



US 20090196175A1

(19) **United States**

(12) **Patent Application Publication**
Sammour et al.

(10) **Pub. No.: US 2009/0196175 A1**

(43) **Pub. Date: Aug. 6, 2009**

(54) **METHOD AND APPARATUS FOR
INITIALIZING, PRESERVING, AND
RECONFIGURING TOKEN BUCKETS**

(22) Filed: **Jan. 27, 2009**

Related U.S. Application Data

(75) Inventors: **Mohammed Sammour**, Montreal
(CA); **Stephen E. Terry**, Northport,
NY (US); **Jin Wang**, Central Islip,
NY (US)

(60) Provisional application No. 61/025,478, filed on Feb.
1, 2008.

Publication Classification

Correspondence Address:
VOLPE AND KOENIG, P.C.
DEPT. ICC
**UNITED PLAZA, SUITE 1600, 30 SOUTH 17TH
STREET**
PHILADELPHIA, PA 19103 (US)

(51) **Int. Cl.**
H04L 12/26 (2006.01)

(52) **U.S. Cl. 370/230.1**

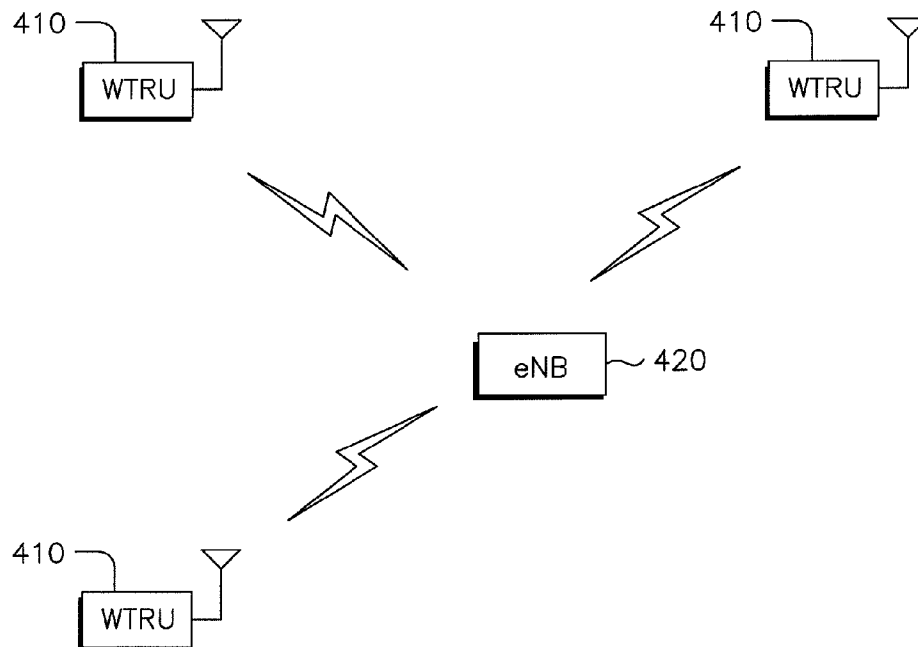
(73) Assignee: **INTERDIGITAL PATENT
HOLDINGS, INC.**, Wilmington,
DE (US)

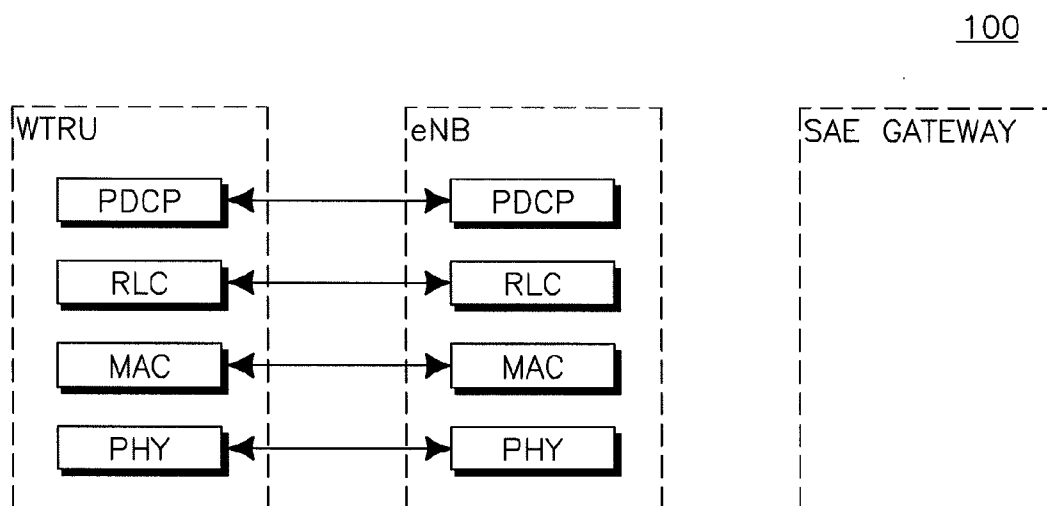
(57) **ABSTRACT**

A method and an apparatus are provided for initializing token
buckets to a non-zero value, preserving the token buckets
during a medium access control (MAC) reset or handover,
resetting the token buckets based on a minimum bucket size
or ratio of sizes, and transmitting a token status report.

