Performance Evaluation of the Radio Link Control Protocol in 3G UMTS

Jun Li
School of Mathematics & Statistics, Carleton University, 1125 Colonel By Dr. Ottawa, Ontario, Canada K1S 5B6
Phone: (613)520-2600 ext. 1806 Fax: (613)520-3536 e-mail: jli@math.carleton.ca

Delfin Y. Montuno
Nortel Networks, PO Box 3511, Station C, Ottawa, Ontario, Canada K1Y 4H7
Phone: (613)763-3116 Fax: (613)768-1140 e-mail: delfin@nortelnetworks.com

Jianyu Wang
Nortel Networks (China) Limited, 10/F Nortel Networks Tower Sun Dong An Plaza, No 138 Wang Fu Jing Street, Beijing, China
Phone: (8610)6510-8259 Fax: (8610)6528-7657 e-mail: jaywang@nortelnetworks.com

Yiqiang Q. Zhao
School of Mathematics & Statistics, Carleton University, 1125 Colonel By Dr. Ottawa, Ontario, Canada K1S 5B6
Phone: (613)520-2600 ext. 2131 Fax: (613)520-3536 e-mail: zhao@math.carleton.ca

Abstract— **With the rapid advances in wireless communications and internetworking, Universal Mobile Telecommunications Systems (UMTS), playing a main role in the emerging third-generation (3G) wireless networks, have aimed to provision high-speed data integrated with voice services. Since today's data traffic is largely based on Transmission Control/Internet Protocol (TCP/IP) suite, UMTS networks have employed a Radio Link Control (RLC) protocol to support reliable upper layer protocols such as TCP. RLC employs sophisticated retransmission schemes to perform partial error recovery at the link layer, thus hiding transmission errors from upper layers. Since TCP will treat every segment loss as a sign of congestion, the above mentioned schemes can reduce the chances of a mistaken invocation of the TCP congestion control mechanism, which results in performance degradation. Therefore, it is critical to optimize the RLC protocol to achieve the best performance. In this paper, the RLC performance, in terms of the Service Data Unit (SDU) average delay, the RLC throughput, and the RLC goodput in a scenario of bulk data transfer from a Radio Network Controller (RNC) to a User Equipment (UE), is obtained using simulation. Two important retransmission triggers, the poll timer and the status period timer, are evaluated under various Block Error Rates (BLERs). The performance trends against the poll timer and the status period timer settings are analysed and summarized, respectively, for use in the design of optimized 3G UMTS systems for data services.**

Keywords: 3G wireless networks, Universal Mobile Telecommunication Systems (UMTS), Radio Link Control protocol (RLC), performance evaluation, error correction

I. Introduction

Due to rapid advances in the areas of wireless communications and the global Internet connection, provision of data services for Internet applications over wireless networks is gaining importance. The two accepted systems of the third generation (3G) wireless networks that are aiming at provision of high-speed cellular data services, UMTS and CDMA2000, are both using link-layer schemes to improve the performance of such networks. UMTS employs the Radio Link Control (RLC) protocol [rlc02] for flow control and error recovery based on the Multiple Rejects ARQ (MR-ARQ) mechanism [YPM00]. RLC hides transmission errors from upper layers to reduce the chances of mistaken invocations of the upper layer's (usually TCP) congestion control mechanisms. As a result, the performance of data traffic over the air interface is improved. Since the RLC protocol has many features and options that make the protocol very flexible and adaptive for UMTS operators with many parameter settings to choose from in various radio environments, it is important to understand the protocol performance over the UMTS specific physical link. A few studies [ZS02], [XXC$^+$02], [XCX$^+$02] have been conducted to optimise the settings of the parameters based on the web browsing behaviours of the upper layers' data traffic model. However, the performance evaluation of RLC needs to be further investigated under different data traffic models. In this paper, the performance evaluation of polling and status transmission mechanisms based on a heavy-load FTP traffic model is conducted using simulation. Based on the quantitative results, the performance trends against poll timer and status period timer settings under various BLERs are analysed and summarized respectively for guiding the design of optimized 3G UMTS systems for data services.

II. RLC Protocol in UMTS

The UMTS network architecture consists of three components: The Core Network (CN), the UMTS Terrestrial Radio Access Network (UTRAN) and the User Equipment (UE). In UTRAN, three layers are specified. The physical layer, which uses WCDMA on the radio link interface. The link layer contains Medium Access Control (MAC), RLC, Broadcast/Multicast Control (BMC), and Packet Data Convergence Protocol (PDCP) sublayers, of which BMC and PDCP are in the user plane. The Radio Resource Control (RRC) layer is in the control plane only.

Spread spectrum and fast power control mechanisms are employed by the physical layer to transform the radio frames between the Node B and the UE. One radio frame, which may include several RLC PDUs, is sent during each Transmission Time Interval (TTI). Through an attribute of the transport format, the MAC layer decides which PDUs to send in each TTI. Three types of services are provided by the RLC

This research is supported by Nortel Networks, NCIT and MITACS.

protocol, including Transparent Mode (TM), Unacknowledged Mode (UM), and Acknowledged Mode (AM). The RRC is responsible for the transport format and also determines the transmission mode of the RLC. Since AM is designed to provide reliable transmission of packet data, it is the mode used in our simulation to conduct the protocol performance evaluation.

