# Evaluation methodologies for the NB-IOT system: issues and ongoing efforts

Samuele Foni, Tommaso Pecorella,
Romano Fantacci
Dpt. of Information Engineering
Università di Firenze
Firenze, Italia
samuele.foni@unifi.it,
tommaso.pecorella@unifi.it,
romano.fantacci@unifi.it

Camillo Carlini,
Pietro Obino
Telecom Italia S.p.A.
Roma, Italia
camillo.carlini@telecomitalia.it,
pietro.obino@telecomitalia.it

Maria-Gabriella Di Benedetto
Dpt. of Information Engineering,
Electronics and Telecommunications
Università di Roma La Sapienza
Roma, Italia
mariagabriella.dibenedetto@uniroma1.it

*Abstract*—**The recently standardized Narrow Band Internet of Things (NB-IOT) is expected to become a disruptive technology in the IoT market. The low cost, manageability, and reliability offered by a Mobile Network Operator (MNO) can be the enabler for the widespread acceptance and deployment of IoT devices. In this market phase, network planning plays a crucial role in the possible services forecast, with obvious consequences on the market analysis by the MNO. Moreover, there are many possible improvements in the NB-IOT device management, which are intentionally left as implementation dependent elements. As a consequence, it is necessary to evaluate the performances of an NB-IOT system, both for scientific and deployment purposes. In this paper we will outline the various evaluation systems and the ongoing efforts to build a new and efficient NB-IOT simulation platform.**

*Index Terms*—**Internet of Things, NB-IOT, LTE, Wireless Sensor Networks**

## I. INTRODUCTION

In 2016 $737 billion was spent in Internet of Things (IoT) applications and, according to the new data from International Data Corporation (IDC) [1] the IoT market will grow up with a rate of the 15.6% from the 2015 to 2020. As a consequence, telecommunication and service providers will have an increasing interest in the IoT systems.

The new radio interface proposed by 3GPP, named NB-IOT (Narrow Band Internet of Things) has been officially released in mid 2016 [2]. It has been built to meet the new market requirements related to the IoT growing needs. The main aim of this technology is to be able to adequately handle a massive number of connected devices and to prolong the battery operated nodes lifetime by using very aggressive sleep algorithms.

The features offered by NB-IOT will allow mobile operators to deliver new network services, adapted to a new User Equipment (UE) profile. Not surprisingly, this radio interface will become part of the fifth generation mobile connection system (5G). Currently there are only some physical prototypes implementing the NB-IOT standard, although the infrastructure needed is fully compatible with the LTE (Long Term Evolution) one.

Like in all the standards, also the NB-IOT specification have many elements that are implementation-dependent. Moreover, the deployment of this new system will require a precise evaluation of various parameters like terminal density, data gathering frequency, etc. In order to allow the academy and industry alike to study and develop new algorithms and services for the NB-IOT system, it is mandatory to have a set of reliable and efficient tools specifically suited for this task.

The available evaluation methodologies include simulation tools (e.g., MATLAB, ns-3, OMNeT++, etc.), emulation tools, testbeds, pilot sites, etc. Each method is best suited for a particular purpose, and none of them can be neglected.

At the moment the available tools for NB-IOT are not fully developed. We will describe in this paper the pros and cons of each tool and methodology, and we will outline our efforts to develop a novel simulation framework based on ns-3 [3].

This paper is structured as follows. In Section II we will describe and compare the current NB-IOT oriented tools. In Section III we will detail our system and its development status and road-plan. Sections IV and V will describe the preliminary results and the conclusions.

## II. NB-IOT EVALUATION METHODS

In general, two different approaches can be followed to evaluate the performance of a telecommunication system: analytical models or appropriate network evaluation systems.

The first strategy is based on the definition of a suitable mathematical model of the system. The system model is created through appropriate models (e.g., Markov chains), allowing to study the behaviour of the network (or parts of it) by simply changing the input parameters of the model. This approach is particularly effective for simple network protocols. However, it can easily become extremely complex, and some real-world effects are hard to reproduce (e.g., complex data generation patterns or the effects of active queue management techniques).

The second option is to use of a network evaluation system that can mimic the actual behaviour, providing an observable

result. In the world of telecommunications networks, this technique is based on the use of a simulator, a test-bed or an emulator.

The simulation allows to model complex scenarios at a high level of detail and to use real parts of code from actual implementations. It is based on the definition of a simulation model, which describes the evolution of the system over time. Through the analysis of this model it is possible to observe the dynamics of the system and evaluate its performance. In summary, the simulation process consists in reproducing the behaviour of a real system within a computer.

