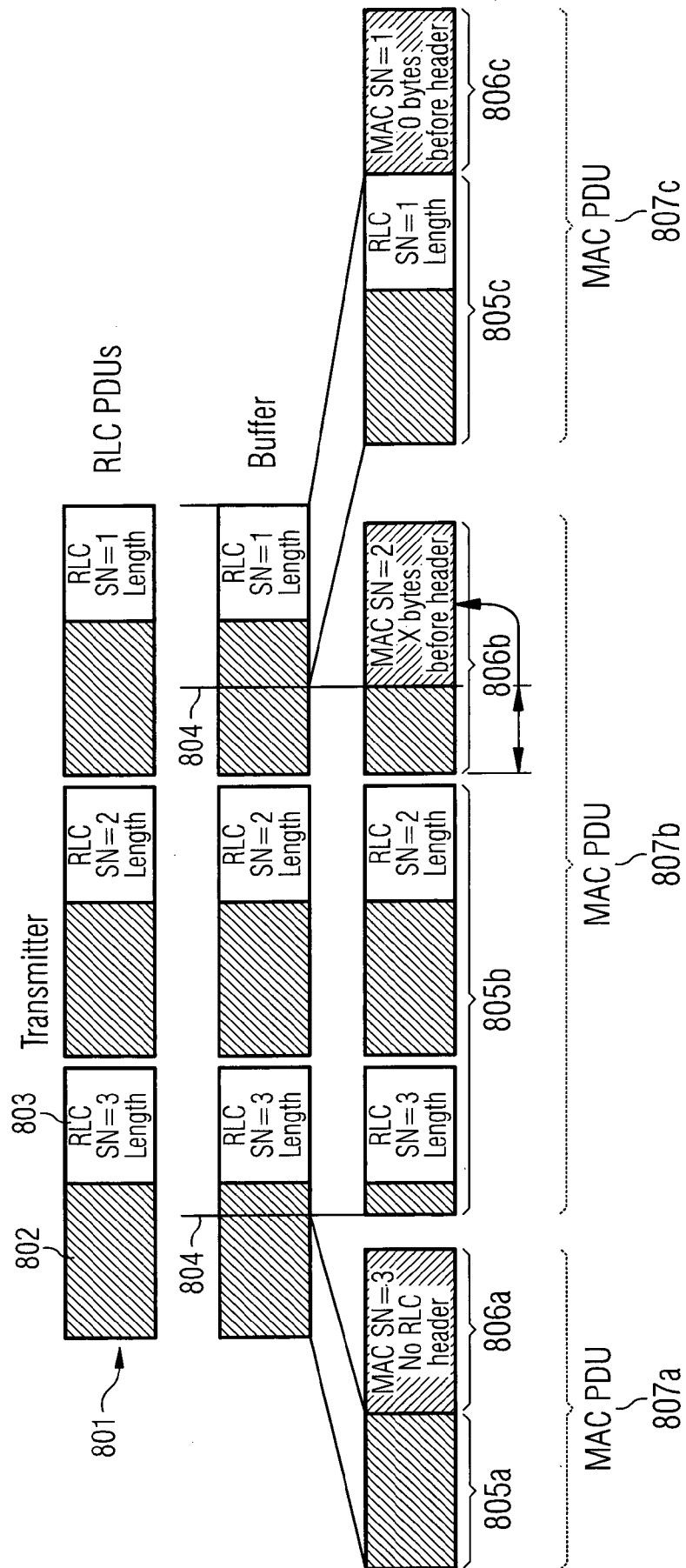


FIG 8B

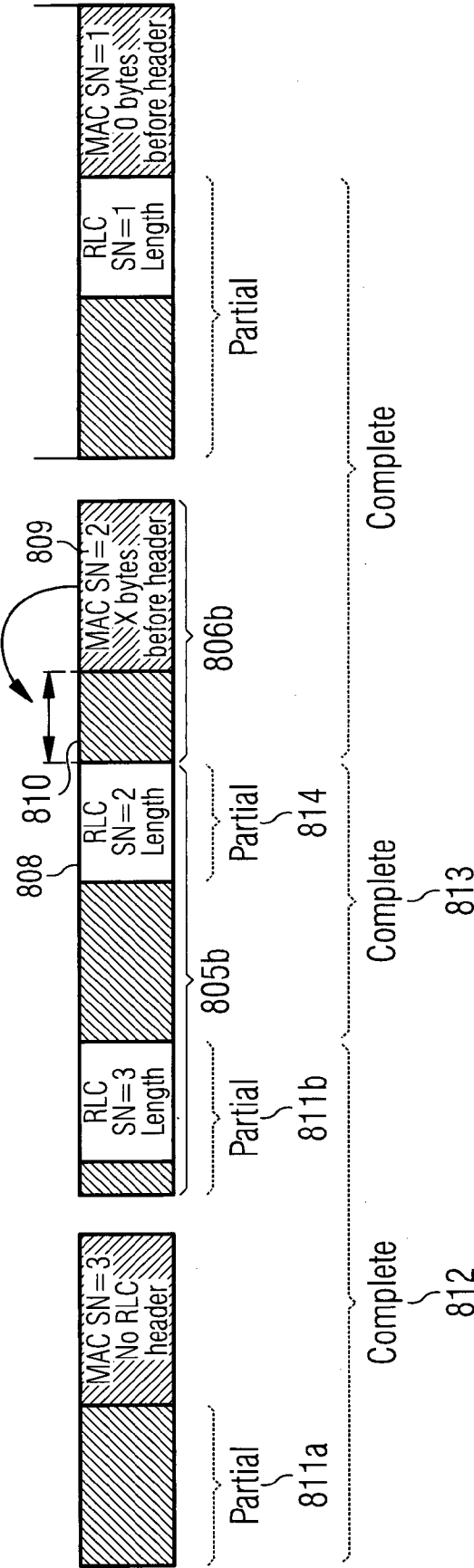


FIG 9A

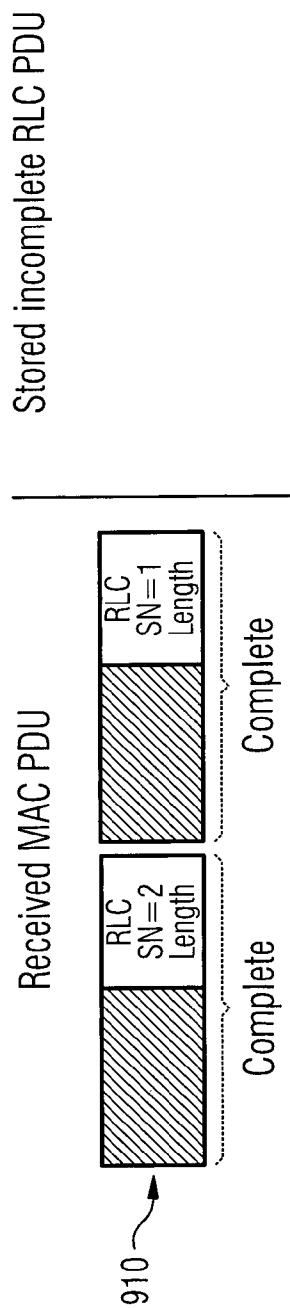