The RLC layer performs segmentation/reassembly of higher layer PDUs, or RLC SDUs, into/from smaller RLC PDUs. To maintain the order of RLC SDUs, in-sequence delivery option is configured. There are mainly two types of RLC PDUs:

- **Acknowledged Mode Data PDU (AMD PDUs)** contains user data and a sequence number, and a polling bit to poll the receiver among possibly other information.
- **Status Information Control PDU (status PDU)** contains status information about the receiver and the transmitter, including missing blocks, RLC SDU discard notification, window size among possible others.

Segmentation/concatenation, transfer of user data, error correction, in-sequence delivery of RLC SDUs, and duplicated detection are some of the basic RLC functions. The three RLC operations needed to make flow control and error control more efficiently are polling mechanism, status transmission mechanism, and SDU discard mechanism.

Polling mechanism

In case of erroneous or lost AMD PDUs, retransmissions are conducted by the sender upon reception of one or more status PDUs from the receiver. Sending a status PDU can be triggered by either the sender or the receiver. For the polling mechanism, the sender sends the polling request and the receiver responds by sending one or more status PDUs back to the sender. For the status transmission mechanism, sending a status PDU is triggered by the receiver.

The polling mechanism is implemented by setting the polling bit in one AMD PDU's header. There are eight triggers kept in the transmitting side and used to set the polling bit when the polling mechanism is initiated. They include: (1) $poll\ timer$, (2) $poll\ periodic\ timer$, (3) $poll\ prohibit\ timer$, (4) $last\ PDU\ in\ buffer$, (5) $last\ PDU\ in\ retransmission\ buffer$, (6) $poll\ PDU\ PDU$, (7) $poll\ SDU\ SDU$, (8) $window\ based\ polling$. Which of the triggers shall be used is/are decided by upper layers.

The buffer-based triggers, $last\ PDU\ in\ buffer$ and $last\ PDU\ in\ retransmission\ buffer$ can prevent deadlock of the RLC entities [ZS02]. $Poll\ periodic\ timer$ polls the receiver periodically, and the timer value controls the polling frequency. It starts when a session is created. Every time the timer expires, a polling is set in the header of an AMD PDU to be transmitted if it is available in the transmission buffer, and the timer is reset. $Poll\ timer$ starts when a polling contained in an AMD PDU is submitted to the lower layer in the sender, and should be reset each time a polling is set and submitted to the lower layer. It thus can guarantee to poll the receiver at least periodically with the period being the $poll\ timer$ value. $Poll\ prohibit\ timer$ controls the frequency of polling the receiver by prohibiting transmissions of polls with a certain period. So when multiple polling bit settings are triggered before $poll\ prohibit\ timer$ expires, the settings are deferred till $poll\ prohibit\ timer$ expires. $Poll\ PDU\ PDU$ will poll the receiver when the number of sent AMD PDUs after the last polling reaches a certain number. $Poll\ SDU\ SDU$ polls the receiver when the number of sent RLC SDUs after the last polling reaches a certain number. The trigger $window\ based\ polling$ polls the receiver each time the AMD PDUs exceeds a certain percentage in the transmission window.

Status transmission mechanism

Another way of sending status PDU(s) is driven by the receiver, instead of the sender. Such mechanism can allow the receiver to send status PDU(s) more aggressively. The $detection\ of\ missing\ PDU(s)$ trigger can make the receiver send status PDU(s) to request for retransmission when one or more missing PDUs are detected by the receiver. It is an effective trigger to shorten the RLC SDU delay. The $status\ period\ timer$ trigger requests the receiver to send status PDU(s) back to the sender periodically. The Estimated PDU Counter (EPC) mechanism makes the receiver send status PDU(s) more actively by estimating the time needed to recover the erroneous AMD PDUs included in the latest status PDU. The $status\ prohibit\ timer$ timer controls the frequency of sending status PDU(s) to the sender. All of the triggers of the status transmission mechanism are set optionally in the RLC protocol.

1) **Detection of missing AMD PDU(s)**
   Once the receiver detects one or several missing AMD PDUs, one or more status PDU(s) will be sent to the sender, requesting the retransmission of all erroneous/missing AMD PDUs.
2) **Timer-based status transmission**
   The receiver sends status PDUs to the sender periodically. The timer $status\ period\ timer$ controls frequency of sending status PDUs. This timer is started when the RLC entities are created, and when it expires, the status PDU(s) is transmitted and the timer is reset.
3) **The EPC mechanism**
   The EPC mechanism prevents excessive exchanges of status PDUs. $EPC\ timer$ controls the period of scheduling the transmissions of status PDUs in this mechanism, and is set each time the first status PDU is submitted to the lower layer. Then, the state variable $VR(EP)$ is set to be the number of AMD PDUs to be recovered. If not all AMD PDUs requested for retransmission have been received when $VR(EP)$ equals to zero, a new status PDU is sent by the receiver. However, if another transmission of STATUS report is triggered while $VR(EP)$ is not equal to zero, the status PDU will be delayed until it is decremented to zero.
4) $Status\ prohibit\ timer$
   The timer $status\ prohibit\ timer$ prohibits the receiver from sending consecutive status PDUs. It is started each time the first status PDU is submitted to the lower layer. The receiver is not allowed to transmit status PDUs before the timer expires.

