A Simulation Implementation of the LTE-Uu Interface Datalink Layer in OMNeT++

Mohammad Arouri[1], Ziyad Atiyyeh[1], Anas Mousa[1],
Amna Eleyan[1], and Hussein Badr[2]

[1] Computer Systems Engineering Department
Birzeit University, Birzeit, Palestine
{mohammadarouri,ziyad.a.2010,anas.n.mousa}@gmail.com,
aeleyan@birzeit.edu
[2] Department of Computer Science
University at Stony Brook, Stony Brook, New York, U.S.A.
badr@cs.stonybrook.edu

Abstract. The 3rd Generation Partnership Project (3GPP)'s Long Term Evolution (LTE) standards define the next major step in the evolution of cellular systems towards higher data rates, low latency, and greater spectral efficiency. This occurs in the context of a System Architecture Evolution (SAE) that specifies a packet-switched IP architecture for both voice and data transmission. We present a simulation implementation of a key component of the overall LTE-SAE: the L2 (Datalink) layer of the LTE-Uu interface between mobile User Equipment (UE) and a base station eNodeB. The simulation is developed for the INET Framework of OMNeT++4.0, an open-source computer-network simulation environment, and implements the Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), and Medium Access Control (MAC) layers of the LTE-Uu. The implementation is extensible, and is intended to serve as a publicly-available, open-source platform for further simulation development of various aspects of LTE-SAE.

Keywords: LTE (Long Term Evolution), SAE (System Architecture Evolution), PDCP (Packet Data Convergence Protocol) Layer, RLC (Radio Link Control) Layer, MAC (Medium Access Control) Layer, Simulation, OMNeT++, INET Framework.

1 Introduction

The Long Term Evolution (LTE) standards of the 3rd Generation Partnership Project represent a major development in the evolution of UMTS (Universal Mobile Telecommunication System) beyond 3G (3rd Generation) mobile cellular technology, and aims at providing high data rates, lower latencies, and greater spectral efficiencies. LTE development goes hand-in-hand with SAE (System Architecture Evolution) which defines an AIPN (All IP Network) core network architecture, the EPC (Evolved Packet Core). The major elements of the combined LTE-SAE system, collectively referred to as the EPS (Evolved Packet System), are shown is Fig. 1. The reader is referred to [1] for further details.

K. Pentikousis et al. (Eds.): MONAMI 2010, LNICST 68, pp. 270–284, 2011.

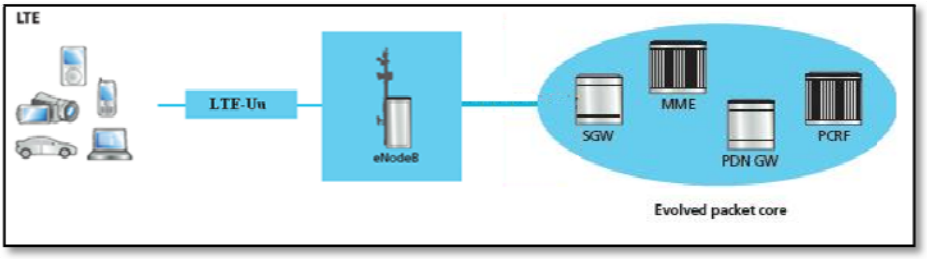


Fig. 1. Elements of the LTE-SAE architecture. The node *PDN GW* is the gateway to a Public Data Network.

In this paper, we present the development of a simulation model for a key component of the LTE-SAE architecture: the L2 (Datalink) layer of the LTE-Uu wireless interface between mobile User Equipment (UE) and a base station eNodeB. The model is implemented as a component that extends the INET Framework [2] of the open-source, extensible, discrete-event network simulator OMNeT++4.0 [3]. Our model design and implementation are themselves also extensible, and are intended to serve as a publicly-available, open-source platform for further simulation development of various aspects of LTE-SAE.

So far as we can tell from the published literature, the only previous work that is similar to ours in breadth and scope is that of Qiu *et al.* [4], in which the authors report on the implementation of an LTE-SAE simulation model using the well-known, open-source network simulator ns-2 [5]. Their work, however, differs from ours in several important respects. Our work focuses on a highly-detailed implementation of the L2 (Datalink) layer protocols (see §2.1 below), strictly in accordance with the relevant defining specifications. The work in [4] models these protocols' dynamics in a more abstract manner using queues. On the other hand, [4] uses a richer and more varied set of traffic classes to report their results than we do. It is probably fair to say that the primary aim of [4] is to provide – quoting the authors – an "accurate enough" simulation model for the study of various "optimization features" for the performance improvement of the LTE-SAE network. Our primary aim, on the other hand, is to provide the research community with an extensible development platform in the form of a specification-compliant simulation implementation of the protocols at a fine granularity of detail.