(21) Appl. No.: **12/360,676**

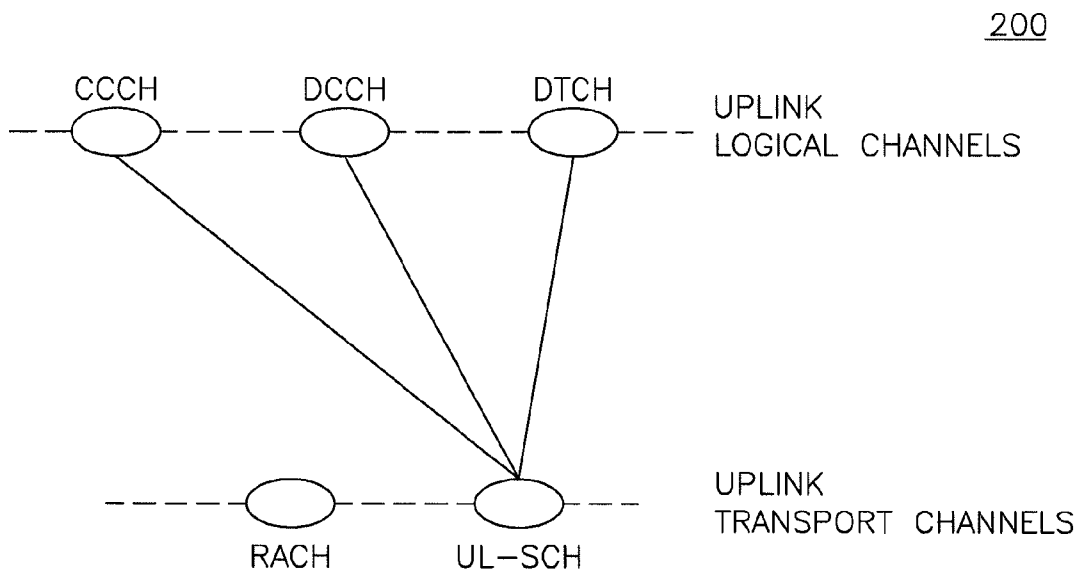
400





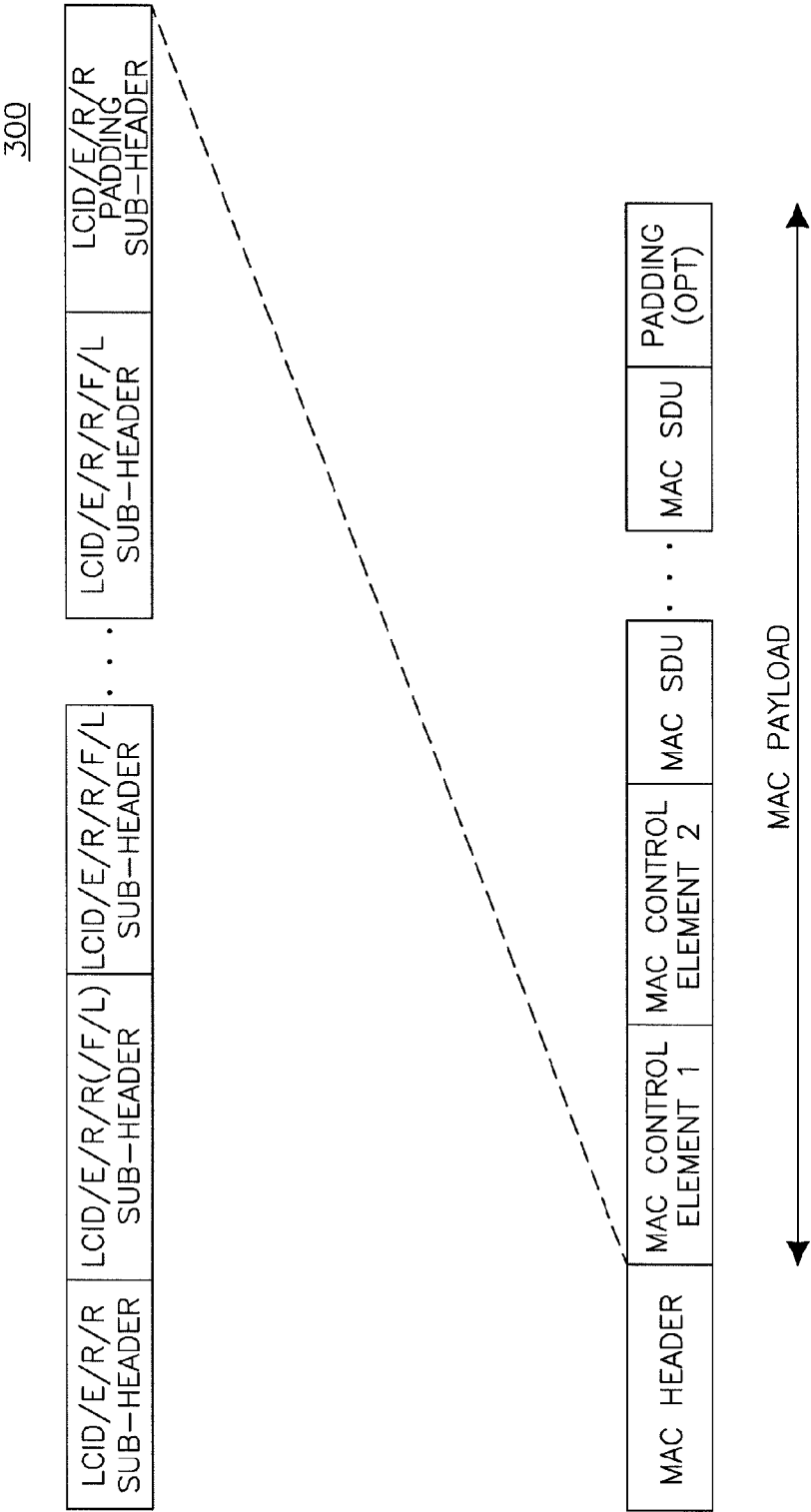
(PRIOR ART)

FIG. 1



(PRIOR ART)