SDU discard mechanism

RLC SDU discard mechanism allows the sender to discard the AMD PDUs associated with a SDU from the transmission buffer and the retransmission buffer. It is initiated when the transmission of the AMD PDUs does not succeed within a period of time or for a number of transmission attempts. This mechanism can avoid buffer overflow in the RLC layer and reduce the maximum transmission delay. There are two RLC SDU discard functions that can be configured according to the QoS requirements: Timer based discard with explicit signaling, and RLC SDU discard after $MAXDAT$ number of transmissions.

1) **Timer based discard with explicit signaling**
   This function discards a RLC SDU after its corresponding timer, called $timer\_discard$ expires. This makes the SDU discard function insensitive to variations of the channel data rate and error rate, thus controls the maximum delay each RLC SDU may experience. The SDU loss rate, however, increases as RLC SDUs are discarded. The function works as follows. For each RLC SDU received from upper layers, a timer $timer\_discard$ is started. When $timer\_discard$ of a SDU expires, the sender discards the SDU, and uses explicit signaling to inform the receiver of the discarded SDU.
2) **RLC SDU discard after $MAXDAT$ number of transmissions**
   An alternative RLC SDU discard function is "SDU discard after $MAXDAT$ number of transmissions." It tries to keep the SDU loss rate constant. However, its delay performance is variable and dependent on the channel condition. If $VT(DAT)$ of an AMD PDU reaches the value $MAXDAT$, the sender discards all RLC SDUs contained in the AMD PDU, and uses explicit signaling to inform the receiver about the discarded AMD PDU by sending a status PDU.

III. Simulation Description

Fig. 1 illustrates the RLC details implemented with OPNET Modeler by OPNET Technologies Inc. [Inc02]. The upper layer Protocol Data Units (RLC SDUs) generated by the FTP Traffic module, which is included in the transmitting side and can generate incoming traffic, are stored in the SDU buffer. Then these SDUs are segmented or concatenated into RLC PDUs, added by RLC PDU headers, and put into the multiplexer (MUX). Meanwhile, a copy of theses PDUs is stored in the retransmission PDU buffer. The multiplexer determines transmission order of PDUs giving the PDUs to be retransmitted higher priority than the PDUs waiting for transmission for the first time up to a maximum number of retransmissions configured by RRC. During each TTI, only the allowed maximum number of PDUs can be sent from the multiplexer to the lower layer through the Transmission buffer. A PDU in the retransmission buffer can be transmitted to the lower layer only when the missing message corresponding to that PDU is received. Before a PDU goes to the lower layer, all polling triggers will be checked to determine if the polling bit in that PDU's header needs to be set. For the receiving side, a correctly received PDU is first classified into a data PDU or status PDU. If it is of the data type, the header is removed. Reassembly is conducted if all PDUs corresponding to a complete SDU is available, and the condition of delivery to the upper layer is checked. If one or more SDUs are delivered to the upper layer, all performance variables are updated accordingly. If the correctly received PDU is a status PDU, the protocol control information will be analysed, and the correspondent actions, such as retransmission of some missing PDUs, shrinking the retransmission buffer, giving up of recovering a PDU, and so on, are performed.

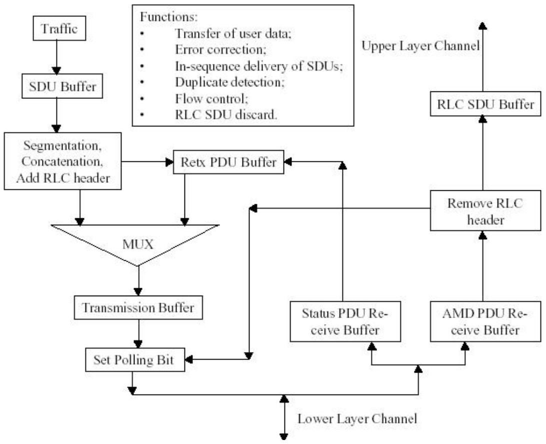

Fig. 1. RLC Process Model of Simulation

The traffic arrival process model and traffic statistics are the important factors for considering the RLC performance and parameters' settings. In this study, we use the traffic model for bulk data transfer (heavy-loaded FTP traffic) to study the performance of the RLC protocol. The traffic coming from the upper layer of RNC is modeled as a determinate process with a constant mean packets. The arrival rate of the upper layer packets is set to be the same as the data rate of the wireless forward link.

The physical layer of UMTS introduces transmission errors with the Block Error Rate (BLER) depending on the link characteristics. BLER is the probability for a RLC PDU being incorrectly received. It is configured before starting a simulation and is not changed during the simulation. To simplify the problem, this study assumes that a RLC PDU being incorrectly received is independent of the status of other PDUs and the error distribution is uniform. More sophisticated and realistic transmission error models with bursty nature will be investigated in our future studies.