FIG 9B

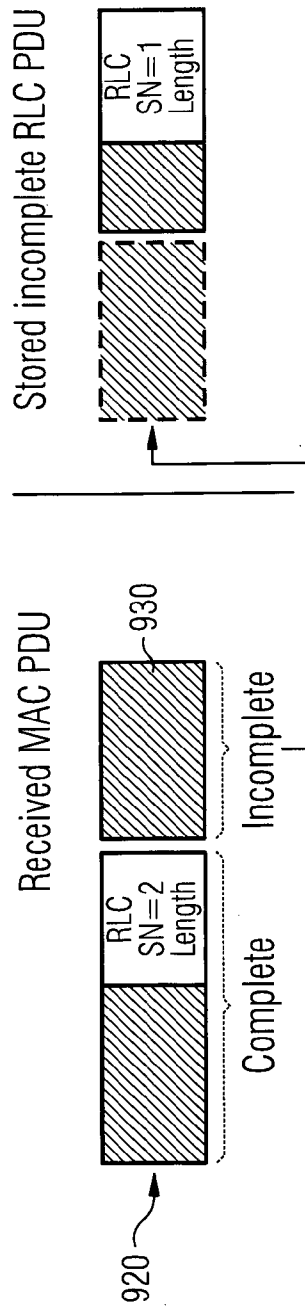


FIG 10

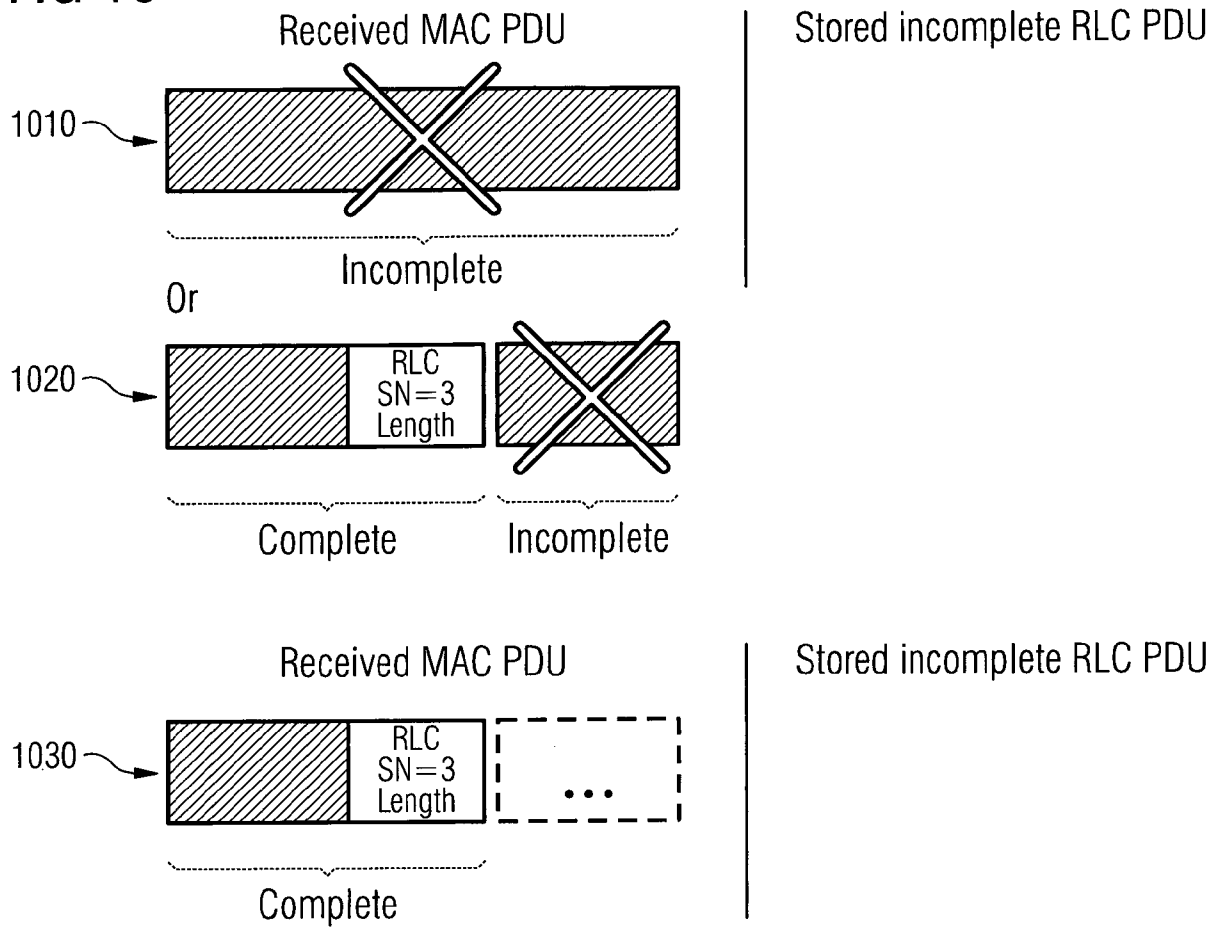