FIG. 2



(PRIOR ART)
FIG. 3

400

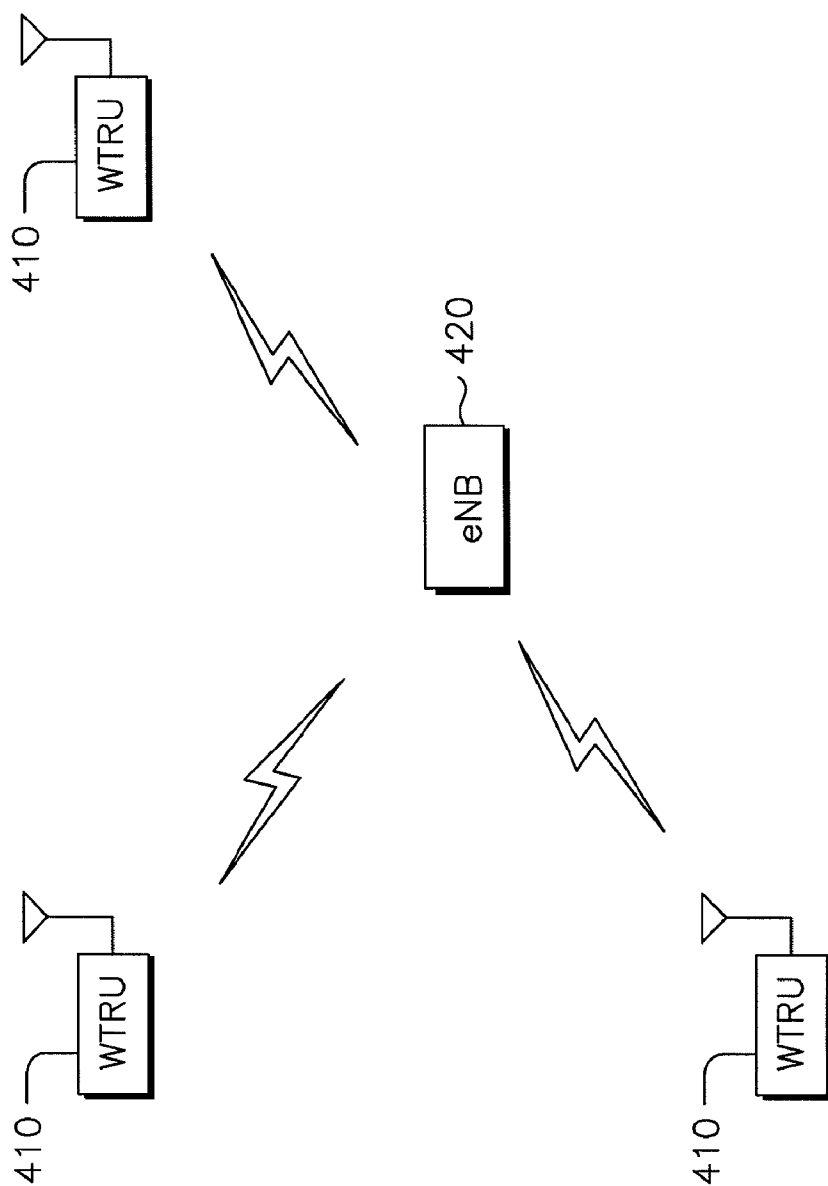


FIG. 4

500

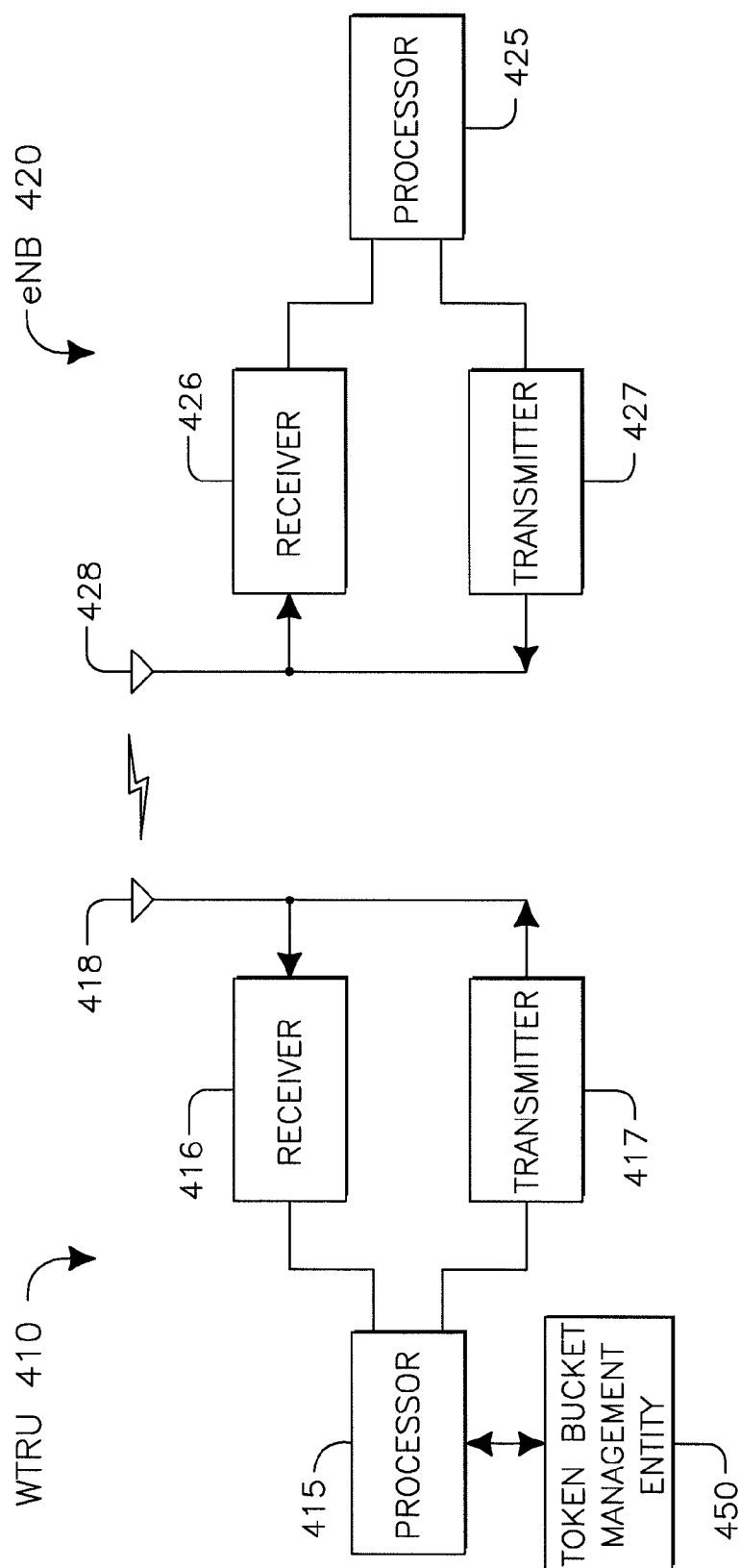
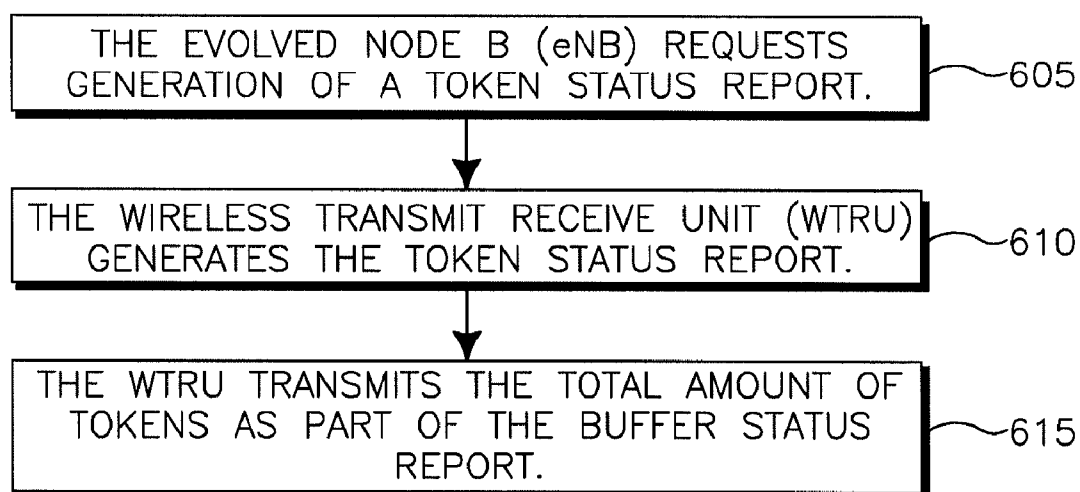


FIG. 5

**FIG. 6**

METHOD AND APPARATUS FOR INITIALIZING, PRESERVING, AND RECONFIGURING TOKEN BUCKETS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application Ser. No. 61/025,478, filed on Feb. 1, 2008, which is incorporated by reference as if fully set forth.

FIELD OF INVENTION

[0002] This application is related to wireless communications.

BACKGROUND

[0003] FIG. 1 shows a conventional user plane protocol stack **100** for the Long Term Evolution (LTE) architecture. The stack **100** includes a Packet Data Convergence Protocol (PDCP) layer, Radio Link Control (RLC) layer, Medium Access Control (MAC) layer, and physical (PHY) layer in a wireless transmit receive unit (WTRU) and corresponding layers in an evolved Node-B (eNB). The eNB is connected to a System Architecture Evolution (SAE) gateway by means of an S1-U interface.

[0004] According to the Third Generation Partnership Project (3GPP) standard, the LTE MAC sublayer supports mapping between logical channels and transport channels. The MAC sublayer supports multiplexing of MAC service data units (SDUs) from one or more logical channels onto transport blocks (TBs) for delivery to the PHY layer via transport channels. The MAC sublayer also supports demultiplexing of the MAC from one or more logical channels from the TBs delivered from the PHY layer via transport channels. In addition, the MAC sublayer supports logical channel prioritization and transport format selection.