The IMT-2000 standard requires the data rate of 3G wireless networks has no less than 384 kbps capability for pedestrian (micro-cell) and low speed vehicular environment, and 2 Mbps for indoor office using wide-band 1.6 MHz carrier [GG00]. In this study, we use 384 kbps as the data rate of the physical channel to simulate a micro-cell wireless environment. The data rate of 2 Mbps for indoor office communications will be included in our future research work.

In our study of the RLC performance evaluation, we focus on the three RLC performance metrics defined as follows.

1) **The RLC SDU delay of in-sequence delivery** is the time at which a RLC SDU reaches the RLC layer of the sender till the time at which it is correctly reassembled

and delivered by the RLC of the receiver to the RLC upper layer. It includes three parts: The queueing delay at RNC, the out-of-sequence delivery delay (analysed in [EL02]), and the re-sequencing delay at UE.

2) **The RLC throughput** is the total data bits transmitted on the forward link at a unit of time. It includes transmissions of data PDUs, either transmitted for the first time or for retransmissions, and STATUS PDUs.
3) **The RLC goodput** is the total useful information bits received correctly and delivered to the upper layer by the receiver. So neither duplicate AMD PDUs nor STATUS PDUs are counted for the RLC goodput.

IV. Simulation Results

The parameters used in the RLC performance evaluation are summarized in TABLE I.

Parameters	**Values**
RLC SDU size	1216 bytes (or 32 PDUs)
SDU traffic model	Bulk data transfer with FTP
SDU delivery	In-sequence delivery
PDU size	40 bytes
TTI	20 ms
Physical layer data rate	384 kbps (or 24 PDUs per TTI)
PDU one-way delay	100 ms
SDU discard function	Max. number of transmissions based discard
Maximum number of transmissions	5 times
Block error rate	0%, 10%, 15%, or 25%
Transmission window size	2047 PDUs
Receiving window size	2047 PDUs

TABLE I

SIMULATION PARAMETERS

We have done intensive simulations to evaluate the RLC protocol performance by varying the values of two timers: poll timer and status period timer under different block error rates. We will present the effects of these two timers on the performance of the RLC protocol.

A. *Effect of Status Period Timer*

To evaluate the effect of status period timer, we enable this trigger and turn off all others, including "detecting missing AMD PDU(s)," "EPC mechanism" and "status prohibit" triggers.

1) RLC SDU Delay: Fig. 2 shows the effects of the status period timer on the SDU average delay under various block error rates. Usually, the in-sequence delivery delay a SDU experiences includes three parts as described above.

The values of the status period timer affect the interval between the retransmissions of lost PDUs that can influence the SDU delay of out-of-sequence delivery, which is the second part of a SDU delay of in-sequence delivery as described above. When the BLER is zero, no PDUs are lost. All RLC PDUs are transmitted only once, so no effect of the status period timer is observed. Since no retransmission is performed, and each RLC SDU can be delivered to upper layers as soon as its segmented PDUs are received completely, there is no re-sequencing delay of a RLC SDU at UE (the receiver). Meanwhile, the upper layers' traffic has a constant rate with 24 SDUs per TTI, which can be transferred to the lower layer during the same TTI due to no RLC PDUs requesting for retransmission. Then no queueing delay happens for any RLC PDU in the transmission buffer of RNC. Hence, SDU delay is the summation of the PDU one-way latency time and one TTI time. As shown in Fig. 2, the simulated result of SDU delay with BLER 0% is a little more than 120 ms, which perfectly agrees with our expected result analysed above. Since our expected optimised timers values happen around the TLC Round Trip Time (RTT), a non-uniform X-axis of timer values are used for comparison.

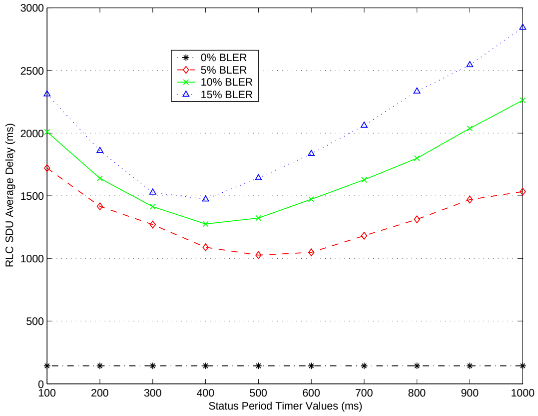

Fig. 2. Effect on SDU Average Delay with Different BLERs

When the block error rate is non-zero, the SDU average delay is not a fixed value. The SDU average delay curves under different BLERs in Fig. 2, have similar trend regardless of the block error rate values. At small status period timer values such as 40 ms, RLC SDU delays are large. As the status period timer values increases to a threshold, the RLC SDU delay decreases. The threshold values are 500 ms for 5% BLER, 400 ms for 10% BLER and 360 ms for 15% BLER. When the status period timer is at these values mentioned above, SDU average delays reach the lowest for the three curves, respectively. After those values, SDU delays begin to increase. The reason for such a trend of SDU delay is discussed as follows.