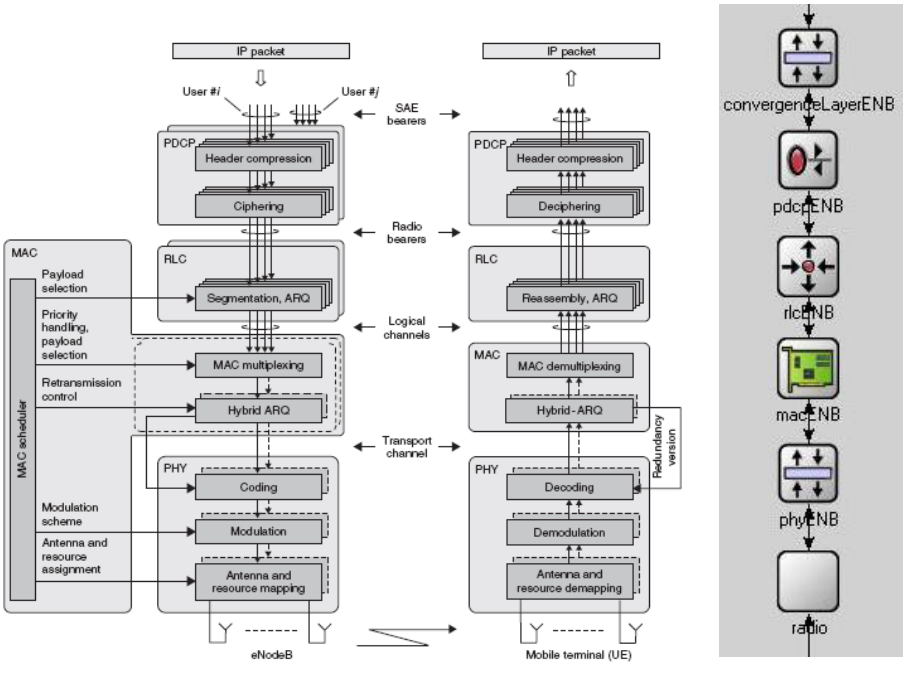
The rest of the paper is structured are follows. Section 2 describes the model design and implementation. Section 3 presents some results from the model. In Section 4 we close with some brief concluding remarks.

2 The Simulation Model

2.1 LTE-Uu Interface

The (user-plane) protocol stack of the LTE-Uu interface is composed of the L2 (Datalink) and L1 (Physical) layers as shown in Fig. 2(a). Fig. 2(b) shows our OMNeT++ implementation. From Fig. 2(a), it can be seen that the L2 layer is composed of three sublayers [1, 6, 7]. From top down:

- **Packet Data Convergence Protocol (PDCP)** [1, 6, 8]. Performs IP header compression/decompression to reduce the number of bits transmitted over the radio link; and ciphering/deciphering where required.



(a) (b)

Fig. 2. (a) The L2/L1 LTE protocol architecture, shown here operating on the Downlink Shared Channel from *eNodeB* to *UE* (reproduced from [6]). Note the *PDCP*, *RLC* and *MAC* sublayers of the L2 layer. *PHY* is the L1 (Physical) layer. (b) INET Framework simulation implementation of the protocol architecture, shown here for the *eNodeB*. The architecture in the *UE* is structurally the same.

- **Radio Link Control (RLC)** [1, 6, 9]. On the outgoing side, it performs segmentation and concatenation of the incoming PDCP protocol data units (PDUs – called RLC Service Data Units (SDUs) from the perspective of the RLC) to produce dynamically-resized, rate-adapted RLC Protocol Data Units (PDUs) for the MAC sublayer, in line with decisions made by the latter's scheduler. On the incoming side, it performs reassembly of incoming MAC PDUs (RLC SDUs).
  The RLC has two modes of operation, Unacknowledged Mode (UM) and Acknowledged Mode (AM). In AM, the RLC implements an ARQ mechanism to guarantee reliable, in-order data transmission on the Downlink Shared Channel (DL-SCH) from eNodeB to UE.
- **Medium Access Control (MAC)** [1, 6, 10]. The MAC sublayer in the eNodeB performs scheduling on both the Downlink (eNodeB to UE) and Uplink (UE to eNodeB) Shared Channels (DL-SCH and UL-SCH, respectively). These are the

main channels for data transmission. The MAC sublayers in the UE and the eNodeB produce dynamically-sized Transport Blocks (TBs) for transmission on these channels, with one TB, if available, being transmitted during a Transmission Time Interval (TTI; typically 1 msec.). In the case of Multiple-Input Multiple-Output (MIMO) antenna spatial multiplexing, more than one TB may be transmitted per TTI, but this is not currently implemented in our simulation. The MAC sublayer may also perform multiplexing/demultiplexing between multiple logical channels (data, control, *etc*.) for transmission on the DL-SCH and UL-SCH, but this feature is also not currently implemented in our simulation.

The MAC sublayer implements a hybrid-ARQ (HARQ) mechanism on the DL-SCH and UL-SCH, in the form of up to eight (the exact number is a parameter of our simulation model) parallel Stop-and-Wait ARQ processes with ACK/NACK signalling in the reverse direction.

Fig. 2(b) shows the protocol architecture implemented in our simulation model, which parallels that of Fig. 2(a). The figure shows the implementation in the eNodeB; the implementation in the UE is structurally exactly the same. At the top of the protocol stack, we introduced a 'convergence sublayer' whose purpose is to 'stitch' the pre-existing INET Framework structure to the LTE sublayers. Its sole function is to implement conversion between pre-existing OMNNeT++ message object types that are passed to and from the next layer up (not shown in the figure; typically, in the UE this would be the Network Layer, and in the eNodeB it would be an INET Framework *relayUnit* module) and those we implement for PDCP SDUs. At the bottom of our stack we have a physical and a radio sublayer. These will be discussed in §2.3 below.