[0005] A function of the MAC sublayer is the prioritization of the logical channels. The MAC entity may receive the MAC SDUs from different logical channels that are from the RLC layer. The MAC entity then multiplexes the SDUs onto one transport channel.

[0006] FIG. 2 shows the MAC multiplexing data from multiple logical channels onto one transport channel. The multiple logical channels may include dedicated traffic channels (DTCHs), dedicated control channels (DCCHs), or common control channels (CCCHs). In the example of FIG. 2, the one transport channel is shown as an uplink shared channel (UL-SCH).

[0007] The logical channel prioritization is applied when a new MAC transmission is performed. A radio resource control (RRC) sublayer in the WTRU controls scheduling of uplink data by assigning each logical channel a priority value. An increasing priority value indicates a lower priority level for the logical channel. Additionally, each logical channel is assigned a prioritized bit rate (PBR) and a maximum bit rate (MBR). A WTRU serves all logical channels in a decreasing priority order up to their configured PBR. If any resources remain, all logical channels are served in a decreasing priority order up to their configured MBR. If a MBR is not configured, then the logical channels are served until the data for the logical channel is exhausted or an uplink grant is exhausted, whichever comes first. The WTRU serves all logical channels configured with the same priority equally. The MAC control

elements for buffer status report (BSR), with the exception of padding BSR, have higher priority than user plane logical channels.