A small status period timer value makes the receiver send status PDUs too frequently, which results in too many RLC PDUs to be transmitted for the first time being blocked at the transmission buffer of the sender. When the status period timer value is so small that it is less than the round trip time, status period timer will expire soon after a status report is sent from the receiver. No associated RLC PDUs will be arriving within this period of time since it is shorter than the round trip time. The receiver will inadvertently be sending one or more status reports requesting retransmission of the same PDU which might already been retransmitted by the sender. If the PDU(s) on the way is/are received correctly, the requested retransmission becomes unnecessary. However, the receiver is not given enough time to verify if the last one or more sent status reports has recovered the erroneous PDU(s). Hence, more bandwidths of the wireless forward link will be used to unnecessarily transmit the PDUs requested for retransmission. Since the RLC PDUs requested for retransmission have a higher transmission priority than those to be transmitted for the first time, lots of new RLC PDUs are being blocked at the transmission buffer of the sender, which results in a higher queueing delay RLC SDUs experience at the sender. As a consequence, the SDU average delays are extremely large when the status period timer value is small. With the increase of that value, fewer status PDUs are sent to the sender, and less bandwidths of the wireless forward link is used for unnecessary retransmissions. Then the blocking of RLC PDUs to be transmitted for the first time in the transmission buffer at the sender is alleviated, and the SDU average delays decrease. When the period of sending status reports is slightly greater than the round trip time, say 300 ms or 360 ms (the round trip of RLC PDU in our simulation is a little more 240 ms), fewer unnecessary PDUs are requested for retransmission, and the timing of sending status reports is optimal. Consequently, the SDU average delays are the smallest around those points. With higher BLER, the optimal status period timer value for smallest SDU delay is closer to the round trip time. The reason is, comparing with a relatively large status period timer value, such as 400 ms, a smaller status period timer value, such as 300 ms, will decrease the re-sequencing delay that a RLC SDU experiences in the receiver because of faster recovery of transmission errors, but causes slightly larger queueing delay in the sender due to more wireless bandwidths used to transmit RLC PDUs requested for retransmission. More errors to recover implies more resequencing at the receiver and thus more blocking at the sender. These two effects will compromise each other on the total SDU delay. For a larger block error rate like 15%, the effect of decreasing the re-sequencing delay is more obvious than the other effect, so its status period timer value at which SDU average delay is the smallest is closer to the round trip time than the optimal timer value for a smaller block error rate like 5%.

On the contrary, a large status period timer value asks the receiver to infrequently send status PDUs to the sender, which results in taking more time for SDUs to be delivered to upper layers at the receiver. For the status period timer value greater than the optimal status period timer, the period of sending status PDUs is longer and thus a RLC SDU with transmission errors will have to wait for longer time to be recovered. Consequently, all of its following SDUs cannot be delivered to the upper layers until it is recovered by retransmission or replaced by a failed recovery message. This results in a larger re-sequencing delay that a RLC SDU will experience at the receiver. Hence, the SDU average delay increases as the status period timer value increases.

2) RLC Throughput: The effect of the status period timer on the RLC throughput is illustrated in Fig. 3. By the definition of the RLC throughput in the previous section, the maximum possible RLC throughput is $(S * (1 - p))$, where S is the physical layer data rate and p is the block error rate of RLC PDUs. As we know, the RLC throughput can always reach its possible maximum value so long as the transmission window of the sender is not blocked.

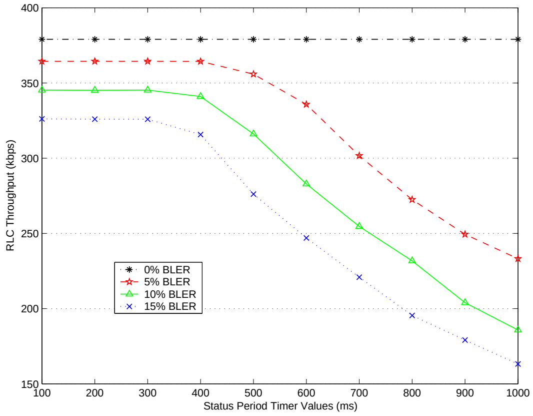

Fig. 3. Effect on RLC Throughput with Different BLERs

When no transmission error happens on RLC PDUs, the possible maximum value of the RLC throughput is equal to the wireless link data rate. As we can observe from the curve of the RLC throughput with BLER 0%, for all status period timer values, the RLC throughput is equal to 384 kbps, which is the defined physical layer data rate. A status PDU sent periodically acknowledges the sequence number whose previous PDUs have been correctly received, and when the sender receives such information, the transmission window can move forward. Then, for the small values of the status period timer, the transmission window is not blocked as a result of the higher frequent acknowledgement information included in status PDUs. When the timer value increases to the maximum tested value 1000 ms, which is equal to 50 TTIs (each Transmission Time Interval or TTI is 20 ms), the sender's transmission window can be occupied by a maximum of 1200 PDUs (during each TTI a maximum of 24 PDUs could be sent), which need to be acknowledged. The transmission window size is set to be 2047 PDUs so that it is large enough to ensure no window blocking occurs at RNC (the sender) for the maximum tested value 1000 ms when BLER is 0%. Consequently, the curve of the RLC throughput with BLER 0% keeps a straight line at 384 kbps as observed in Fig. 3.