2.2 Connection Establishment

Before data transfer can occur, connection establishment between UE and eNodeB has to take place [1, 6]. In the simulation model, connection establishment consists of a sequence of four phases, as shown in Fig. 3. Some of the activity of these phases pertains to the Radio Resource Control (RRC) layer [1, 6, 7, 11] which lies in the control-plane protocol stack, immediately above the PDCP sublayers shown in Fig. 2, but is not otherwise explicitly represented in our model.

1. **Cell search** [1, 6, 7]. The UE acquires time and frequency synchronization with a cell and detects the Cell ID, based on the Primary and Secondary Synchronization Signals (PSS and SSS, respectively) transmitted in the downlink by the eNodeB.
2. **Cell acquisition and system information** [1, 6, 7, 11]. The UE acquires cell system information by means of a Master Information Block (MIB) and System Information Blocks (SIBs) of multiple types, transmitted with regular periodicities by the eNodeB. In our implementation, the number of SIB types that a UE must receive during connection establishment is sampled from a uniform, integer distribution that is defined as a model parameter.
3. **Random access procedure** [1, 6, 7, 10]. The UE requests and achieves connection setup. The simulation model implements the contention-based form of the random access procedure because we wanted to simulate a UE initiating connection establishment as it comes to the E-UTRAN (Evolved Universal Terrestrial Radio

Access Network; *i.e.*, the LTE network) from the outside. In the contention-free form of the procedure, it is the E-UTRAN that initiates connection establishment.

4. **Initial security activation and radio bearer establishment** [1, 6, 11]. Activates integrity protection and ciphering, and establishes the radio bearer(s). Our model simplifies this somewhat by implementing only the first and last message exchange in what is actually a two-step process. Each step has two handshakes and the messages exchanged in the first step may be interleaved with those of the second.

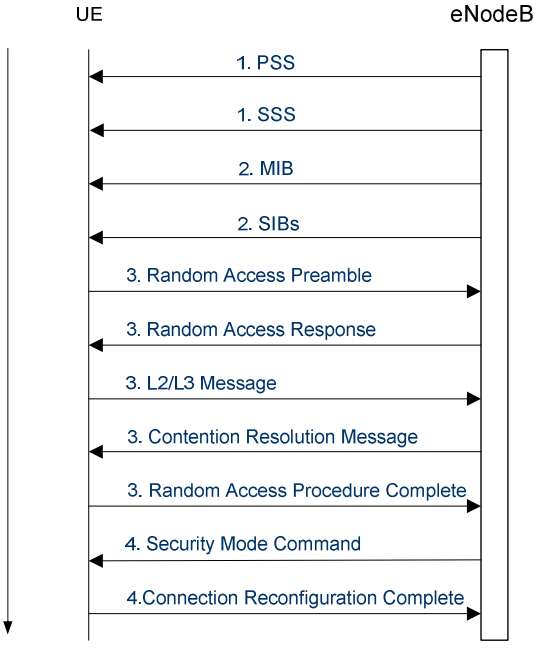


Fig. 3. The four phases of connection establishment

2.3 Some Salient Details of the Model Implementation

As noted in §2.1, our INET Framework implementation for the LTE-Uu protocol stack includes a physical sublayer and a radio sublayer. It is important to note, however, that these sublayers are, at present, basically just stubs. Our implementation models the LTE L1 (Physical) layer in a highly abstract, reductive manner that is nevertheless flexible. The wireless channel is modelled by three distributions which define, respectively: the channel transmission rate; the Block Error Rate (BER) for TBs; and the signalling error rate in the reverse direction for the ACKs/NACKs of the MAC sublayer HARQ. All three distributions are implemented as model parameters and may thus easily be changed from run to run; each may also be dynamically resampled during a simulation run. Our radio sublayer is based on the *AbstractRadio* module that exists in the INET Framework (where it is used to implement the IEEE 802.11 radio link). We have stripped this down so that it just simply transmits available TBs over the air link at 1-msec TTIs, with appropriate propagation delays

that are a function of the relative positions of UE and eNodeB. The fate of these transmitted TBs is determined in our physical sublayer module on the receiving side, in accordance with the values sampled from the BER distribution. Segmentation and concatenation in the RLC sublayer, on the other hand, is determined by the values sampled from the distribution that defines the current channel transmission rate. Future, substantive OMNeT++ simulation models for the LTE L1 layer may, of course, easily be incorporated into our framework. We note that such simulation models already exist, but not for OMNeT++: [12], for example, written in MATLAB and publicly available, is one such. We have not investigated the possibility of combining such simulators with the OMNeT++ framework; though possible in principle, this would probably be a far from trivial task.

Implementation of the HARQ mechanism[1] in the MAC sublayer, and of the interaction between HARQ and the ARQ of the RLC sublayer in AM, is based on the simulation model developed by Chuan & Lin [13]. Appropriate modification and extension to this model were made to take account of the different framework structure for our implementation; of the enhanced functionality we introduced in the form of RLC segmentation and concatenation; and of RLC UM operations.