FIG 11

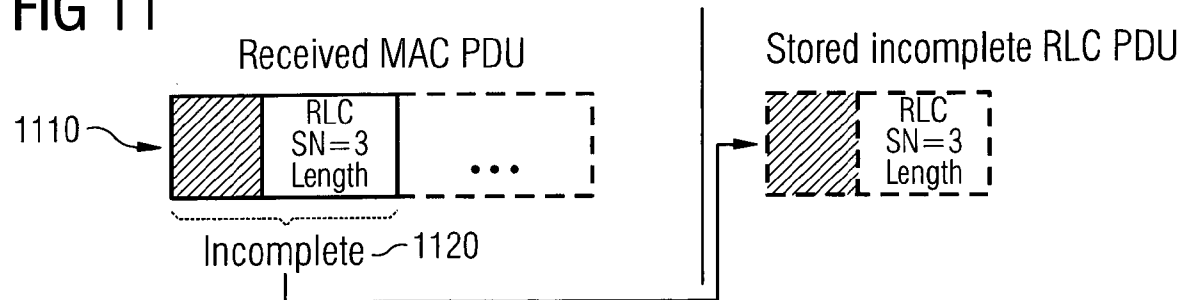


FIG 12

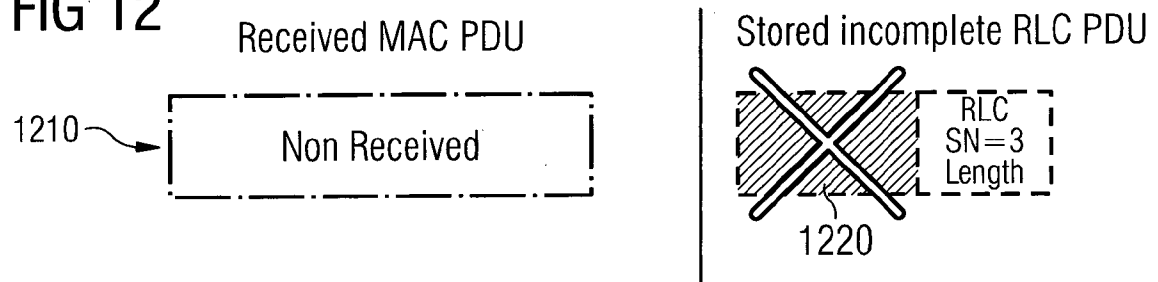


FIG 13A