However, for the other three curves of the RLC throughput with non-zero block error rates, there are differences. When the status period timer values are small, there is no chance of the transmission window being blocked, as described above. So the RLC throughput for each curve is much close to $(S * (1 - p))$. As the timer value increases to a threshold, which is different for different BLERs, the RLC throughput starts decreasing. The larger the block error rate is, the smaller the threshold. The reason is, for a fixed status period timer value, the transmission window with the bigger BLER is more likely to be blocked due to more RLC PDUs being retransmitted. As observed from Fig. 3, the threshold value is bigger than the round trip time when BLER is smaller than 15%. After passing the threshold, the RLC throughput starts to decrease as the status period timer value increases. The reason for decreasing RLC throughput is the sender's transmission window blocking. Blocking occurs because the transmission window is only being updated as often as the status PDUs are being received from the receiver. In the event of no status PDU being received for a long time, less protocol control information is exchanged between the sender and the receiver. Then, the sender has fewer chances to update its transmission window because of lack of protocol control information. Therefore, the transmission window at the sender will be blocked eventually due to its limited window size. At such a situation, no data are transferred to the wireless link, and the wireless link bandwidths are not fully utilised. As a consequence, the RLC throughput decreases.

3) RLC Goodput: Fig. 4 shows the effect of the status period timer on the RLC goodput. The curve of the RLC goodput is a straight line when BLER is 0%, and for a non-zero block error rate, the RLC goodput has a peak value as observed in Fig. 4.

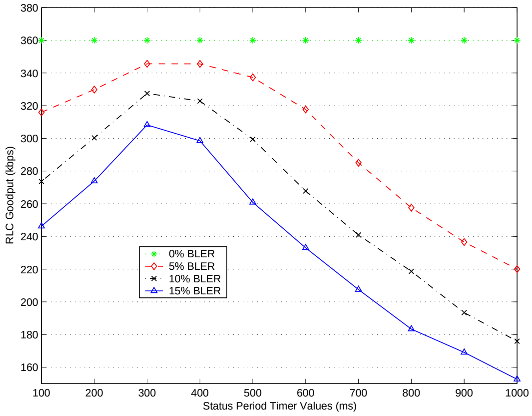

Fig. 4. Effect on RLC Goodput with Different BLERs

As we know, there is no transmission window blocking or RLC PDUs retransmitted when BLER is 0%. Therefore, all bandwidths of the wireless link are used to transmit new RLC PDUs that are all delivered to upper layers. Each RLC PDU excluding the 2-bytes PDU header is the useful information for calculations of the RLC goodput. Then the RLC goodput value with 0% BLER is $384 * 38/40 = 364.8$ kbps (excluding the two-byte header in each RLC PDU). As we observed, the straight line in Fig. 4 is the simulated RLC goodput when BLER is 0%, which perfectly matches the analysed result.

Fig. 4 shows that a peak value of the RLC goodput is reached with a non-zero BLER at a value of the status period timer that is slightly larger than the round trip time of a RLC PDU. Before this value, the smaller the status period timer value is, the smaller the RLC goodput. After this value, the larger the STATUS Period Timer value, the smaller the RLC goodput.

Even though for small values of the status period timer, there is no chance for the transmission window of the sender to be blocked and the RLC throughput can reach its possible maximum value, a significant portion of the wireless link bandwidths are used to transmit unnecessary RLC PDUs requested for retransmission. Hence, a small status period timer value may not produce any benefit to, or probably even degrade the RLC goodput because of an excess of unnecessary retransmissions of RLC PDUs. The smaller the status period timer value is, the greater the degradation of the RLC goodput. However, if the status period timer value is much larger than the round trip time, there will be more chances of the transmission window blocking due to fewer status PDUs received by the sender. This causes fewer RLC PDUs to be transmitted for the first time sent by the sender, which results in less useful information received by the receiver. Consequently, the RLC goodput degrades. The larger the status period timer value is, the higher the chance of blocking the transmission window, and the lower the RLC goodput.

B. *Effect of Poll Timer*

In order to put poll timer in effect, the poll timer trigger for polling status reports needs to be enabled in the sender. Two other triggers in addition to this one, “Last PDU in buffer” and “Last PDU in retransmission buffer,” also need to be enabled in the sender to prevent protocol deadlock [ZS02]. The timer starts when an AMD PDU containing a polling in the PDU header’s polling bit, which is triggered by other polling triggers such as the submission of “Last PDU in buffer” and “Last PDU in retransmission buffer,” is submitted to the lower layer in the sender. Since then, each time at which an AMD PDU containing a polling is submitted to the lower layer, the poll timer value is reset. The polling can be triggered by either the expiration of the poll timer, or “Last PDU in buffer” or “Last PDU in retransmission buffer” triggers in our simulation. Hence, status PDUs are polled with a least frequency of the period time with the poll timer value. They may be polled at a time less than the period, which is invoked by the triggers other than the expiration of the poll timer.

1) RLC SDU Delay: Fig. 5 shows the effects of the poll timer on the SDU average delay. When the block error rate is zero, the SDU average delay keeps a same value. For non-zero block error rates, however, at a small poll timer value such as 40 ms, the SDU average delay is very large. As the poll timer values increase until 300 ms or 360 ms, the SDU average delays decrease. When the poll timer is around such a value, the SDU average delay reaches the smallest. After that, the SDU average delay begins to increase.