The following were implemented as parameters of the model, allowing flexible configuration for simulation runs (default values for these parameters are given in parentheses):

- number of parallel HARQ processes (8);
- maximum number of retransmissions per HARQ process (5);
- RLC operates in UM or in AM (AM);
- RLC *t-PollRetransmit* timer [9] (5 msec.);
- RLC *t-Reordering* timer [9] (5 msec.);
- RLC *t-StatusProhibit* timer [9] (5 msec.);
- RLC reception buffer (512 bytes).

Further details on this and other aspects of the simulation model design and implementation are made available in [14].

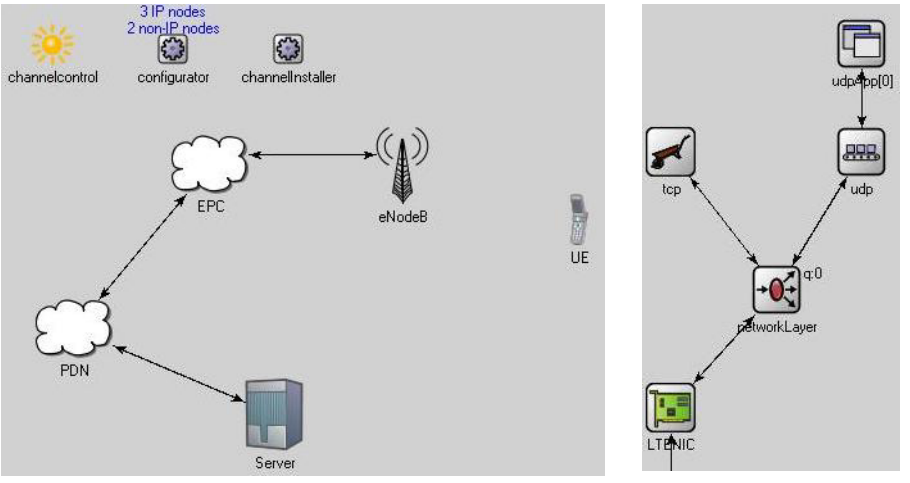
3 Results

To demonstrate the model's capabilities, we present a sample of performance results on various aspects of the RLC sublayer's UM and AM operating modes, and the MAC sublayer's HARQ mechanism. We note that complementary results on related aspects of the MAC HARQ and/or RLC AM have been published elsewhere – *e.g.*, [13, 15], amongst others.

We simulated the network shown in Fig. 4(a), in which a UE downloads a UDP stream from the node *Server*. Node *EPC* represents the Evolved Packet Core, and *PDN* the Public Data Network (in actuality, these are INET Framework switch and router nodes, respectively). Delays on the links *eNodeB* ↔ *EPC*, *EPC* ↔ *PDN* and

[1] The highly simplified, abstract model for the wireless link in our implementation makes the distinction between synchronous/asynchronous HARQ schemes less significant than would be the case in a real system. It also renders the difference between adaptive/non-adaptive HARQ schemes essentially meaningless. The HARQ mechanism we implemented comes closest in flavour to an asynchronous, 'adaptive' scheme.

PDN ↔ *Server* can be set to capture tunnelling latency through the EPC between *eNodeB* and the PDN GW, and latency in the PDN itself. Fig. 4(b) shows the internal structure of node *UE*, which is composed of the typical TCP/IP protocol stack, including a UDP video-streaming client application-layer process. *LTENIC* on the lower left of Fig 4(b) is a compound module, composed of the L2/L1 protocol stack shown in Fig. 2(b). Fig. 2(b) actually shows the equivalent compound module located in *eNodeB*, but the one in *UE* is exactly the same.



(a) (b)

Fig. 4. (a) The network simulated in INET Framework. (b) Internal structure of the node *UE* of Fig. 4(a). The compound module *LTENIC* on the lower left is composed of the L2/L1 protocol stack, which is structurally the same as that shown for *eNodeB* in Fig. 2(b).

For the results reported, the UDP download was a constant bit rate (CBR) stream composed of 10,487 UDP datagrams with a payload of 1,000 bytes each, for a total of 10MB of data. The application process at node *Server* emitted the traffic at a rate of one datagram every 0.1 msec. Note that this implies that the stream is injected into the network at a rate ≥ 80Mbps. The bandwidth of the links on the path between nodes *Server* and *eNodeB* was set to 100Mbps.

In the LTE configuration, and in order to enable straightforward comparison between results for the RLC sublayer's UM and AM[2] mechanisms, RLC segmentation in *eNodeB* was disabled so that each IP packet (PDCP SDU) corresponded to exactly one MAC sublayer TB. Also, the RLC *t-StatusProhibit* timer was disabled in *eNodeB*. The *eNodeB* → *UE* downlink channel transmission rate was fixed such that a 1-msec TTI could accommodate exactly one TB (implying a transmission rate of a little over 8 Mbps). The signalling error rate on the *UE* → *eNodeB* uplink channel was set to 2%. Finally, we ensured that sufficient buffer space was provided in the PDCP

[2] UDP traffic would, of course, normally be handled with the RLC operating in UM not AM, but this does not effect the substance of the results presented.

sublayer at *eNodeB*, with no discard timer in effect [1], so that none of the arriving packet stream is lost prior to transmission over the downlink channel.