[0008] In the 3GPP standards, a WTRU has an uplink rate control function that manages sharing of uplink resources between radio bearers. The RRC controls the uplink rate control function by giving each bearer a priority and a PBR. The RRC also provides an MBR per guaranteed bit rate (GBR) bearer. The values signaled by the WTRU may not be related to the values signaled via interface S1 to the eNB. If more than one radio bearer has the same priority, then the WTRU serves these radio bearers equally.

[0009] There are several proposals discussing the details of logical channel prioritization and MAC multiplexing. The proposals agree on specifying the input parameters and the constraints for the output of the WTRU.

[0010] The proposals assume a token bucket model for the specification of input parameters. The PBR or the MBR is derived from the WTRU using the token rate, from a fixed size, or is signaled from the eNB. The PBR or the GBR does not limit the reported buffer status. The proposals utilize a token bucket model to describe the rate calculations, whereby each logical channel may have token buckets, related to the PBR and the MBR. The rates at which tokens are added to the buckets are the PBR and the MBR. A size of the token bucket may not exceed a predetermined maximum value.

[0011] One proposal describes a process for rate calculations or token bucket calculations. For each time increment of a bearer, T_j , that has a PBR, the PBR credit associated with the bearer j is incremented by a value of $T_j \times \text{PBR}_j$. If the bearer also has an MBR, then the MBR credit associated with the bearer j is incremented by a value of $T_j \times \text{MBR}_j$. If upper limits are set for the maximum PBR and if the accumulated values exceed the maximum values, then they are set to the maximum value. If the MBR credits are set for the bearer j , and if the accumulated values exceed the maximum values, then they are set to the maximum value. At each scheduling opportunity transmission time interval (TTI), where the WTRU is permitted to transmit a new data, the scheduler selects data from the highest priority bearer that has a non-empty buffer state and non-zero PBR credit. The scheduler may add to the TB data equal to the size of the buffer, the size of the PBR credit, or the available capacity of the TB, whichever is the smaller. The PBR credit and the MBR credit are decremented by the quantity of data assigned.

[0012] If the PBR credit of all bearers is zero and a space is available in the TB, then the scheduler accepts data from the highest priority bearer with a buffered data, up to the size of the available space in the TB or the MBR credit of the WTRU, whichever is the smaller. The MBR credit is decremented by the quantity of data that was accepted.

[0013] FIG. 3 shows a conventional MAC protocol data unit (PDU), which includes a MAC header, MAC control elements, MAC SDUs, and padding bits. Both the MAC header and the MAC SDUs may be of variable size. The MAC PDU header includes at least one MAC PDU sub-header, where each sub-header corresponds to either a MAC SDU, a MAC control element, or padding bits. The MAC layer may generate MAC control elements, such as the BSR control elements. The MAC control elements may be identified via specific values for the Logical Channel ID (LCID), as illustrated in Table 1 below.

TABLE 1

Index	LCID values
00000-yyyyy	Identity of the logical channel
yyyyy-11100	reserved
11101	Short Buffer Status Report
11110	Long Buffer Status Report
11111	Padding

[0014] The indexes 00000-yyyyy, shown in Table 1 above, may correspond to actual logical channels that have corresponding RLC entities. The remaining indexes may be used for identifying the MAC control elements, such as BSRs or padding.

[0015] According to the 3GPP standards, some services and functions of the LTE RLC sublayer include transfer of upper layer PDUs supporting acknowledged mode (AM) or unacknowledged mode (UM). The RLC sublayer also include transparent mode (TM) data transfer, error correction through automatic repeat request (ARQ), in-sequence delivery of upper layer PDUs (except at handover in the uplink), flow control between the eNB and WTRU, SDU discard, and RLC reset. Accordingly, the RLC supports three modes of operation: the AM, the UM, and the TM.

[0016] To ensure service continuity and to minimize service interruption, it would be beneficial to provide a method an apparatus to initialize token buckets, to preserve token buckets upon certain events, to reconfigure token buckets, and to communicate the status of token buckets using enhanced BSRs or new control elements.

SUMMARY

[0017] A method and an apparatus are provided to initialize token bucket to a non-zero value, to receive a scheduling grant, and to transmit data in response to receiving the grant.

[0018] A method and an apparatus are provided with initiate a medium access control (MAC) reset, determine whether to preserve a value of a token bucket or whether to re-initialize the value of the token bucket during the MAC reset, preserve the value of the token bucket in response to a determination to preserve the value of the token bucket, and re-initializing the value of the token bucket in response to a determination to re-initialize the value of the token bucket.

[0019] A method and an apparatus are provided to receive a request to generate a token status report, wherein the request is received as part of a medium access control (MAC) element, to generate the token status report in response to the request, and to transmit a total amount of tokens as part of a buffer status report.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] A more detailed understanding may be had from the following description, given by way of example in conjunction with the accompanying drawings wherein:

[0021] FIG. 1 shows a conventional LTE user plane protocol stack;

[0022] FIG. 2 is an illustration of conventional MAC mapping and multiplexing for uplink communication;

[0023] FIG. 3 shows a conventional structure of a MAC PDU according to a 3GPP standard specification;

[0024] FIG. 4 shows an example wireless communication system including a plurality of WTRUs and an eNB;

[0025] FIG. 5 is an example functional block diagram of a WTRU and the eNB of the wireless communication system shown in FIG. 4; and

[0026] FIG. 6 is an example flow diagram of a method of processing a token status report.

