Modeling LTE Protocol for Mobile Terminals using a Formal Description Technique

Anas Showk[1], David Szczesny[1], Shadi Traboulsi[1], Irv Badr[2], Elizabeth Gonzalez[1], and Attila Bilgic[1]

[1] Institute for Integrated Systems, University of Bochum, 44801 Bochum, Germany
{anas.showk, david.szczesny, shadi.traboulsi, elizabeth.gonzalez, attila.bilgic}@is.rub.de
[2] IBM Rational, Chicago, IL, USA
ibadr@us.ibm.com

Abstract. The Long Term Evolution (LTE) radio communication is the upgrade of the current 3G mobile technology with a more complex protocol in order to enable very high data rates. The usage of Model Driven Development (MDD) has arisen as a promising way of dealing with the increasing complexity of next generation mobile protocols. In this paper, a light version of the LTE protocol for the access stratum user plane is modeled using the SDL Suite™ tool. The tool shows easy understanding of the model as well as easy testing of its functionality using simulation in cooperation with Message Sequence Chart (MSC). The simulation result shows that the implemented Specification and Description Language (SDL) guarantees a good consistency with the target scenarios. The system implementation is mapped to multiple threads and integrated with an operating system to enable execution in multi core hardware platforms.

Key words: Service-Oriented applications, formal modeling, automatic code generation, formal verification, formal validation

1 Introduction

Developing next generation mobile communication and wireless technologies greatly benefit from reusing prevailing approaches and best practices. For over a decade, most global installations have taken advantage of Model Driven Development (MDD) for communication products; oftentimes through tools using the Specification Description Language (SDL)[1]. This especially applies to most protocol based products developed within the last twenty years

Besides other domains, a big majority of wireless base stations and personal handset devices world-wide currently use SDL with Testing and Test Control Notation version 3 (TTCN-3) [2]. Consequently, SDL generated, and TTCN tested systems now drive over eighty percent of all wireless technology in the world [3]. As the merits of domain specific modeling are just now becoming apparent,

one has to commend the foresight used in development of the SDL language, over two decades ago. SDL, coupled with TTCN, has provided a solid workflow for model driven and Agile development approach for 2G and 3G wireless systems [4–7], and is slated to repeat the same for beyond 3G, and 4G wireless protocol development. Consequently, we use SDL to develop a light version of the Long Term Evolution (LTE) protocol stack layer 2 (L2) and layer 3 (L3) for the mobile terminal.

1.1 Development Methodology

The block diagram in Fig. 1 shows the implementation steps of the LTE protocol stack using SDL Suite^{TM}. In the top left side, the Message Sequence Chart (MSC) [8] editor is utilized to produce the target design for the LTE user plane which is compatible with the LTE standard [9]. The MSCs are used as guidance for graphical modeling in SDL and C functions implementation for header processing. After mapping the graphical model to C code, the generated code is linked and compiled with the hand written C implementation. The overall system is simulated to check the functionality and compare it with the design target MSCs. Further corrections of the LTE models can be done when needed.

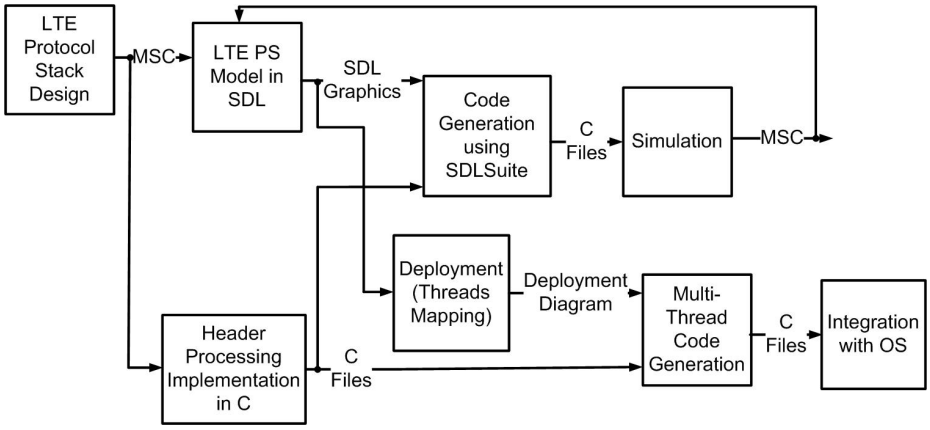


Fig. 1. Block diagram showing the LTE protocol stack implementation steps.

As shown in the lower branch of Fig. 1, the LTE system implementation can be integrated into an operating system. The modeled system is divided into separate threads using the deployment editor. The multithread SDL model is converted to C code, compiled, and linked with the hand written code.

This paper is organized such that Sect. 2 gives an introduction to the LTE protocol stack and its data flow. Section 3 shows the LTE user plane target design. The development environment is presented in Sect. 4. After that, Sect. 5

discusses the SDL model of the system in detail. Section 6 is dedicated to the integration with operating systems. In Sect. 7, the simulation results are presented followed by the conclusion and outlook to future research work in Sect. 8.

2 The LTE Protocol Stack

The LTE is the successor of the Universal Mobile Telecommunications System (UMTS) cellular technology. It enables much higher data rates to be achieved along with much lower packet latencies in an IP based system. The LTE will provide maximum data rates of 100 Mb/s in the downlink and 50 Mb/s in the uplink direction. Currently upcoming and future mobile devices enable the best quality of services in the current environment of the user with a future vision of an all IP based network. Another aspect is the dramatic increase of multimedia applications in the broadest sense including video streaming, video conferencing, and online gaming. Initiated in 2004, the LTE project focused on enhancing the Universal Terrestrial Radio Access (UTRA) and optimizing 3GPPs radio access architecture [9].

The LTE protocol stack L2 is divided into three sublayers, Medium Access Control (MAC), Radio Link Control (RLC), and Packet Data Convergence Protocol (PDCP). To summarize the flow of uplink data through all the protocol layers, an example with three IP packets is illustrated in Fig. 2. The PDCP performs IP header compression through Robust Header Compression (ROHC), followed by ciphering. A PDCP header is added, carrying information required for deciphering in the mobile terminal. The output from the PDCP is fed to the RLC.