Tables 1 and 2 below present performance results for some aspects of the RLC sublayer UM and AM mechanisms, respectively (based on averages from five independent runs of the simulation):

- Columns (a) of the tables give the Block Error Rate (BER), *i.e.*, the loss rate for TBs on the *eNodeB* → *UE* downlink channel.
- Columns (b) give the number of successful IP packets delivered by the RLC sublayer to the PDCP and higher layers at the receiving node *UE*. This is out of a total of 10,487 packets received off the network by *eNodeB* for onward transmission to node *UE*.
- Columns (c) give the total time to download the 10MB stream, in msec.

Table 1. Performance of the RLC sublayer UM mechanism in node *UE*
(with *t-Reordering* timer = 15 msec, *t-PollRetransmit* timer = 15 msec ;
and 8 parallel MAC sublayer HARQ processes, each with maximum retransmission limit = 3)

(a)	(b)	(c)	(d)	(e)	(f)
BER	Number of PDUs (IP packets) successfully received by upper layers in node *UE* (out of 10487 sent)	Total download time (msec)	Number of PDUs dropped from RLC reception buffer	Number of PDUs needing HARQ retransmission (out of 10487)	Number of TBs suffering residual HARQ errors
5%	10461	11404	25	539	1
10%	10385	12008	91	1061	11
25%	9890	14145	430	2645	167
50%	8232	18711	935	5270	1320

- Columns (d) differ between the two tables. For UM in Table 1, it gives the number of PDUs dropped from the RLC sublayer's reception buffer at node *UE*. These are PDUs that were delayed due to retransmission by the MAC sublayer's HARQ mechanism, and subsequently rejected by the reception buffer due to the way the lower boundary of this "sliding window" buffer is updated and the effects of the *t-Reordering* timer [9].
  There was no occurrence of reception buffer drops in AM. This was presumably due to the fact that out-of-order PDUs arriving at the RLC sublayer of *UE* caused its AM ARQ mechanism to trigger off requests for retransmission of the missing PDUs, and these happened to arrive in time before the lower end of the reception buffer was updated past them. So instead, we chose to report on the number of PDUs retransmitted by the AM ARQ mechanism in column (d) of Table 2.
- Columns (e) give the number of IP packets that required HARQ retransmission at the MAC sublayer. (Recall that in our case, each packet forms a single RLC SDU, which in turn forms a single MAC sublayer TB.)

- Columns (f) give the number of packets from columns (e) that experienced residual errors even after the maximum number of 3 retransmissions by the MAC sublayer HARQ mechanism, and hence could not be salvaged.

Table 2. Performance of the RLC sublayer AM mechanism in node *UE*
(with *t-Reordering* timer = 15 msec, *t-PollRetransmit* timer = 15 msec ;
and 8 parallel MAC sublayer HARQ processes, each with maximum retransmission limit = 3)

(a)	(b)	(c)	(d)	(e)	(f)
BER	Number of PDUs (IP packets) successfully received by upper layers in node *UE* (out of 10487 sent)	Total download time (msec)	Number of PDUs retransmitted by RLC AM ARQ	Number of PDUs needing HARQ retransmission (out of 10487 + column (d))	Number of TBs suffering residual HARQ errors
5%	10487	11719	299	555	1
10%	10487	12751	1026	1158	12
25%	10487	19929	4309	3736	228
50%	10487	40044	11277	11363	2829

Figs. 5 and 6 below plot some of the values in Tables 1 and 2 in order to highlight and compare some of the performance differences between UM and AM. Fig. 5 plots the data in columns (b) and Fig. 6 the data in columns (c). For both figures we calculated 99.9% level confidence intervals (in order to make ample allowance for the Bonferroni Inequality). All the confidence intervals turned out to be too small to show in the scale of the figures, except for the AM curve in Fig. 6.

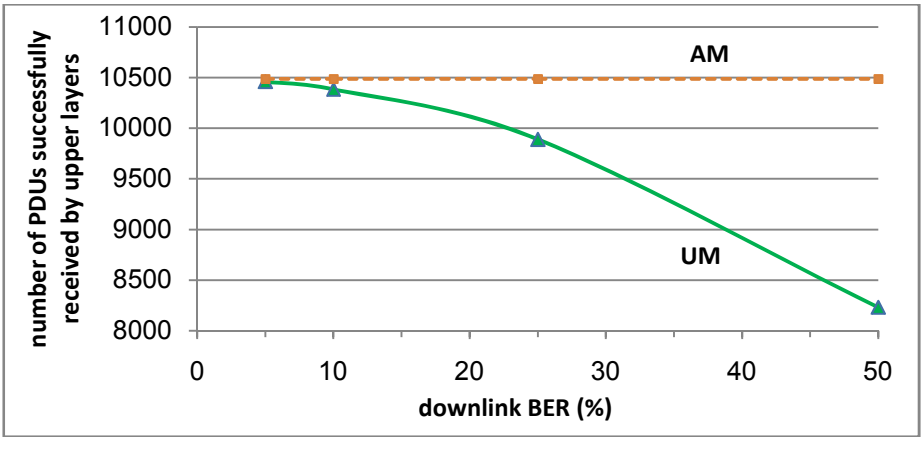


Fig. 5. Number of IP packets successfully received by higher layers at the receiving node *UE*, out of the 10,487 in the 10MB data stream (columns (b) of Tables 1 & 2)