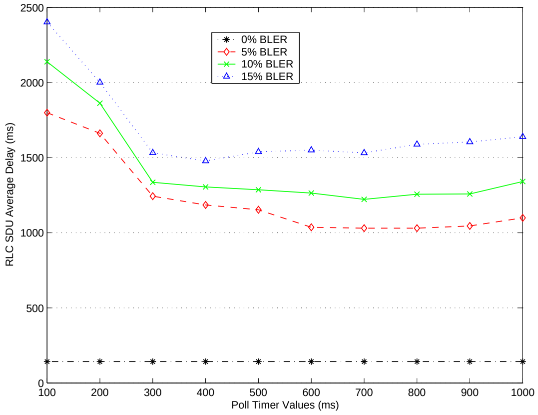

Fig. 5. Effect on SDU Delay with Different BLERs

When block error rate is zero, no transmission error happens to any RLC PDU, and no retransmissions of PDUs are

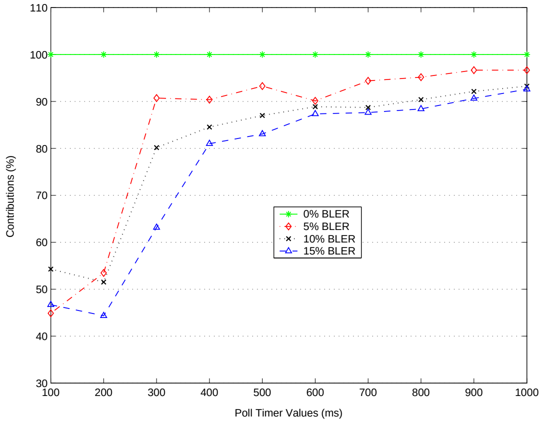

Fig. 6. Contributions of Two Triggers with Different BLERs

performed. The RLC PDUs to be transmitted in each TTI are all new PDUs generated in that TTI time, and there is one status report received in each TTI due to the trigger, “Last PDU in buffer.” Then there is no queueing delay for RLC PDUs in the transmission buffer of RNC, or re-sequencing delay at UE. All delay RLC SDUs experienced is the SDU one-way latency time and one TTI time. As shown in Fig. 5, the simulated result of the SDU average delay with 0% BLER around is 120 ms, which agrees with the expected analysis above.

Since the poll timer trigger functions very similar to the trigger status period timer when the block error rate is non-zero, the SDU average delay is very large when the poll timer values are small (because of too many unnecessary retransmissions of RLC PDUs that block new PDUs to be transmitted for the first time). Then the SDU average delay experiences the lowest value when the poll timer value is slightly larger than the round trip time, say 300 ms or 360 ms (the round trip time in our simulation is more than 240 ms). The reason for the SDU average delay being the lowest at such values is the timely transmission of status reports and fewer unnecessary retransmissions of RLC PDUs. As the poll timer values increase, the SDU average delay starts to increase for a larger block error rate such as 15%, or stay with a same value for a lower block error rate such as 10%. The reason is that a SDU may experience a larger re-sequencing delay at the receiver due to the longer duration between two times of sending status reports. As illustrated in Fig. 6, with non-zero BLERs and for a larger poll timer value, the two triggers, “Last PDU in buffer” and “Last PDU in retransmission buffer,” are involved in polling bit setups more often than for a smaller poll timer value. This shows that the importance of the poll timer decreases when its value increases to a larger one because other triggers are invoked more often.

2) RLC Throughput: The effect of the poll timer on the RLC throughput is illustrated in Fig. 7. The maximum possible RLC throughput is known to be $(S * (1-p))$. When the block error rate is zero, there is one status report received for each TTI due to the trigger “Last PDU in buffer.” As there is no chance of transmission window blocking, the RLC throughput can always reach the maximum value regardless to different poll timer values. When the block error rate is a non-zero value, the RLC throughput for different poll timer values has a trend similar to that generated by the different status period timer values. The reason is that the poll timer can be considered as a trigger of polling status PDUs periodically if it is the only enabled trigger and has already started initially in the sender. Since there are two more triggers set in the simulation to envastigate the poll timer (the status period timer is the only trigger enabled in the simulation of envastigating the status period timer), there is more chance of status PDUs being transmitted from the receiver to the sender at the same timer value for the poll timer and the status period timer. Then the decrease of the RLC throughput for the poll timer values may be slower than that for the status period timer values when they are between 360 ms and 1000 ms. This is confirmed by comparing the curves of the RLC throughput in Fig. 3 and Fig. 7 with the same timer value and block error rate.