The RLC protocol performs concatenation and/or segmentation of the PDCP PDUs and adds an RLC header. The header is used for in sequence delivery (per logical channel) in the mobile terminal and for identification of RLC PDUs in case of retransmissions. Several RLC PDUs are forwarded to the MAC sublayer, which assembles them into a MAC Service Data Unit (SDU), and attaches the MAC header to form a transport block. The transport block size depends on the instantaneous data rate selected by the link adaptation mechanism. Thus, the link adaptation affects both the MAC and RLC processing. Finally, the physical layer attaches a Cyclic Redundancy Check (CRC) to the transport block for error detection purposes. The physical layer processing is performed before transmitting the signal to the air interface [10]. The inverse functionality is done in the downlink when receiving a transport block. More details about uplink and downlink processing are given in Sect. 5.

3 The LTE Protocol Stack Design

As a first step of our implementation, as shown in Fig. 1, the MSCs are designed. Several scenarios are plotted according to the LTE standard as described in the 3GPP Rel8 [9]. As an example, Fig. 3 demonstrates the MSC for a target LTE protocol scenario. The MSC represents a looped LTE protocol data path from

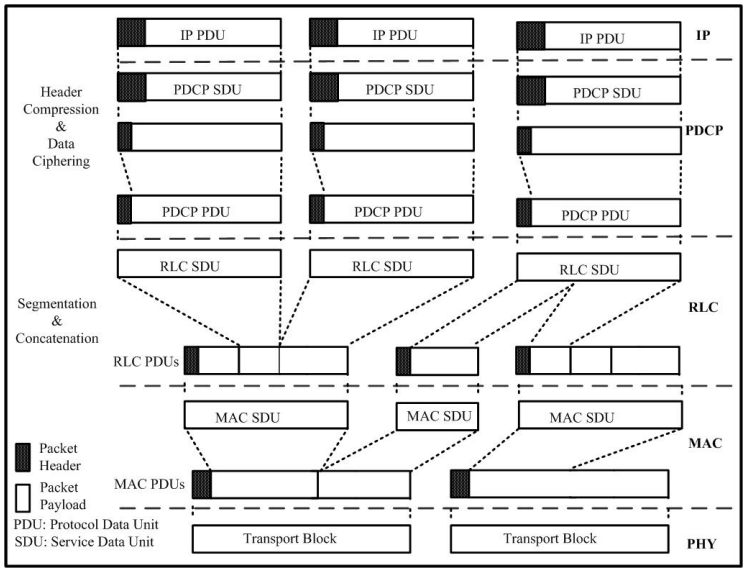


Fig. 2. The LTE protocol data flow in uplink direction from IP layer through L2 to the Physical layer (PHY).

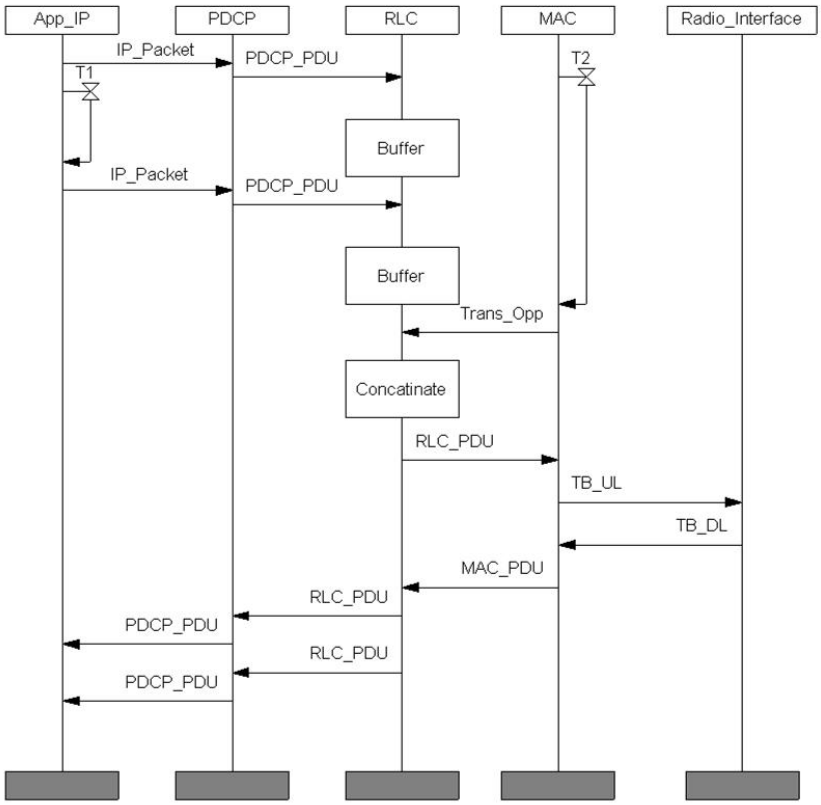


Fig. 3. MSC for the LTE data path in uplink and downlink showing packet flow from L3 through L2 and the transport block reception and processing in both layers.

the IP layer through L2 uplink to L2 downlink. Only the user plane is considered because its real time requirements are associated with longer execution time than the control plane. The MSCs are used for system analysis. The system can also be verified by comparing the designed MSCs with the MSCs generated by SDL Suite^{TM} simulator User Interface (UI).

The application data rate is realized using the timer *T1* and the IP packet payload length. In the uplink direction (see the upper part of Fig. 3) the IP packet should be processed in the PDCP sublayer and sent to the RLC sublayer to be concatenated with other PDCP PDUs depending on the transmission opportunity notification from MAC sublayer. The data rate in the mobile terminal is determined by the timer T2 together with the transport block size.

In the downlink, when MAC sublayer receives a transport block it should process its header and forward the RLC PDUs to the upper sublayers (see Fig. 3). The PDCP PDUs should be extracted from the received RLC PDUs and sent to PDCP sublayer to feed the IP layer with the IP packets.