A test-bed is a platform used to perform a rigorous and transparent testing of a given device. Formally, a test-bed is a testing system that can stimulate another system with certain inputs, comparing the results obtained with the outputs expected. In telecommunications networks, the objective of a test-bed is to make an essential skeleton of the entire infrastructure that has to be tested before being put into production. Therefore, a test-bed is made using both software and hardware parts useful for setting up the entire system.

The purpose of emulation is to test the performance of a real application through a entirely virtual system. Emulators combine repeatability, configurability, isolation and manage-ability of simulators with test-bed realism, avoiding the use of complex physical architecture.

Finally, a pilot site is a technique often used to evaluate a given technology on a small-scale deployment and to extend the results (e.g., users satisfaction etc.) to larger use-cases.

Each technique has its pros and cons. Generally, the more realistic the method is, the longer the experiments needs to be to have statistically significant results. The different approaches, however, are not mutually exclusive: each one is best suited for specific needs and used in different technology development phases [4].

In this context the network simulation/emulation tools are especially useful for network prototyping and to evaluate with a high degree of flexibility a number of different parameters and techniques, like schedulers, priority queues, interactions between different technologies, etc. [5]

### III. NB-IOT SIMULATION MODEL

Among all the evaluation systems presented above, network simulators have the simplicity of software deployment, offer a replicable testing environment, manage controlled queued process and are able to act at any level of detail. An approach of this type is particularly suitable for exploring the evolution of a complex system, such as NB-IOT.

One of the key features of the simulators is the level of trust that the framework can offer to the scientific and industrial community, usually based on the results reproducibility and agreement with testbeds or real deployments. Another impor-tant element is the possibility to extend the models, in order to allow the integration with the pre-existing code, prevent code duplication, and increase the code maintainability. As a consequence, we will consider in this paper only Open Source network simulators.

The three major simulators are based on MATLAB [6], Omnet++ [7] and ns-3 [3].

The MATLAB one is named Vienna [8], and is able to accurately model the LTE Physical layer. However, MATLAB-based simulators are not a convenient choice for simulations involving complex network events, e.g., realistic data genera-tion patterns at application level.

Omnet++ has been recently extended to support LTE [9]. However, the model has been only recently introduced and its has not been yet widely adopted by the scientific community.

The ns-3 LENA module [10] is a well-known and tested LTE model suite, with a great support by its developing community. Moreover, it is being used by a very large user base, ensuring a good confidence on its results.

We decided to implement NB-IOT within the ns-3 network simulator for the following reasons. First, it provides support for network virtualization and allows the reuse of software for creating emulators and test-beds. Secondly, it simplifies the traceability of results, which can be exploited for post-analysis of the entire simulation process. Finally for its open source nature.

The LENA module can be used as a starting point in the process of building a NB-IOT simulator. LENA is a simulation model oriented to the development and testing of LTE-based standards and infrastructures. As a consequence, it should be relatively simple to extend it to support NB-IOT as well. Moreover, integrating NB-IOT in ns-3 / LENA will allow the simulation of complex scenarios and interference-limited use-cases.

The main actual difficulty relies in the LENA structure, which consists of a very large number of source files and classes. Moreover, there is the need to categorize the different NB-IOT features to be added in the simulator into logical groups in order to implement the mandatory ones and and the ones needed to simulate the specific use-cases of interest.

As a matter of fact, implementing the whole NB-IOT standard is a long-term goal, and it is more logical to split the development in phases, each one extending the support to extra features. One of the main goals of this work is to understand and evaluate how to properly modify the structure of LENA and to develop the actual NB-IOT foundation classes.

First, the substantial differences between the two tech-nologies LTE and NB-IOT have to be analyzed. Then, it is necessary to make a classification of the changes to be performed. After elaborating this information, a lot of indepen-dent modules can be identified, which can be developed and integrated like autonomous units within the LENA features, taking advantage of modular programming.