DETAILED DESCRIPTION

[0027] When referred to hereafter, the terminology “wireless transmit/receive unit (WTRU)” includes but is not limited to a user equipment (UE), a mobile station, a fixed or mobile subscriber unit, a pager, a cellular telephone, a personal digital assistant (PDA), a computer, or any other type of user device capable of operating in a wireless environment. When referred to hereafter, the terminology “base station” includes but is not limited to a Node-B, a site controller, an access point (AP), or any other type of interfacing device capable of operating in a wireless environment.

[0028] Hereinafter, the RLC PDUs or RLC PDU segments may be considered as equivalent to MAC SDUs. Further, when referring to updating the token/credit bucket, it may be considered as refer to subtracting an amount of tokens/credits from the bucket with the amount corresponding to the MAC SDU size(s), or adding tokens/credits at the bucket update interval. Also, token bucket, or credit calculations, may be considered as equivalent to data rate calculations or rate control calculations. Furthermore, the token/credit bucket will be referred to hereafter as a token bucket.

[0029] FIG. 4 shows an example wireless communication system 400 including a plurality of WTRUs 410 and an eNB 420. As shown in FIG. 4, the WTRUs 410 are in communication with the eNB 420. Although three WTRUs 410 and one eNB 420 are shown in FIG. 1, it should be noted that any combination of wireless and wired devices may be included in the wireless communication system 400.

[0030] FIG. 5 is a functional block diagram 500 of a WTRU 410 and the eNB 420 of the wireless communication system 400 of FIG. 4. As shown in FIG. 5, the WTRU 410 is in communication with the eNB 420 and both may be configured to initialize, preserve, and reconfigure the token buckets.

[0031] In addition to the components that may be found in a typical WTRU, the WTRU 410 includes a processor 415, a receiver 416, a transmitter 417, and an antenna 418. The processor 415 is configured to perform a method for initializing, preserving, and reconfiguring the token buckets. The receiver 416 and the transmitter 417 are in communication with the processor 415. The antenna 418 is in communication with both the receiver 416 and the transmitter 417 to facilitate the transmission and reception of wireless data. The WTRU 410 also includes a token bucket management entity 450, which will be described in more detail below.

[0032] In addition to the components that may be found in a typical eNB, the eNB 420 includes a processor 425, a receiver 426, a transmitter 427, and an antenna 428. The processor 425 is configured to perform a method for initializing, preserving, and reconfiguring the token buckets. The receiver 426 and the transmitter 427 are in communication with the processor 425. The antenna 428 is in communication with both the receiver 426 and the transmitter 427 to facilitate the transmission and reception of wireless data.

[0033] Continuing to refer to FIG. 5, the token bucket management entity 450 processes token buckets. That is the token bucket management entity 450 determines if there is a initial-

ization, handover, reconfiguration, or reset event. If there is reset event, then the token bucket management entity **450** performs a bucket reset function. If there is bucket reconfiguration, then the token bucket management entity **450** performs a bucket reconfiguration function.

[0034] The WTRU **410** may be configured to initialize the PBR buckets to the amount of tokens corresponding to one token update time interval. Additionally, the WTRU **410** may also be configured to initialize the MBR buckets to the amount of tokens corresponding to one token update time interval. For example, the PBR may be initialized to a value of $T_j \times PBR_j$, while the MBR may be initialized to a value of $T_j \times MBR_j$.

[0035] Initializing token buckets based on a non-zero value allows the WTRU **410** to minimize segmentation and to begin transmitting information when it receives the scheduling grant rather than incurring a delay due to the need to accumulate more tokens.

[0036] Alternatively, the maximum value of the bucket may be the PBR or MBR at the update rate multiplied by a number of updates. The bucket update rate multiplied by the number of updates may be considered the bucket duration. The WTRU **410** may be configured to initialize the PBR and the MBR buckets to the amount of tokens corresponding to N token update time intervals. For example, the PBR may be initialized to the value of $N \times T_j \times PBR_j$ and the MBR may be initialized to the value of $N \times T_j \times MBR_j$, where T_j is a token updating time increment for j bearer and N is an integer that may be configurable. The value of N may be configured using the RRC. Alternatively, N_j may be used instead of N.

[0037] The WTRU **410** may also be configured to initialize the PBR and the MBR buckets to their maximum values. The bucket may be initialized to a maximum number of tokens the bucket is allowed to accumulate. Once the bucket reaches this level it is not increased in size at subsequent bucket update intervals. The bucket update rate multiplied by the maximum number of updates, N, may be considered the maximum bucket duration.

[0038] The WTRU **410** may be configured to initialize the PBR and the MBR buckets to zero. Alternatively, the WTRU **410** may be configured to initialize the PBR and the MBR buckets to predetermined values indicated using RRC information elements (IEs) carried in a RRC message. These IE's may include an initial value or may include the PBR and/or MBR token update size, the update periodicity, and the maximum bucket size or bucket duration.

[0039] The token buckets may be preserved at handover or at a MAC reset, and the WTRU **410** may maintain the token bucket values during handover or a MAC reset. The last value of the PBR and/or MBR bucket determined in the previous cell is applied as the initial value in the new cell. Service continuity and minimization of interruption time may be achieved from this. The WTRU **410** may be configured to not re-initialize the token bucket values during handover or during a MAC reset or the WTRU **410** may be configured to maintain the token bucket values during all handover events.

[0040] The WTRU **410** may be configured to use a predetermined RRC indication to indicate whether the WTRU **410** may maintain a token bucket value or re-initialize the token bucket value during a MAC reset. The predetermined RRC indication may either be a RRC primitive or a RRC IE. In addition, the maintenance of the token buckets may be applied selectively to logical channels. For example, a logical

channel's token bucket value may be maintained while the token bucket value of another logical channel is re-initialized.