4 SDL Environment

Formal Description Techniques (FDTs) are efficient in specifying complex communication protocols. FDTs guarantee syntactically and semantically unambiguous formal descriptions of communication protocols. In addition, they also guarantee interoperable and compatible implementations of these protocols independent of their implementers. This removes a lot of the anxiety of current software vendors to make their stack interoperable with other stacks developed by other vendors. Furthermore, the conformance of these protocols to a given standard can be checked with the help of predefined tests [11].

SDL supports object oriented software design by dedicated elements of the language in contrast to other FDTs. Furthermore, its quick to learn graphical notation is self documenting and thus easily maintainable [12]. These are the reasons why SDL was chosen to model the LTE protocol stack.

SDL is the most widely used FDT in the area of telecommunications. The basic theoretical model of an SDL system consists of a set of Extended Finite State Machines (EFSM) that run in parallel. These machines are independent of each other and communicate with asynchronous discrete signals. The LTE protocol stack model consists of the components described in the following sub sections.

4.1 SDL Structure

The LTE Protocol Stack (PS) is structured using the hierarchical decomposition with system, block, process, and procedure as the main building components. The main hierarchical levels in SDL are utilized to organize the LTE modeling. Every PS sublayer is represented by a sub block (that is MAC, RLC, and PDCP). The sub block in turn, is divided into sub sub blocks or processes according to the functionality or entities of the target layer. Some functionality

like header building/processing is implemented using external C functions to have more efficient code for accessing memory and doing other hardware closer functionalities.

4.2 SDL Communication

In SDL, global data is accessed by an exchange of signals: the language model does not allow direct reading from or direct writing to data owned by an enclosing block. Communication requires that information between processes, or between processes and the environment, is sent with signals (optionally with parameters). SDL signals are sent asynchronously: the sending process continues executing without waiting for an acknowledgment from the receiving process. PDUs are sent as signals. Synchronous communication is possible via a shorthand: remote procedure call. This shorthand is transformed to signal sending and waiting for a signal for the acknowledgment [1].

4.3 SDL Behavior

The dynamic behavior in an SDL system is described in the processes using EFSM. The system/block hierarchy is a static description of the system structure. Processes in SDL can be created at system start, or created and terminated dynamically at runtime. More than one instance of a process can exist. Each instance has a unique process identifier (Pid). This makes it possible to send signals to individual instances of a process. The concept of processes and process instances that work autonomously and concurrently makes SDL a true real time language [13]. The other advantage of the processes concurrency that it makes the parallelism easier to identify and exploit than the pure C programming.

5 The LTE Protocol Stack Model

The LTE protocol stack model in the SDL system is composed of two blocks called *LTE_PS* and *Radio_Interface*, as illustrated in Fig. 4(a). The radio interface forwarding functionality is modeled using two processes. The first for receiving the transport block and the other for forwarding the transport block to the downlink. As illustrated in Fig. 4(b), inside the *LTE_PS* block, there are four sub blocks, and 20 processes used to model the LTE protocol stack L2 and L3. All the models are implemented according to the LTE standard [14–16]. More details of the process modeling is given in the next subsections.

5.1 IP Packet Generation

The IP packet generation consists of two processes: one to emulate an application transmitting data and the other to build the IP header. The IPv4 [17] header with a length of 20 bytes is used in this implementation. The functionality is implemented using an external C function, which returns the pointer to the