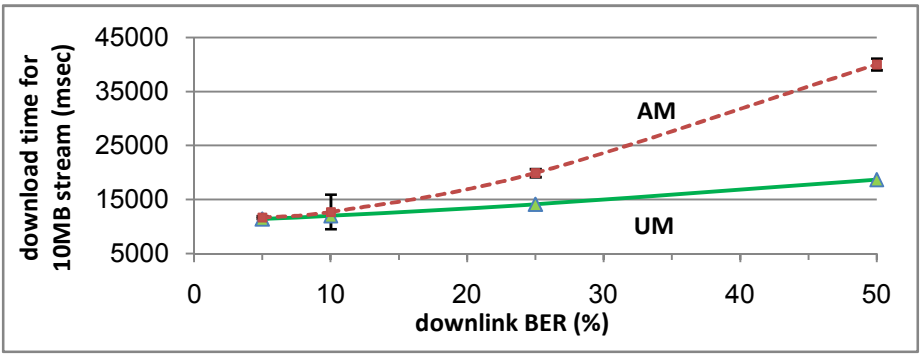


Fig. 6. Total time to download the 10MB data stream (columns (c) of Tables 1 & 2)

Fig. 5 serves to highlight how UM performance degrades in the presence of increasing BER. But it also indirectly demonstrates the effectiveness of the MAC sublayer HARQ mechanism. For example, at a somewhat extreme BER of 50%, an average of 8,232 packets out of 10,487 in the data stream (*i.e.*, 78.5%) arrive successfully. One would expect that, on average, only 50% should do so. The difference is accounted for by the MAC HARQ, which salvages only just short of 25% of the number of packets in the stream.

Fig. 5 also shows that the RLC's AM ARQ is effective in ensuring that all packets are successfully received. But as can be seen from Fig. 6, this comes at a potentially significant cost in download time as the BER increases. Note that with a 1-msec TTI during which one TB constituting one packet may be transmitted from *eNodeB* to *UE*, the theoretical lower bound for the data stream download time is of the order of 10,500 msec. Our simulations can probably do with tighter calibration of the protocol processing delays in the L2 layer stack, so it is probably advisable to approach absolute delay results from simulations with a little caution. But we can use these values on a relative, comparative basis with more confidence. From Fig. 6 we can see that the download time in UM increases by two-thirds (from an average of 11,404 msec. to 18,711 msec.) as the BER increases from 5% to 50%, due to increasing MAC HARQ recovery activity. Now, comparing UM with AM, it can be seen that the RLC AM ARQ mechanism imposes a further, ever-increasing overhead, causing the download time to more than double at a BER of 50% (18,711 msec for UM *vs*. 40044 msec for AM). One further point worth noting in this context is that, returning to Table 2 and comparing column (d) with column (f), it is clear that the amount of retransmission by the RLC AM ARQ mechanism is overwhelmingly disproportionate to the number of packets that the MAC HARQ was unable to salvage.

We now turn our attention to the effects of the RLC sublayer's *t-Reordering* and *t-PollRetranmsit timers*. Results (based on single runs of the simulation) are presented in Table 3 below:

- Column (a) gives various values for the *t-Reordering* timer, in msec.
- Column (c) gives the number of PDUs dropped by the RLC sublayer's reception buffer at node *UE* (*cf.* Table 1, column (d)).

- Column (d) gives the number of PDUs retransmitted by the AM ARQ mechanism (*cf.* Table 2, column(d)). This column does not apply to RLC UM.
- Column (e) give the total time to download the 10MB stream, in msec (*cf.* Tables 1 and 2, columns(c)).
- The *t-PollRetransmit* timer is only used in AM. It is used by the sender's RLC ARQ mechanism to control the solicitation of status reports from the receiver.
  Each table entry for AM has two values for a given setting of *t-Reordering*: one for *t-PollRetransmit* = 10 msec and the other for *t-PollRetransmit* = 20 msec. Thus, each AM entry in columns (c) - (e) is split into two subcells, one for each of *t-PollRetransmit* = 10 and 20 msec, respectively.
  As an example of how to read the table, consider when the *t-Reordering* timer is set at 5 msec (column (a)). In UM, the download time (column (e)) is 12,046 msec. In AM, on the other hand, and with *t-PollRetransmit* = 10 msec (column (e), left subcell), the download time is 17,549 msec; with *t-PollRetransmit* = 20 msec (column (e), right subcell), it is 17,847 msec.

As before, it is probably easier to absorb the data in the table through plots, which we present in Figs. 7, 8 and 9 below.

Table 3. Effect of RLC sublayer *t-Reordering* and *t-PollRetransmit* timers (with 8 parallel MAC sublayer HARQ processes, each with maximum retransmission limit = 3 ; and BER = 10%)

(a)	(b)	(c)		(d)		(e)	
t--Reordering **(msec)**		**Number of PDUs dropped from RLC reception buffer**		**Number of PDUs retransmitted by RLC AM ARQ**		**Total download time (msec)**	
	UM	**1065**		**Not applicable**		**12046**	
5	***t-PollRetransmit*** **(msec)**	**10**	**20**	**10**	**20**	**10**	**20**
	AM	**0**	**0**	**5280**	**5007**	**17549**	**17847**
	UM	**100**		**Not applicable**		**12046**	
10	***t-PollRetransmit*** **(msec)**	**10**	**20**	**10**	**20**	**10**	**20**
	AM	**0**	**0**	**1058**	**944**	**13209**	**13086**
	UM	**0**		**Not applicable**		**12046**	
20	***t-Poll-Retransmit*** **(msec)**	**10**	**20**	**10**	**20**	**10**	**20**
	AM	**0**	**0**	**57**	**69**	**12118**	**12132**

Fig. 7 below highlights the negative impact that too small a value for *t-Reordering* has on the number of PDUs dropped from the RLC sublayer's reception buffer in UM. Each PDU not dropped from the buffer translates, of course, to one more PDU successfully delivered to the upper layers at the receiving node. Note from Fig. 9 below that increasing the value of *t-Reordering* in UM has no effect on the download time of the data stream. In AM, no PDUs are dropped from the buffer, as was has already been discussed in the context of the Tables 1 & 2 results.