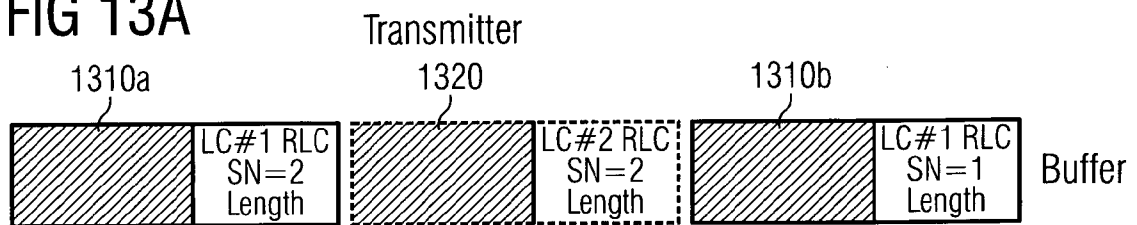


FIG 13B

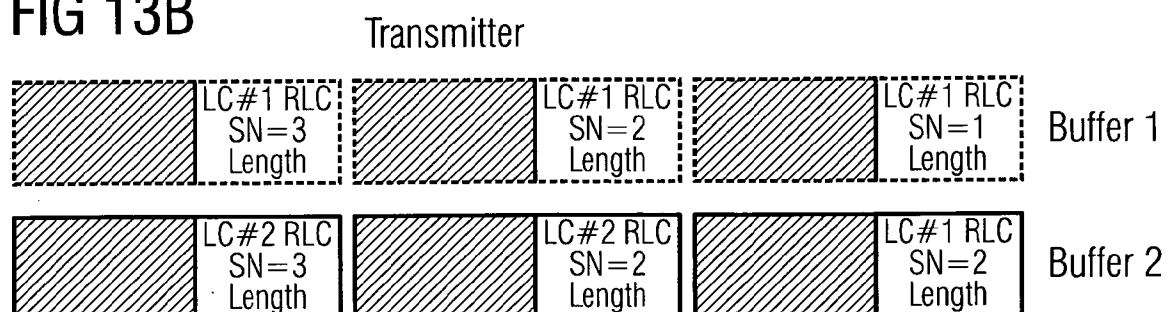
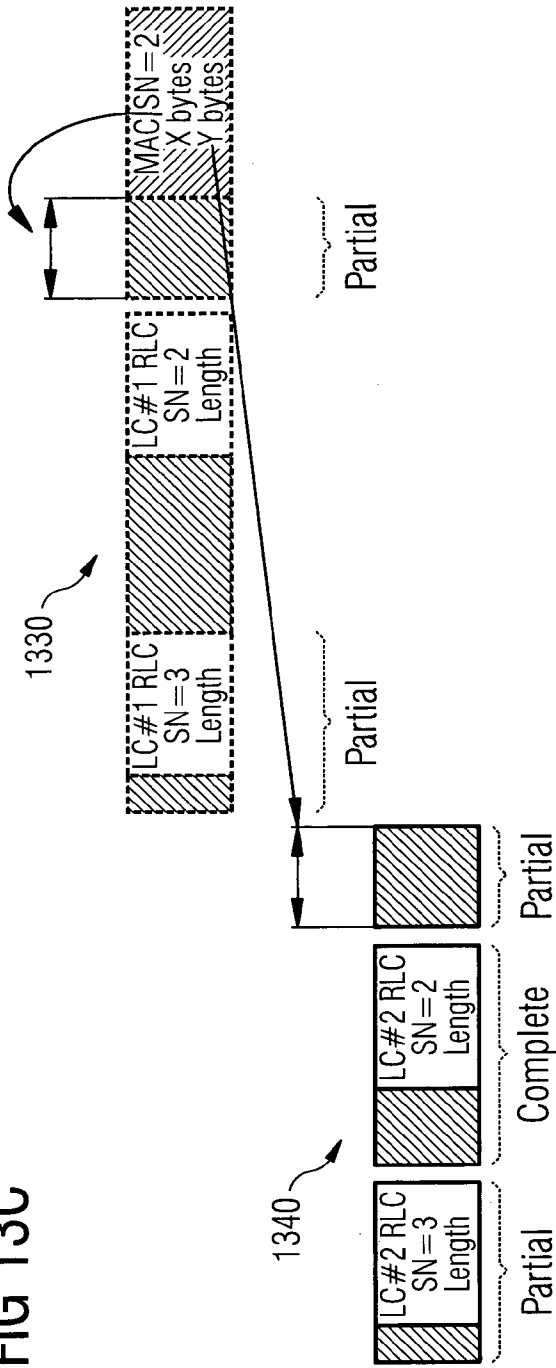