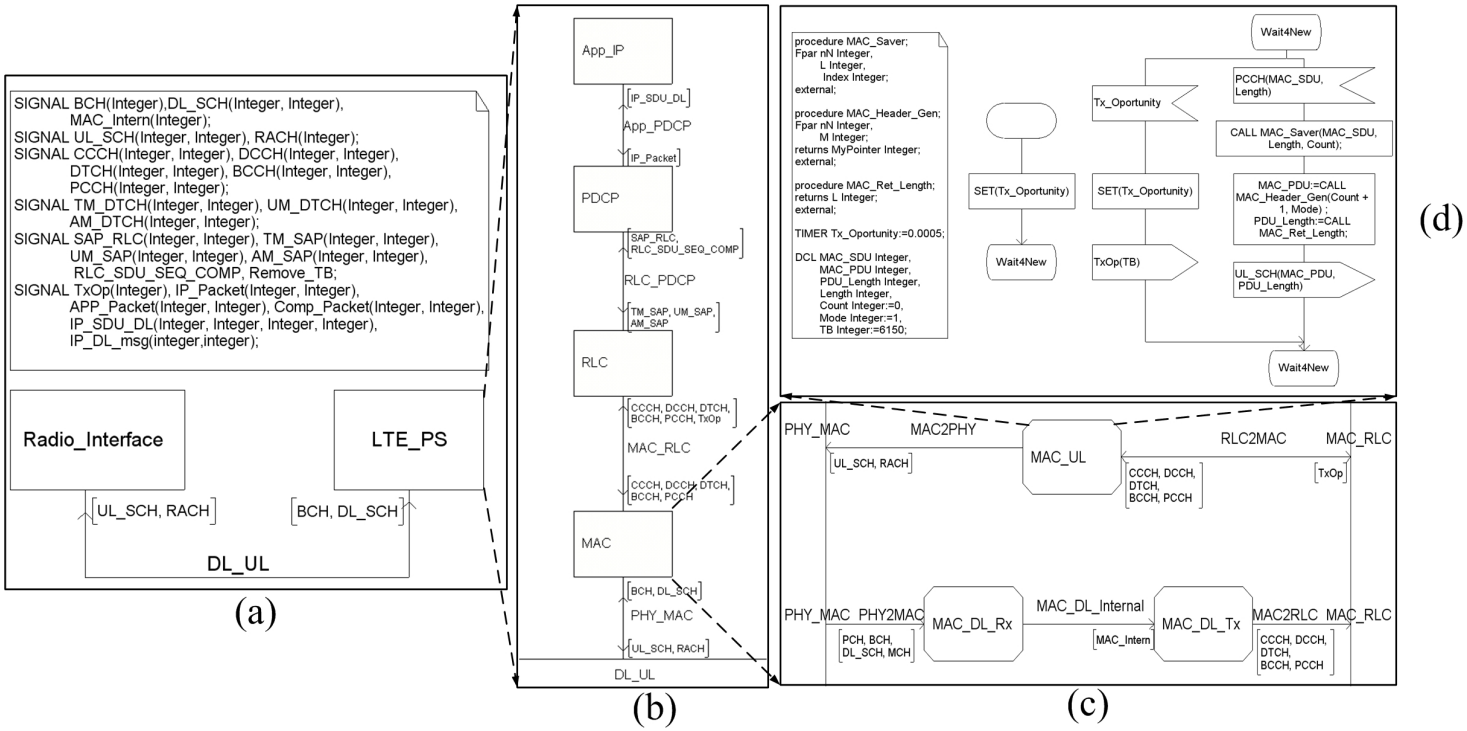


Fig. 4. The LTE protocol model hierarchy levels in SDL:
(a) SDL System, (b) The LTE protocol block, (c) MAC sub block, and (d) MAC uplink process.

IP packet, and is invoked inside the SDL process. The IP packet length and the pointer to the IP packet are forwarded to L2 by an SDL signal for further processing and header building.

5.2 PDCP Uplink

In the *PDCP* sub block, there are two processes used to model the LTE PDCP uplink. The first process receives the pointer to the IP packet from the upper layer. The Robust Header Compression (ROHC) and data ciphering are not included in this implementation because they are already implemented and tested with 3G system. The components with names relevant to the above tasks are not fully implemented but used only to forward the received packet. For every IP packet received from the upper layer, the second process appends a PDCP header and increments the sequence number counter. Then the pointer to the PDCP PDU is forwarded to the RLC sublayer.

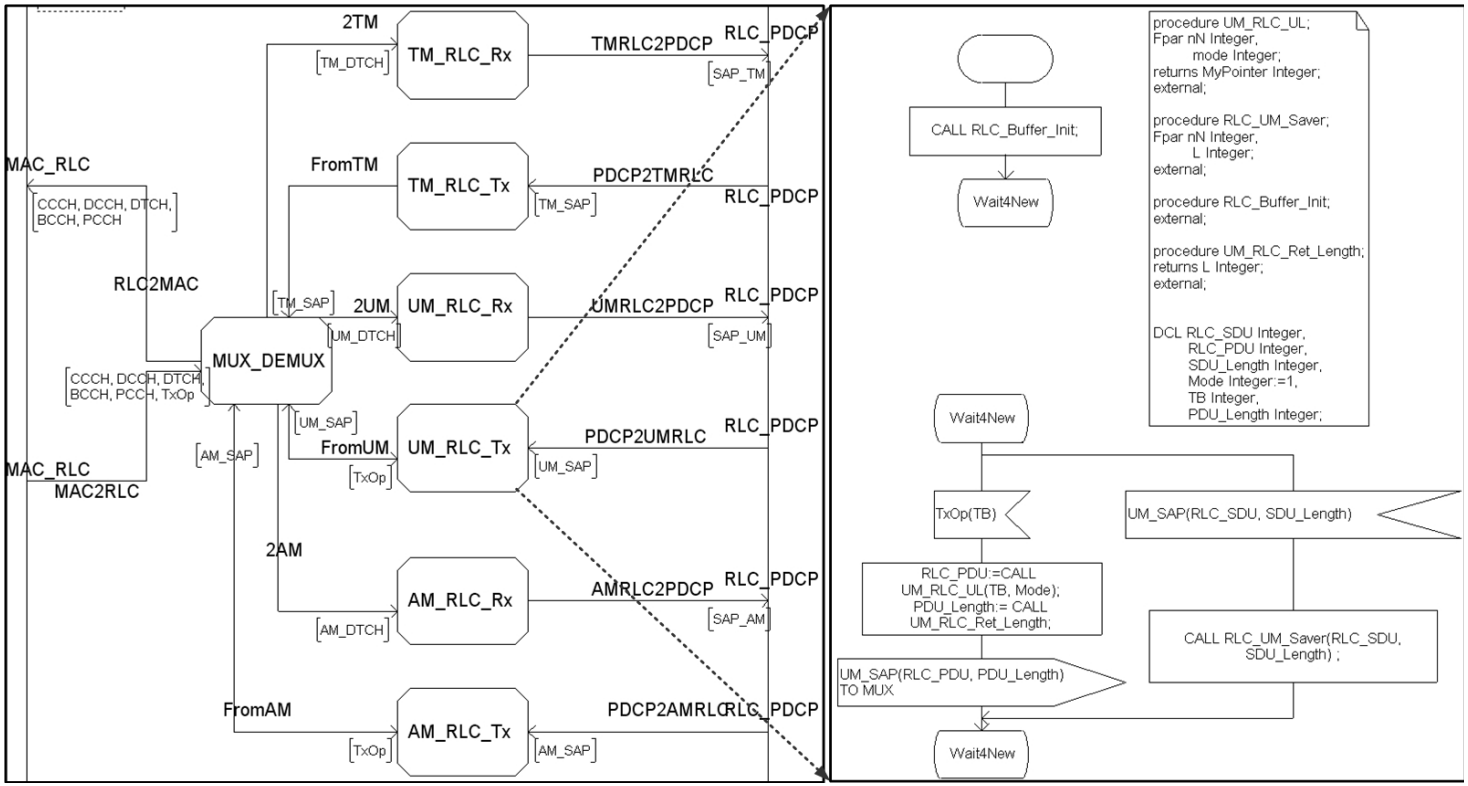
5.3 RLC Uplink

The RLC sublayer has three different modes: Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM). In the TM the RLC entity does not concatenate RLC SDUs nor add RLC headers, but forwards the received RLC SDUs to the MAC sublayer. On the other hand, when the RLC operates in the UM or AM, the RLC entity concatenates and/or segments the received RLC SDUs and adds the RLC headers to build the RLC PDUs. Then, the RLC PDUs are forwarded to the MAC sublayer. The main difference between AM and UM is that in the AM, the acknowledgement should be received from the receiving entity. The resegmentation can be done in the AM mode as well.