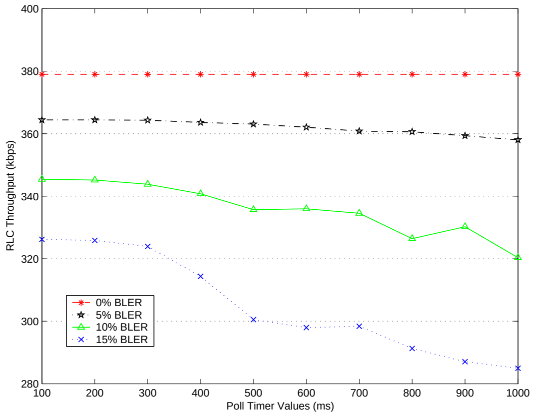

Fig. 7. Effect on RLC Throughput with Different BLERs

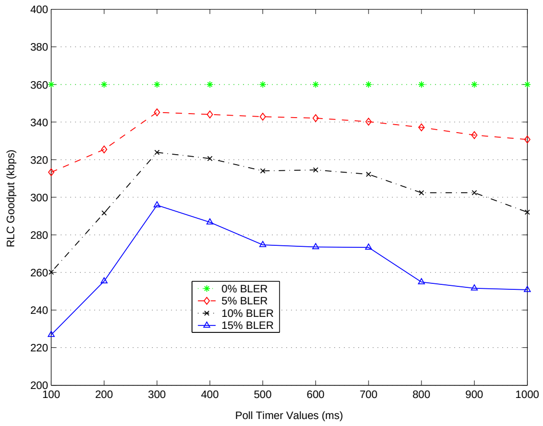

Fig. 8. Effect on RLC Goodput with Different BLERs

3) RLC Goodput: Fig. 8 illustrates the effect of the poll timer on the RLC goodput. When the block error rate is zero, the RLC goodput reaches its maximum values for all poll timer values. The reason is that no transmission window blocking or RLC PDUs’ retransmissions happen, and all bandwidths of the wireless link are used to transmitted new RLC PDUs to be delivered to the upper layer. For the non-zero block error rates, the RLC goodput is low when poll timer values are small. Then it increases to a peak value and starts to decrease with the increase of poll timer values. The poll timer value at which the RLC goodput reaches its maximum value is about 360 ms, which is slightly larger than the round trip time value. The reason is that the poll timer can cause similar effects as the status period timer once it starts. Then the sender polls status PDUs at least periodically with the period of the poll timer value. The RLC goodput can reach its maximum value with the polling period slightly larger than the round trip time.

V. Conclusions and Future Work

Our simulation results suggest that, based on a large enough transmission and reception window size, the RLC protocol can have the best performance when the value of status period timer is set to be slightly larger than the round trip time. At such a value of the status period timer, the SDU average delay can be minimum, and the RLC goodput can reach the maximum value. When the timer is smaller or larger than this value, the SDU average delay becomes larger and the RLC goodput becomes smaller because of unnecessary retransmissions of AMD PDUs at a smaller value of the status period timer, and more chances of the transmission window blocking at a larger status period timer value.

Based on a large enough transmission and reception window size, our simulation results show that the poll timer value should never also be less than the round trip time in order to reach the best RLC performance. The larger is the block error rate, the closer the value of poll timer should be to the round trip time.

This study has only evaluated two of the RLC retransmission triggers on the RLC performance considered independently of the other. There are more triggers that can invoke transmission of a status report, and two or more triggers can be set at the same time. Our future studies will focus on the optimization of two or more triggers which are simultaneously active.

Our major goal of the wireless network studies is the detailed performance evaluation of TCP over UMTS networks, especially its interactions with RLC when TCP is used as an upper layer protocol for data service.

ACKNOWLEDGEMENTS

The authors would like to thank OPNET Technologies Inc. for OPNET Modeler software support.

References

[EL02] Nicolas Enderle and Xavier Lagrange. Radio link control-acknowledged mode protocol performance modeling in UMTS. In *Proceedings of IEEE 4th International Workshop on Mobile and Wireless Communications Network*, volume 1, pages 332–336, September 2002.

[GG00] Vijay K. Garg and Mridu Garg. Role of MAC/RLC in achieving higher-data rates and QoS in third generation (3G) UMTS system. In *IEEE International Conference on Personal Wireless Communications*, volume 1, pages 459–463, Hyderabad, India, December 2000.

[Inc02] OPNET Technologies Inc. Optimised network engineering tools (OPNET) release 9.0. Technical report, 2002.

[rlc02] Radio link control (RLC) protocol specification (release 5). Specification 3GPP TS 25.322 V5.3.0, 3rd Generation Partnership Project (3GPP), December 2002. Web page: www.3gpp.org.

[XCX+02] Hua Xu, Yi-Chiun Chen, Xiao Xu, Eren Gonen, and Peijuan Liu. Performance analysis on the radio link control protocol of UMTS system. In *IEEE 56th Vehicular Technology Conference*, volume 4, pages 2026–2030, September 2002.

[XXC+02] Xiao Xu, Hua Xu, Yi-Chiun Chen, Eren Gonen, and Peijuan Liu. Simulation analysis of RLC timers in UMTS systems. In *Proceedings of the 2002 winter simulation conference*, volume 1, pages 506–512, December 2002.

[YPM00] Uooyeol Yoon, Seongsoo Park, and Pall S. Min. Performance analysis of multiple rejects ARQ at RLC(radio link control) for packet data service in W-CDMA system. In *Proceedings of IEEE GLOBECOMM '00*, volume 1(27), pages 48–52, San Francisco, CA, December 2000.

[ZS02] Qinqing Zhang and Hsuan-Jung Su. Performance of UMTS radio link control. In *Proceedings of IEEE International conference on Communications*, volume 5, pages 3346–3350, 2002.