FIG 13C



12/12

FIG 14

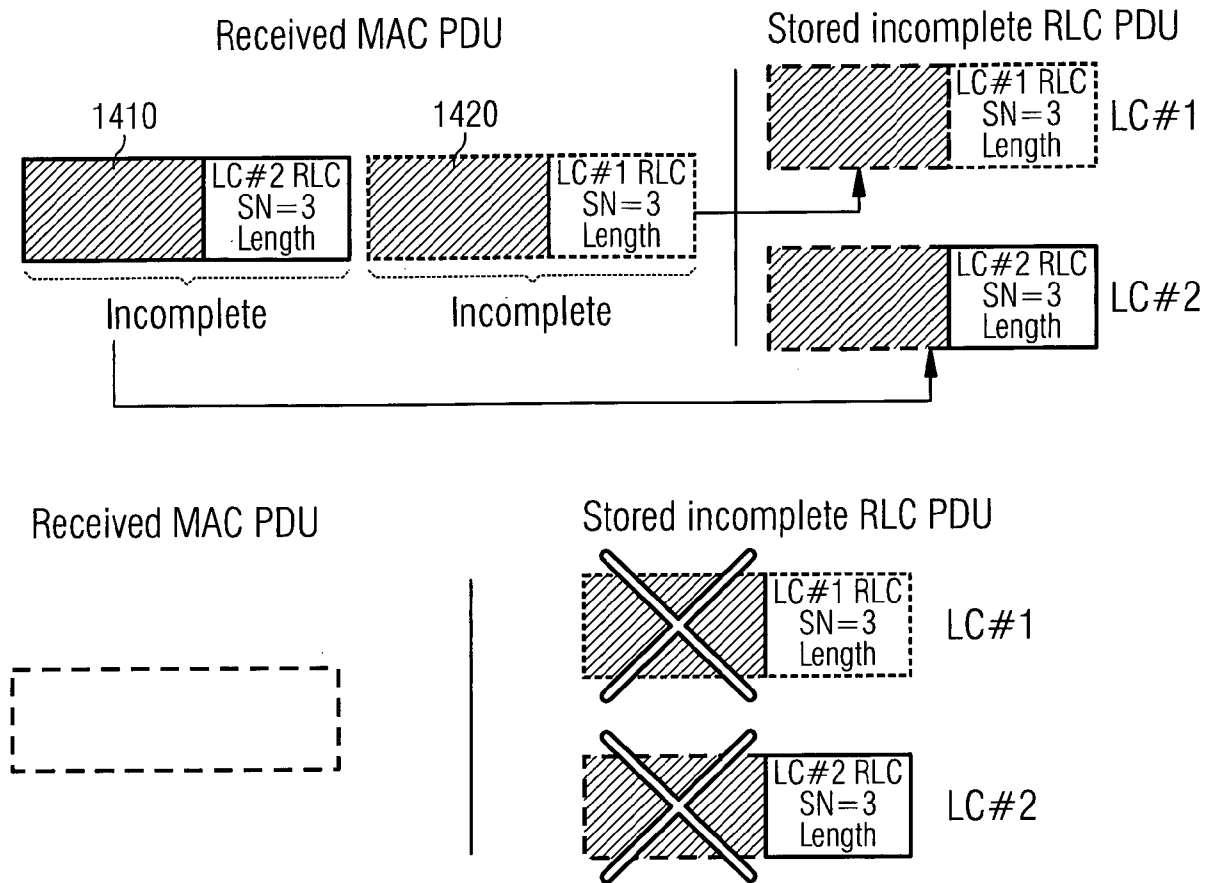
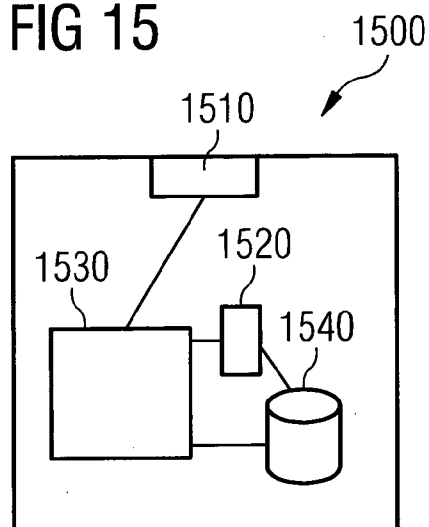