There are seven processes used to model the RLC functionalities in all three modes as shown in Fig. 5(a). The RLC entity saves the received RLC SDUs (that is, the PDCP PDUs) in a buffer and waits for the transmission opportunity indicated by the MAC sublayer. When receiving an indication from the MAC entity to transmit, combined with the length of the transport block allowed to be sent, the RLC SDUs are concatenated according to the transport block length to build the RLC PDU. In the TM entity (the *TM_RLC_Tx* process) no RLC header is built. On the other hand, the UM and AM transmitter processes build the RLC header and append it to the concatenated SDUs using external C functions. A pointer to the RLC PDU and its length are forwarded to the MAC sublayer, via the *MUX_DEMUX* process using the Dedicated Traffic Channel (DTCH), as shown in Fig. 5(b).

5.4 MAC Downlink

The received MAC PDU header is decoded using a C function. As a consequence, the number of multiplexed MAC SDUs, the logical channel identifier (LCID),



(a) (b)

Fig. 5. SDL modeling of the RLC sublayer:
(a) RLC sub block showing all processes for uplink and downlink and (b) Unacknowledged mode uplink process

and the length of every MAC SDU are used to forward the received SDUs to the higher layer via the dedicated logical channel as shown in Fig. 6. There are two processes used to model the above functionalities (see Fig. 4(c)). The first process decodes the received packet header and sends the number of MAC SDUs to the second process which in turn transmits every MAC SDU to the RLC sublayer via logical channels. The traffic data is sent via the DTCH.

5.5 RLC Downlink

The pointer to the RLC PDU is received by the *MUX_DEMUX* process and forwarded to the corresponding process, depending on the RLC entity mode. In the TM, the RLC PDU is forwarded to the PDCP layer without any processing according to [14]. On the other hand, in UM and AM, the RLC entity decodes the received header to check the starting address, the sequence number, and the length of every RLC SDU. As illustrated in Fig. 7, in the case that the SDUs are received out of order, the reordering is carried out and sent to the PDCP sublayer. If there is more than one copy of the same packet received correctly, only one will be processed and the other(s) will be discarded. The main difference between AM and UM is the retransmission of the packet which is not received at all or corrupted in the AM. As a consequence, the error correction using automatic repeat request (ARQ) is done in the acknowledged mode (see [14]). The ARQ procedure is not implemented in this version and a unidirectional channel is assumed.

5.6 PDCP Downlink and IP Packet Reception

In the downlink direction, the received RLC SDUs represents PDCP PDUs. The header of the PDCP PDU is decoded and the sequence number and packet length are used for in sequence delivery of the packets to L3. The duplicated packets are eliminated as well. After removing the PDCP header, the pointer to the beginning of the PDCP SDU is sent to the IP layer.

In the IP layer only the IP header checksum is calculated and evaluated for the received IP packet. The IPv4 header of length 20 bytes is removed such that the payload data can be sent to the application.

6 Integration with Operating Systems

The LTE system implementation is integrated into an operating system. The modeled system is divided into separate threads using the deployment editor. In our setup, every SDL process acts as one thread using multiplicity (*) on the aggregation between a component and the thread that contains the SDL block as illustrated in Fig. 8. The multi threaded model is converted to C code, compiled and linked with the hand written C code. With the threaded integration, SDL Suite^{TM} supports different operating systems like POSIX, Win32, VxWorks, and Nucleus Plus. As a consequence, the generated C code can be integrated with