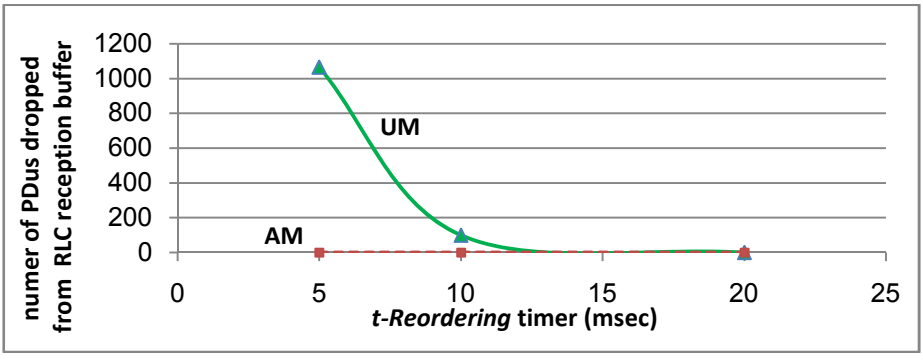


Fig. 7. Number of PDUs dropped from the RLC reception buffer (column (c) of Table 3)

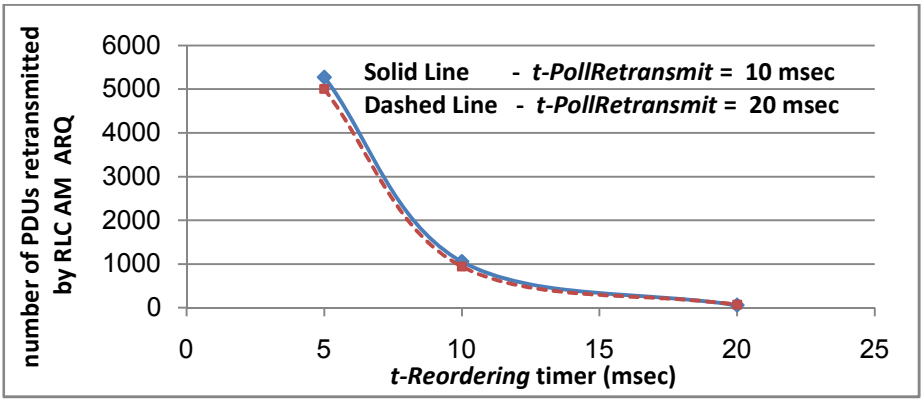


Fig. 8. Number of PDUs retransmitted by the RLC sublayer AM ARQ (column (d) of Table 3)

Fig. 8 above shows the number of PDUs retransmitted by the RLC ARQ mechanism in AM. While the *t-PollRetransmit* timer values we used had no significant effect on performance, the figure again highlights – as does Fig. 7 for UM – the critical need to configure adequate values for *t-Reordering* in order to achieve protocol efficiency. We have already previously noted that an overwhelming proportion of the AM ARQ retransmissions seem to be unnecessary, and this is

further confirmed by the decrease in the download time for AM, as seen in Fig. 9, that goes hand-in-hand with the Fig.8 decrease in the number of ARQ-retransmitted PDUs (AM will always, of course, successfully deliver the entire 10,487 packets of the data stream in whatever time – be it short or be it long – the download takes).

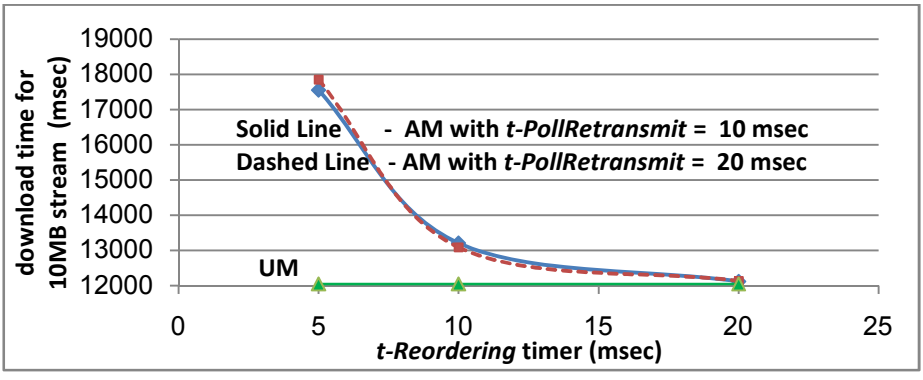


Fig. 9. Total time to download the 10MB data stream (column (e) of Table 3)

Finally, we present some results on the performance of the MAC sublayer's HARQ mechanism in Tables 4 and 5 below (based on averages from three independent runs of the simulation). Table 4 takes a look at the effect of the maximum number of HARQ retransmissions per HARQ process for a fixed number of 8 parallel such processes and a BER of 10%. Table 5, on the other hand, fixes that maximum number of retransmissions at 3, but varies the number of parallel HARQ processes. Constraints of space do not permit us to do more than simply offer these results for the reader's consideration without further ado.