[0041] Alternatively, if the token buckets are re-initialized during the handover or the MAC reset, the WTRU **410** may be reconfigured to re-initialize the token buckets to a non-zero value during the handover or the MAC reset. For example the bucket may be initialized to $PBR \times T_j$ or $N \times PBR \times T_j$.

[0042] The WTRU **410** may move from one cell or eNB **420** to another during intra-LTE handover. During the handover operation, the MAC entity may be reset or reconfigured. The MAC entity may also be reset or reconfigured when there is a radio link failure, which may occur due to an RLC reset or an upper layer reset. In each of these cases the token bucket may be reset or preserved as described above.

[0043] The PBR or the MBR of the WTRU **410** may be reconfigured by the eNB **420** at any time or during a handover. Additionally, the reconfiguration of the PBR and/or the MBR of the WTRU **410** may be performed as part of a MAC reconfiguration procedure.

[0044] As a result, the WTRU **410** may need to re-initialize its token buckets in accordance with a new configured bit rate. The token bucket management entity **450** may be located at RRC layer or may be located at MAC layer (not shown). The computation of the token bucket sizes may be performed by the MAC or by the RRC. If the token bucket sizes are performed by the RRC, the RRC may then reconfigure to the MAC. This may require signaling of primitives between the MAC and the RRC.

[0045] For example, the MAC provides the RRC with its current bucket values upon reconfiguration. Then, the RRC computes new bucket values using the signaled information and reconfiguration parameters. The RRC signals the new bucket values to the MAC.

[0046] Upon the PBR or the MBR reconfiguration, the WTRU **410** may be configured to re-initialize the token buckets. The WTRU **410** may be configured to compare the size of token buckets, and the WTRU **410** may be configured to re-adjust or scale the size of the token buckets.

[0047] Upon PBR or MBR reconfiguration, the WTRU **410** may reinitialize the token buckets by initializing the PBR and the MBR buckets to the maximum value of the PBR and the MBR bucket, respectively. The WTRU **410** may be configured to initialize the PBR and the MBR buckets to the amount of tokens corresponding to N token update time intervals.

[0048] For example, the PBR may be initialized to the value of $N \times T_j \times PBR_j$ and the MBR may be initialized to the value of $N \times T_j \times MBR_j$, where T_j is the token updating time increment for j bearers and N is an integer that may be configurable. The value of N may be configured using the RRC. Alternatively, N_j may be used instead of N.

[0049] Alternatively, the WTRU **410** may initialize the PBR and MBR buckets to their maximum values, or initialize the PBR and MBR buckets to zero. Alternatively, the WTRU **410** may initialize the PBR and MBR buckets to predetermined values indicated using RRC IEs carried in a RRC message.

[0050] The WTRU **410** may initialize the PBR and the MBR token buckets to the amount of tokens corresponding to one token update time interval. For example, the PBR may be initialized to a value of $T_j \times PBR_j$, and the MBR may be initialized to a value of $T_j \times MBR_j$, where T_j is a token updating time increment.

[0051] Upon the PBR or the MBR reconfiguration, the RRC entity of the WTRU **410** or the MAC entity of the

WTRU 410 may be configured to compare the current size of a token bucket with a newly specified maximum bucket size. If the current token bucket size is smaller than the new maximum bucket size, then the WTRU 410 may be configured to maintain the current bucket size. If the current token bucket size is larger than the new maximum bucket size, then the WTRU 410 may be configured to set the bucket size to the new maximum bucket size. The setting of the bucket size to the minimum of the current bucket size and the new maximum bucket size may also be referred to as token bucket clamping.

[0052] Upon the PBR or the MBR reconfiguration, the RRC entity of the WTRU 410 or the MAC entity of the WTRU 410 may be configured to readjust or scale the token bucket size in proportion to the ratio of the maximum bucket size. For example, the number of new tokens is a function of the number of old tokens times the ratio of the new maximum bucket size to the old maximum bucket size. Then, the token bucket is scaled relative to the old to new maximum bucket size ratio.

[0053] Alternatively, the token bucket size may be scaled in proportion to the ratio of bit rates. For example, the number of new tokens is a function of the number of old tokens times the ratio of a new bit rate to the old bit rate. Then, the token bucket is scaled to the old to new bit rate ratio.

[0054] FIG. 6 shows a diagram for the token status report. The eNB 420 transmits a request to the WTRU 410 requesting the WTRU 410 to generate the token status report (step 605). The request from the eNB 420 may be received as part of a new MAC control element or an RRC IE in an RRC message. The event that triggers the eNB 420 to transmit the request to the WTRU 410 is determined by the eNB implementation. For example, the triggering event may be as defined below.

[0055] The WTRU 410 may be configured to generate the token status report (step 610) requested by the eNB 420. The WTRU 410 may be configured to transmit total amount of tokens upon a triggering event. This information may be transferred as part of Buffer Status Report (BSR) (step 615). The WTRU 410 may be configured to transmit a token status report containing the current PBR or MBR token bucket values of the WTRU 410. The token status report may also contain any information related to or derived from the token buckets.

[0056] The token status reports may be included as an enhancement to the BSR. For example, the enhanced BSR may include information of the token buckets. The token status reports may be included in a new MAC control element. Alternatively, the token status reports may be included in an RRC IE contained in a RRC message. The WTRU 410 may generate the token status report based on an internal trigger.

[0057] The internal trigger and the triggering event may be any one of the following: handover, MAC reset, a reset at specific layer, MAC reconfiguration, a reconfiguration at a specific layer, and a request from the eNB 420. The triggering event or the internal trigger may be periodical, or at any another predefined time or event.