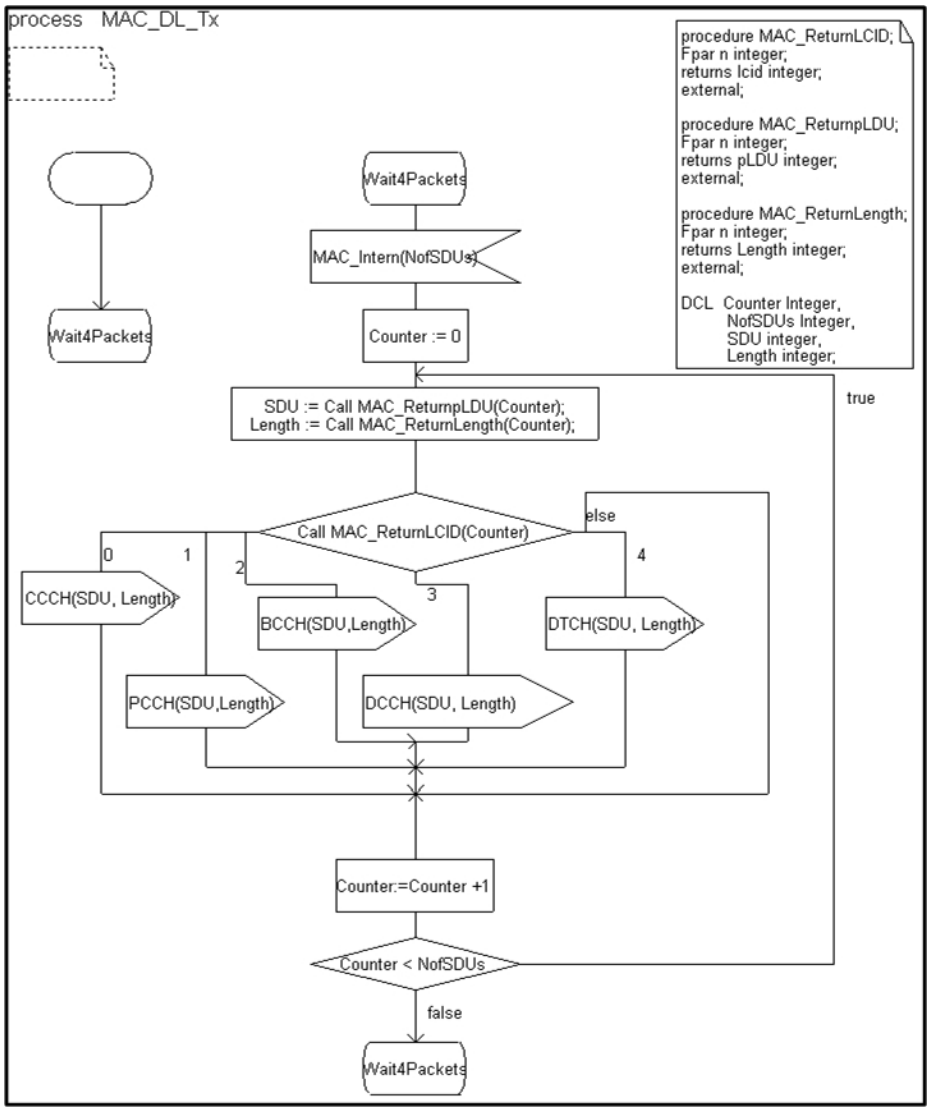


Fig. 6. SDL modeling of MAC DL shows transmitting MAC SDUs to different logical channels according to the LCID field in the received header.

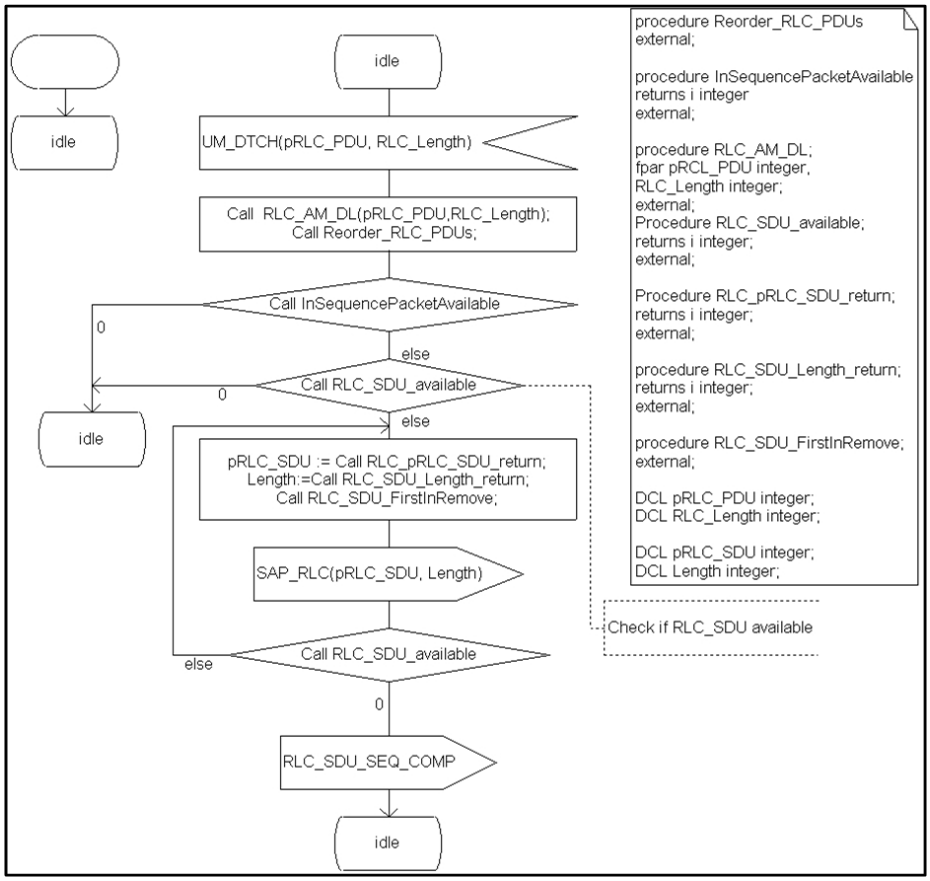


Fig. 7. SDL modeling of RLC DL shows reordering RLC SDUs and send them to the upper layers.

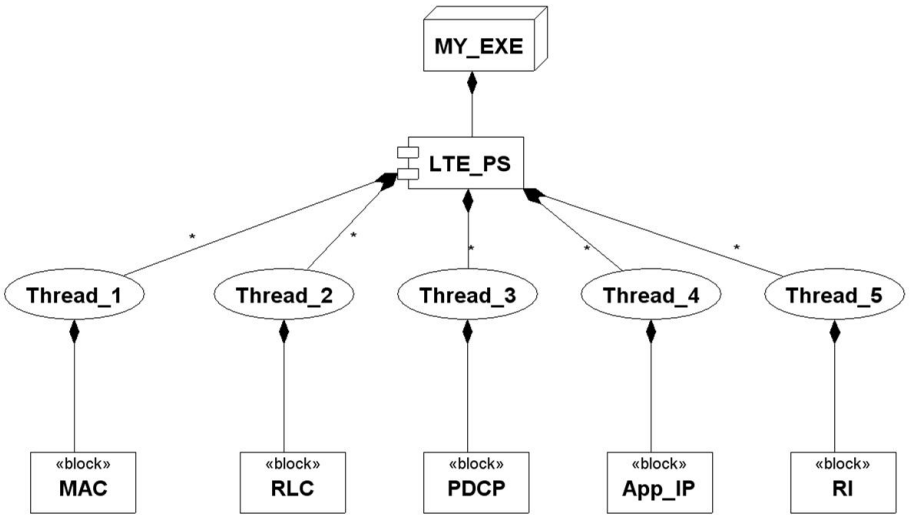


Fig. 8. The deployment diagram for the LTE protocol stack shows the multi-threading of the modeled system.

one of those operating systems. The integration with POSIX is done by taking the advantage of the standard *pthread* library for initiating, creating, starting and managing threads. The concurrency of processes makes the parallelism is easier to identify and exploit than the imperative programming style such as C.