Table 4. Effect of the MAC sublayer maximum retransmission limit for the HARQ processes (with 8 parallel HARQ processes ; BER = 10% ; and RLC operating in UM with *t-Reordering* timer = 15 msec)

(a)	(b)	(c)	(d)
Maximum retransmission limit per HARQ process	Number of TBs needing HARQ retransmission (out of 10487 sent)	Total number of TB transmissions by the HARQ processes (also counting multiple retransmissions for the same TB)	Number of TBs suffering residual HARQ errors
3	1070	1185	8
5	1070	1193	0
7	1070	1193	0
10	1070	1193	0

Table 5. Effect of the number of MAC sublayer parallel HARQ processes (with maximum retransmission limit = 3 per process ; BER = 10% ; and RLC operating in UM with *t-Reordering* timer = 15 msec)

(a)	(b)	(c)	(d)
Number of parallel HARQ processes	**Number of TBs needing HARQ retransmission (out of 10487 sent)**	**Total number of TB transmissions by the HARQ processes (also counting multiple retransmissions for the same TB)**	**Number of TBs suffering residual HARQ errors**
1	**1070**	**1186**	**11**
3	**1056**	**1183**	**13**
8	**1073**	**1188**	**9**

4 Conclusion

In this paper we have described the development of a 3GPP-specification-compliant simulation model, with a fine granularity of detail, for the operations of the L2 layer protocols of the LTE-Uu interface. The model is implemented in the INET Framework of OMNeT++4.0. Performance results from the model were presented. The model is intended to provide the research community with an extensible, open source simulation development platform for research in LTE-SAE.

Acknowledgments

We gratefully acknowledge the help extended by Ching-Hsiang Chuan and Professor Phone Lin of the Department of Computer Science & Information Engineering, National Taiwan University, in making available to us the details of their simulation model for the HARQ-ARQ Interaction for LTE [13].

References

1. Sesia, S., Toufik, I., Baker, M. (eds.): LTE – The UMTS Long Term Evolution: From Theory to Practice. John Wiley & Sons, Chichester (2009)
2. OMNeT++, http://www.omnetpp.org
3. INET Framework for OMNeT++4.0, http://inet.omnetpp.org
4. Qiu, Q., Chen, J., Ping, L., Zhang, Q., Pan, X.: LTE/SAE Model and its Implementation in NS 2. In: 2009 Fifth International Conference on Mobile Ad-Hoc and Sensor Networks, pp. 299–303. IEEE, Los Alamitos (2009), doi:10.1109/MSN.2009.58
5. Ns-2 Wiki, http://nsnam.isi.edu/nsnam
6. Dahlman, E., Parkvall, S., Sköld, J., Beming, P.: 3G Evolution – HSPA and LTE for Mobile Broadband. Academic Press, Elsevier, Oxford (2008)
7. 3rd Generation Partnership Project, 3GPP TS 36.300: E-UTRA and E-UTRAN; Overall Description; Stage 2, Release 9 (2010), http://www.3gpp.org/ftp/Specs/html-info/36300.htm

8. 3rd Generation Partnership Project, 3GPP TS 36.323: E-UTRA; PDCP specification, Release 9 (2010), http://www.3gpp.org/ftp/Specs/html-info/36323.htm
9. 3rd Generation Partnership Project, 3GPP TS 36.322: E-UTRA; RLC protocol specification, Release 9 (2010), http://www.3gpp.org/ftp/Specs/html-info/36322.htm
10. 3rd Generation Partnership Project, 3GPP TS 36.321: E-UTRA; MAC protocol specification, Release 9 (2010), http://www.3gpp.org/ftp/Specs/html-info/36321.htm
11. 3rd Generation Partnership Project, 3GPP TS 36.331: E-UTRA; RRC; Protocol specification, Release 9 (2010), http://www.3gpp.org/ftp/Specs/html-info/36331.htm
12. Mehlführer, C., Wrulich, M., Ikuno, J.C., Bosanska, D., Rupp, M.: Simulating the Long Term Evolution Physical Layer. In: 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, pp. 1471–1478 (2009), http://www.eurasip.org/Proceedings/Eusipco/Eusipco2009/contents/papers/1569184698.pdf
13. Chuang, C.-H., Lin, P.: Performance Study for HARQ-ARQ Interaction of LTE. J. Wirel. Commun. Mob. Comput. (2009), doi:10.1002/wcm.834
14. Public release of our code and documentation: url will be made available in time for the conference. In the meantime, this material is available on demand from the authors
15. Ikuno, J.C., Wrulich, M., Rupp, M.: Performance and Modeling of LTE H-ARQ. In: Proceedings of WSA 2009, Berlin, pp. 130–135 (2009), http://www.eurasip.org/Proceedings/Ext/WSA2009/WSA2009proceedings.pdf