[0058] The WTRU 410 may be configured to generate a token status report in response to a request from the eNB 420. The eNB 420 is able to optimize scheduling operations using the capability of the token status report. The eNB 420 may be configured to synchronize its local computations with the computations of the WTRU 410 using the token status report

capability. Furthermore, the eNB 420 may reconfigure bit rate parameters or token bucket values after receiving the token status report capability.

[0059] The token status report may include the absolute size of the PBR or the MBR buckets, the size of the PBR or the MBR buckets relative to their maximum bucket size, a bit to indicate whether the PBR or MBR bucket is full or not, a bit to indicate whether PBR or MBR bucket is empty or not, an indication of whether the WTRU 410 is PBR or MBR limited, or any other information related to or derived from the PBR or the MBR tokens. The information may be aggregated across multiple logical channels or buckets, or may be aggregated on a per logical channel basis.

[0060] The BSR is enhanced to provide token status report information by extending the BSR. For example, the BSR may include an extension flag. Alternatively, the WTRU 410 may be configured to transmit the total amount of accumulated tokens, PBR or MBR tokens, to the eNB 420 as part of the BSR to assist the eNB 420 with scheduling. Alternatively, the WTRU 410 may be configured to compare accumulated tokens with a predefined threshold, which may be configured by an RRC IE. Then, the WTRU 410 may be configured to report a value of 1 if the amount of accumulated tokens meets the predefined threshold or a value of 0 if the amount of accumulated tokens fails to meet the predefined threshold.

[0061] Although features and elements are described above in particular combinations, each feature or element can be used alone without the other features and elements or in various combinations with or without other features and elements. The methods or flow charts provided herein may be implemented in a computer program, software, or firmware incorporated in a computer-readable storage medium for execution by a general purpose computer or a processor. Examples of computer-readable storage mediums include a read only memory (ROM), a random access memory (RAM), a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, and optical media such as CD-ROM disks, and digital versatile disks (DVDs).

[0062] Suitable processors include, by way of example, a general purpose processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) circuits, any other type of integrated circuit (IC), and/or a state machine.

[0063] A processor in association with software may be used to implement a radio frequency transceiver for use in a wireless transmit receive unit (WTRU), user equipment (UE), terminal, base station, radio network controller (RNC), or any host computer. The WTRU may be used in conjunction with modules, implemented in hardware and/or software, such as a camera, a video camera module, a videophone, a speakerphone, a vibration device, a speaker, a microphone, a television transceiver, a hands free headset, a keyboard, a Bluetooth® module, a frequency modulated (FM) radio unit, a liquid crystal display (LCD) display unit, an organic light-emitting diode (OLED) display unit, a digital music player, a media player, a video game player module, an Internet browser, and/or any wireless local area network (WLAN) or Ultra Wide Band (UWB) module.

What is claimed is:

1. A wireless transmit receive unit (WTRU) comprising:
a processor configured to initialize a token bucket to a non-zero value;
a receiver configured to receive a scheduling grant; and
a transmitter configured to transmit data in response to receiving the grant.
2. The WTRU as in claim 1 wherein the processor is further configured to initialize the token bucket based on a maximum value of a prioritized bit rate (PBR).
3. The WTRU as in claim 2 wherein the PBR is initialized based on an amount of the tokens according to a token update time interval.
4. A wireless transmit receive unit (WTRU), the WTRU comprising:
a processor configured to:
initiate a medium access control (MAC) reset; and
preserve a value of a token bucket during the MAC reset.
5. The WTRU as in claim 4 wherein the processor is further configured to initiate the MAC reset in response to a handover command.
6. The WTRU as in claim 5 wherein the value of the token bucket is preserved until a logical channel has been established.
7. A wireless transmit receive unit (WTRU) comprising:
a processor configured to trigger a handover, reset a medium access control (MAC) layer during the handover, and reset a token bucket to a non-zero value.
8. The WTRU as in claim 7 wherein the processor is configured to reset the token buckets based on a ratio of a new maximum bucket size to an old maximum bucket size.
9. The WTRU as in claim 8 wherein the processor is further configured to compare a current size of the token bucket with a newly predetermined maximum bucket size, in response to a reconfiguration.

10. The WTRU as in claim 9 wherein if the current size is smaller than the maximum bucket size, then the processor maintains the current size of the token bucket.

11. The WTRU as in claim 9 wherein if the current size is larger than the maximum bucket size, then the processor utilizes the token bucket size to the maximum size.

12. The WTRU as in claim 9 wherein the processor is further configured to readjust the token buckets based on a ratio of a new maximum size and an old maximum size.

13. The WTRU as in claim 12, wherein the processor readjusts the token buckets based on a ratio of a new bit rate and an old bit rate.

14. The WTRU as in claim 9 wherein the processor is further configured to re-initialize the token buckets based on the maximum value of a prioritized bit rate (PBR).

15. A wireless transmit receive unit (WTRU) comprising:
a receiver configured to receive a request to generate a token status report, wherein the request is received as part of a medium access control (MAC) element;
a processor configured to generate the token status report in response to the request; and
a transmitter configured to transmit a total amount of tokens as part of a buffer status report.

16. The WTRU as in claim 15 wherein the request is received in response to a MAC reset.

17. The WTRU as in claim 15 wherein the request is received in response to a MAC reconfiguration.

18. The WTRU as in claim 15 wherein the request is received in response to a reset of a layer.

19. The WTRU as in claim 15 wherein the request is received in response to reconfiguration of a layer.

20. The WTRU as in claim 15 wherein the transmitter is further configured to transmit the token status report containing prioritized bit rate (PBR) token bucket values.

* * * * *