We run the generated code of the designed LTE stack model on an embedded ARM11 processor with 32 KB cache size running at a frequency of 350 MHz. The observed average execution time for uplink and downlink processing is around 70 microseconds each. The development efforts for the modeling described in Sect. 5 and the integration with operating system is about 6 man/months including the learning phase for LTE and SDL.

7 Simulation Results

IBM® Rational® SDL Suite™ is a real time, software development solution that provides specification and development capabilities for complex, event driven communications systems and protocol software. The SDL Suite™ v6.1 is used to analyze the LTE model and to automatically generate C code. The generated code is compiled and linked with the hand written C implementation for header processing (see Fig. 1). The overall system is simulated to check the functionality and compare it with design target MSCs presented in Sect. 3.

As illustrated in Fig. 9, the IP packets propagate through PDCP, RLC and MAC sublayers to the radio interface. The SDL processes which are not relevant to the uplink are removed from the MSC for the sake of clarity.

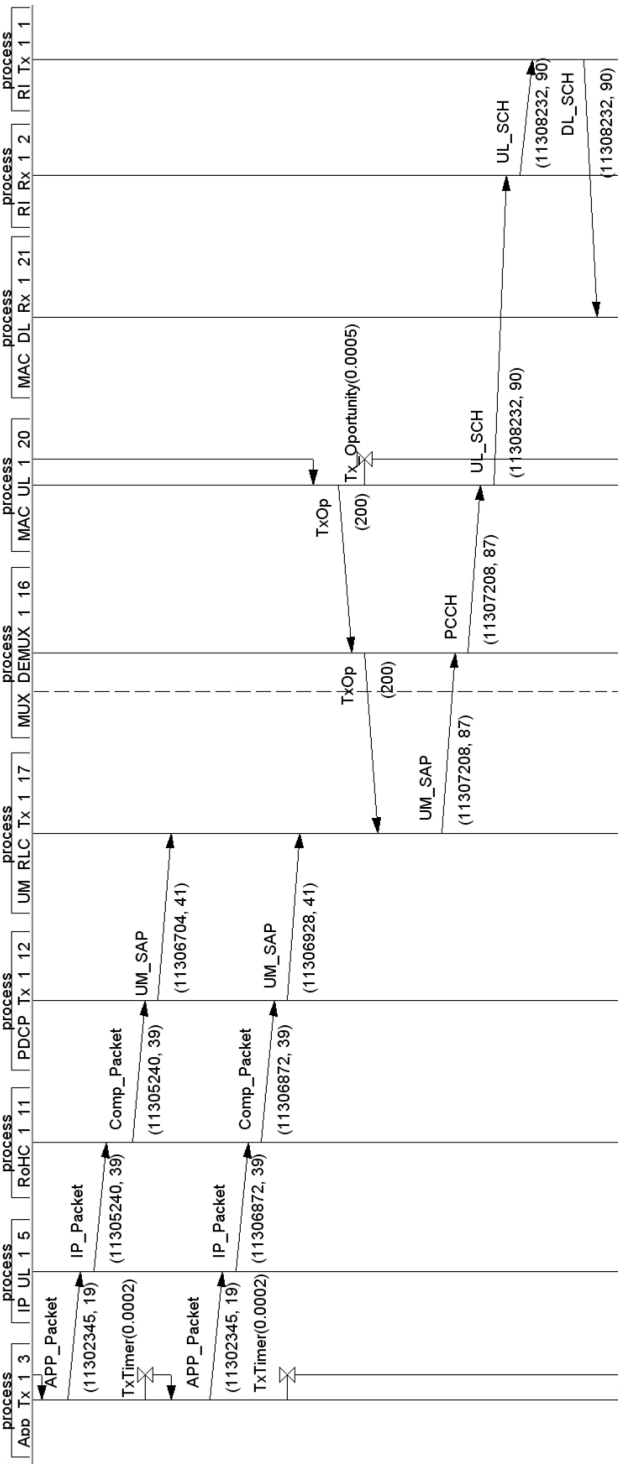


Fig. 9. The MSC trace as a simulation output shows the packet propagation through L2 and L3 for uplink.

The received transport block is processed in MAC, RLC, PDCP, and IP layers to extract the IP packet payload from it (see Fig. 10). The MSC in Fig. 9 and Fig. 10 are comparable to the target design MSC shown in Fig. 3. The main difference is the increased number of entities in the simulation output, because in SDL every process is treated as a separate entity. The SDL processes which are not relevant to the downlink are removed from the MSC for the sake of clarity. From a functionality point of view, the LTE protocol considered in this work is successfully implemented using SDL.

8 Conclusion

In this paper, we present efficient modeling of a light version of the LTE protocol stack in the UE side using SDL together with hand written C code. The overall system is simulated to check the functionality and compare it with design target MSCs which is compliant with 3GPP release 8 standard of the LTE. The simulation result shows that the implemented SDL guarantees a good consistency with the target test scenario, and has not any mismatching of logic flows as well as semantic errors.

The SDL model is decomposed into threads to enable execution in multi core as well as in single core platforms. For future work, we plan to extend the introduced LTE implementation, integrate it with any POSIX like operating system that support multi core and run them on a multi core embedded system.

Acknowledgments. The authors acknowledge the excellent cooperation with all partners within the ICT eMuCo project and the support by the European Commission. Further information is available on the project web site:
http://www.emuco.eu.