At the moment of writing, a number of software components have been identified, and a first module has been completed. This is the messaging system used to synchronize the downlink channel in a NB-IOT system, through the use of MIB-NB (Master Information Block - Narrow Band) and SIB-NB (Sys-tem Information Block - Narrow Band) messages, that are very different from those used in LTE. The use of these messages is fundamental in NB-IOT, because it is the operation fulcrum
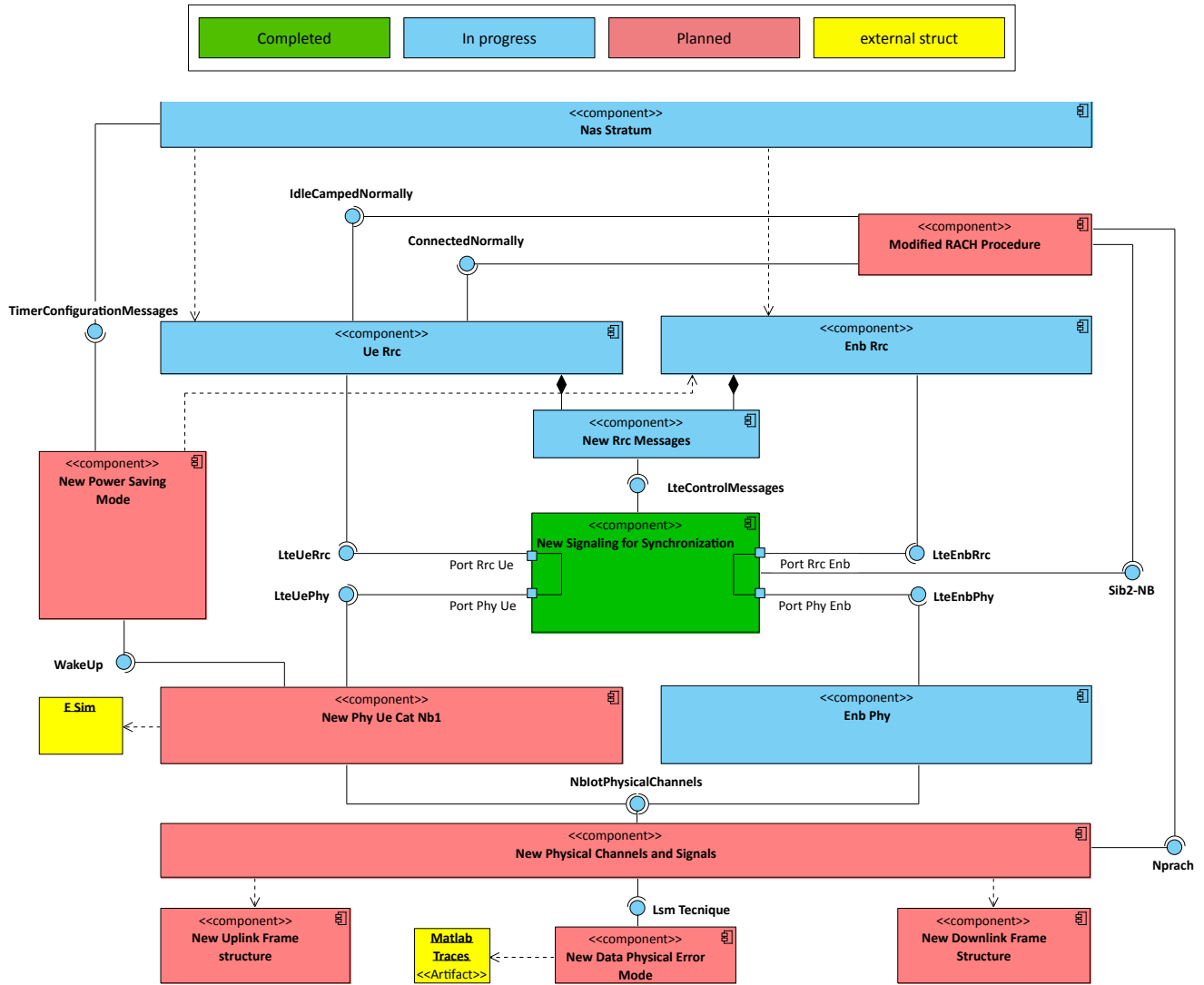
Fig. 1. UML component diagram outlining the main components necessary for NB-IOT.

of the entire control plane stack and it is a pre-requisite for establishing a new connection within this new radio interface.

The developed module has already been integrated within the LENA and tested for its operation. The original LENA code has been extended in order to preserve its functionalities. The current development, as we will describe later, has been mainly focused on the RRC and Physical layers.

### A. Analysis of the differences between LTE and NB-IOT

We will summarize here the main differences between LTE and NB-IOT components. All these points are relevant for the successful implementation of the simulator.

*1) Non-Access Stratum (NAS):* The NAS layer required by NB-IOT contains significant simplifications to the Core Network infrastructure of LTE. The operations needed to establish a connection on behalf of Evolved Packet System (EPS) are optimized, allowing non-IP-based data exchanges.

Furthermore, from an high level point of view the NAS has the task of managing the new random access channel procedure (RACH), used by the UE to establish a connection with an eNB. The NB-IOT RACH procedure is very different from the LTE one and its design can be one of the most critical parts of the simulator.

*2) RRC layer:* NB-IOT introduces a large number of new RRC messages and information elements devoted to manage a new UE category of devices and a new kind of logical connection procedures. In particular, the RRC layer has to manage logically the new traffic flows between the eNB and the UE and vice versa, and to synchronize properly both the uplink and the downlink channels.