FIG 15



INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2016/067650

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L29/08 H04W28/06
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 H04W

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 A	WO 2008/084426 A1 (KONINKL PHILIPS ELECTRONICS NV [NL]; BUCKNELL PAUL [NL]; BAKER MATTHEW) 17 July 2008 (2008-07-17) figures 5, 1 page 1, line 16 - page 2, line 14 page 8, line 22 - page 9, line 8 ----- -/--	1,2,4,6, 14-19, 23,24 3,5, 7-13, 20-22



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"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

21 March 2017

Date of mailing of the international search report

28/03/2017

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Cuevas Casado, A

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/067650

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PHILIPS: "Impact of higher data rates on MAC header", 3GPP DRAFT; R2-070165, 3RD GENERATION PARTNERSHIP PROJECT (3GPP), MOBILE COMPETENCE CENTRE ; 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS CEDEX ; FRANCE, vol. RAN WG2, no. Sorrento, Italy; 20070112, 12 January 2007 (2007-01-12), XP050133272, [retrieved on 2007-01-12]	1,2,4,6, 14-19, 23,24
A	figures 2, 1 page 2 page 3	3,5, 7-13, 20-22
X	----- EP 2 077 687 A1 (SAMSUNG ELECTRONICS CO LTD [KR]) 8 July 2009 (2009-07-08) figures 6, 2 paragraph [0016] - paragraph [0017] paragraph [0025]	1,2,4-9, 15-19, 21,23,24
X	----- US 2010/232356 A1 (MAHESHWARI SHAILESH [US] ET AL) 16 September 2010 (2010-09-16) figures 4, 13 paragraph [0048] paragraph [0074] paragraph [0080] - paragraph [0082] paragraph [0091] - paragraph [0093] -----	1-6, 10-24

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2016/067650

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008084426	A1	17-07-2008	AT 507694 T 15-05-2011
		CN 101578824 A 11-11-2009	
		EP 2119142 A1 18-11-2009	
		JP 5357051 B2 04-12-2013	
		JP 2010516073 A 13-05-2010	
		KR 20090106507 A 09-10-2009	
		TW 200847713 A 01-12-2008	
		US 2009310537 A1 17-12-2009	
		US 2015156126 A1 04-06-2015	
		WO 2008084426 A1 17-07-2008	
EP 2077687	A1	08-07-2009	EP 2077687 A1 08-07-2009
		KR 20090075645 A 08-07-2009	
		US 2009175259 A1 09-07-2009	
US 2010232356	A1	16-09-2010	CN 102356589 A 15-02-2012
		EP 2409446 A1 25-01-2012	
		JP 5318277 B2 16-10-2013	
		JP 2012521161 A 10-09-2012	
		KR 20110138386 A 27-12-2011	
		TW 201116010 A 01-05-2011	
		US 2010232356 A1 16-09-2010	
		WO 2010107708 A2 23-09-2010	