References

1. International Telecommunications Union: Recommendation Z.100 (11/07), Specification and Description Language (SDL), http://www.itu.int/rec/T-REC-Z.100/en
2. International Telecommunications Union: Recommendations Z.161 to Z.170 all (11/07) except Z.167 and Z.168 (11/08),
   Testing and Test Control Notation version 3: TTCN-3,
   http://www.itu.int/rec/T-REC-Z.161/en to
   http://www.itu.int/rec/T-REC-Z.170/en.
3. Badr, I.: Proven Approach for Developing Next Generation Communications Systems. ESE Magazine, vol. 15(6),
   http://www.esemagazine.com/pdf/dl_pdf/ESE-sep07.pdf
4. Hännikäinen, M., Knuutila, J., Hämäläinen, T, Saarinen, J.: Using SDL for Implementing a Wireless Medium Access Control Protocol. In: Proceedings of the 2000 International Conference on Microelectronic Systems Education, pp. 229–236. IEEE Computer Society, Washington DC (2000)

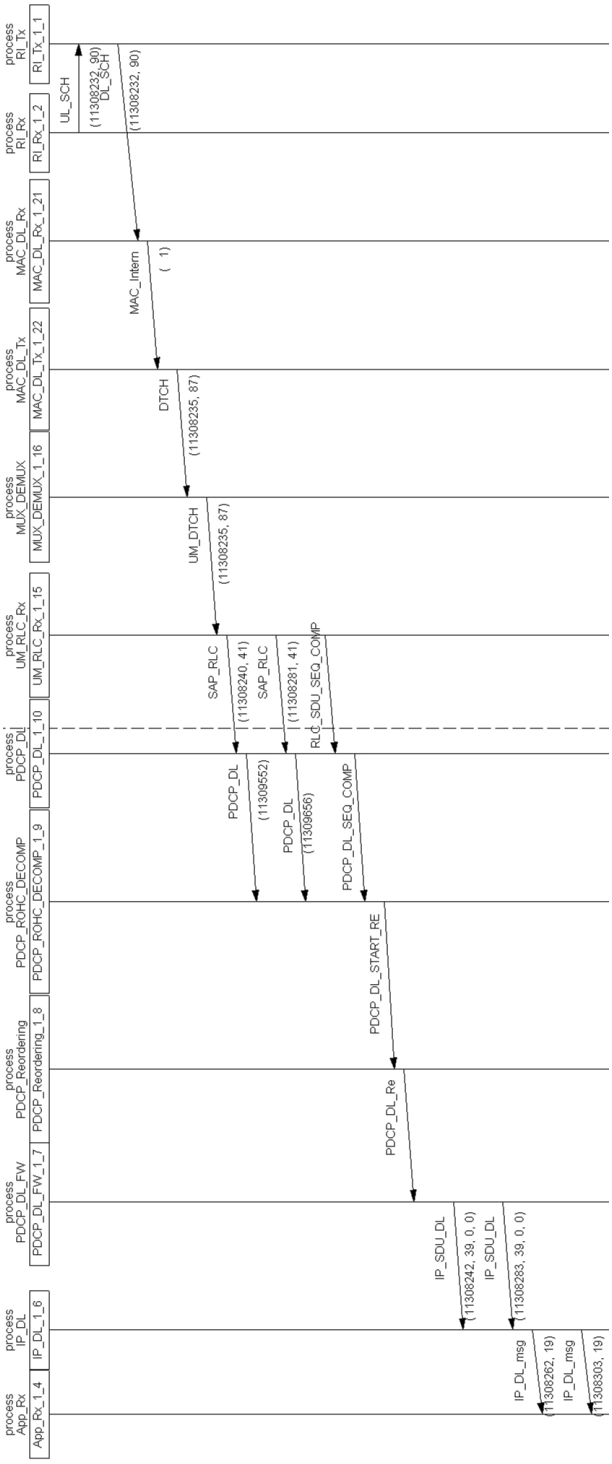


Fig. 10. The MSC trace as a simulation output shows the packet propagation through L2 and L3 for downlink.

5. Park, S-G., Shin, Y-S. Kim, D.Y.: Design and Implementation of Protocols Related to Radio Network Layer Over eNB IN 3GPP LTE System. In: 66th IEEE Vehicular Technology Conference, pp. 194–199. IEEE Computer Society, Washington DC (2007)
6. Álvarez, J.M., Cámara, P-d-l., Martínez, J., Merino, P., Pérez, F.C. and Morillo, V.: An SDL Implementation of the UMTS Radio Resource Control Protocol Oriented to Conformance Testing. In: Proceedings of the Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, pp. 397–401. IEEE Computer Society, Washington DC (2006)
7. Jung, K-R., Choi, J., Song, P., Nam, Y-H.: Design and Implementation of a Radio Resource Control Protocol in WCDMA using SDL. In: IEEE Vehicular Technology Conference (VTC) 54th, vol.2, pp. 990994. IEEE Computer Society, Washington DC (2001)
8. International Telecommunication Union: Recommendation Z.120 (04/04), Message sequence chart (MSC), http://www.itu.int/rec/T-REC-Z.120/en.
9. 3rd Generation Partnership Project (3GPP): The LTE Protocol Specification, 3GPP Rel8, http://www.3gpp.org/Release-8.
10. Dahlman, E, et al.: 3G Evolution: HSPA and LTE for Mobile Broadband, Academic Press, First Edition 2007.
11. Steppler, M.: Performance Analysis of Communication Systems Formally Specified in SDL. In: Proceedings of the 1st International Workshop on Software and Performance (WOSP) 1998, pp. 49–62, ISBN:1-58113-060-0.
12. Hogrefe, D.: Estelle, LOTOS und SDL-Standard Spezifikationssprachen für verteilte Systeme. Springer Verlag (1989)
13. IBM® Rational®: SDL Suite™ User Manual, SDL Suite™ v6.1
14. 3GPP TS 36.321: Evolved Universal Terrestrial Radio Access (E UTRA); Medium Access Control (MAC) Specification.
http://www.3gpp.org/ftp/Specs/html-info/36-series.htm.
15. 3GPP TS 36.322: Evolved Universal Terrestrial Radio Access (E UTRA); Radio Link Control (RLC) protocol specification. For URL see [14].
16. 3GPP TS 36.323: Evolved Universal Terrestrial Radio Access (E UTRA); Packet Data Convergence Protocol (PDCP) Specification. For URL see [14].
17. Internet Society, The Internet Engineering Task Force: RFC 791: Internet Protocol DARPA Internet Program Protocol Specification, September 1981, http://www.ietf.org/rfc/rfc0791.txt