Moreover, some features that were part of the LTE RRC layer are no longer needed for the NB-IOT architecture, such as the ETWS (Earthquake and Tsunami Warning System) and the CMAS (Commercial Mobile Alert System) notification

methods or like the InterRAT (Inter Radio Access Technology) support. At the same time, the RRC layer of the new NB-IOT standard is no longer responsible of the techniques and procedures used by LTE to manage the devices mobility, so handover checkup is no longer necessary.

*3) Middleware and MAC layer:* The greatest middleware innovation introduced in NB-IOT is the ability to manage the data connections in a complete state of energy saving. In particular, a UE sensors battery life of up to ten years is achieved through two innovative techniques: extended Discontinuous Reception (eDRX) and Power Saving Mode (PSM). The eDRX reduces the monitoring frequency of downlink control channels and the number of reports of the physical parameters measurements, while the PSM allows a deep suspension state to properly minimize their power consumption. When a sensor is in a PSM state it is in fact unreachable, while still being online. The sensor exits from this suspension mode using autonomous periodic wake-up procedures [11].

NB-IOT makes use of new procedures and new messages to synchronize downlink and uplink channels. The downlink ones are based on the MIB-NB and SIB1-NB messages, under the control of the RRC layer, while the uplink uses a new specific RACH procedure, defined at the NAS layer and strongly connected to the use of the SIB2-NB messages as well as to the Narrow Band Primary and Secondary Synchronization Signals (NPSS and NSSS respectively), which are created from the physical layer. All these messages are created and initialized by the middleware.

At MAC layer the changes are primarily in the format of Downlink Control Indicator (DCI) messages, which in NB-IOT can only be of three types: N0, N1, and N2, while for LTE there are six formats for the downlink operations and six for the uplink ones.

*4) Physical layer:* The biggest change that involves the physical layer of NB-IOT with respect to LTE is the introduction of a new UE category: the NB1 sensors [12]. NB-IOT introduces a large number of new messages and information elements devoted to the NB1 sensors management. Furthermore, these sensors do not require the use of standard SIM cards like smartphones. Instead, they use special virtual SIMs called e-SIM. The chip that realizes the e-SIM is integrated into the NB1 sensors and is directly configurable by the network operator [13]. These features allow providers to easily support and manage a large number of devices.

NB-IOT introduces five new physical channels to handle communications: three for the downlink (NPBCH, NPD-CCH and NPDSCH) and two for the uplink (NPUSCH and NPRACH), plus four new physical level signals. Three for downlink (NRS, NPSS and NSSS) and one for uplink (DMRS). Moreover, it brings an innovative mapping method of the old LTE transport channels in these new physical channels [14].

Finally, the bandwidth used by the NB-IOT systems is always 180 kHz, while the LTE ones is variable. This feature stems from the design choice to reserve a single Physical Resource Block (PRB) for NB-IOT transmissions in order to minimize energy consumption. This is a common goal in all the Low-Power Wide-Area Network (LPWAN) technologies, i.e., to limit the throughput to minimize the energy waste.

### B. Identification of the new software modules to be developed

The new NB-IOT radio interface shows a long series of innovations, as emerged from a rapid analysis of its architectural differences, compared to those commonly used by LTE. The two could be seen as distinct development branches, but of course they must be both integrated into the LENA model.

NB-IOT is a complex system and both its analysis and its development require a high degree of detail. As a consequence, it is necessary to use a modular programming paradigm to adequately plan all the changes to be made to the LENA structure in order to integrate all the new components required by the NB-IOT standard.

The main advantages deriving from the use of this approach lie in simplifying the development, testing and maintenance of modular software. In this way, each module performs a particular and well-defined task, providing interfaces with both the outside and the other modules of the system, obtaining a series of isolated entities that can be developed separately and independently.

The component diagram in the figure 1 summarizes the architectural choices that were made in the first design phase of the NB-IOT system. The components highlighted within the graph show the logical link between the various modules and the set of modifications that must be actually made to the ns-3 source code.

## IV. RESULTS

The analysis of the differences between LTE and NB-IOT led to the identification of a number of software components to be developed in order to integrate NB-IOT into the ns-3 LENA.

A first useful module for downlink channel synchronization in NB-IOT communications has already been developed. However, all the missing components must be completed to ultimate the development work.

Particular attention must be paid to the creation of low-level software components, especially those belonging to the physical layer, which radically changes with respect to LTE. As an example, it is necessary to use new BLER curves for the NB-IOT modulation and coding schemes, to be derived from physical level simulators, e.g., the Vienna LTE-A Simulator [8]. Under the hypothesis of in-band scenario, it is possible to use the existing BLER curves. However, it will be necessary to extend the existing model to support also the out-of-band and guard-band scenarios.

We concentrated our efforts on the NB-IOT UE and eNB signaling phases, and the related state machines that guarantee the proper devices status evolution, from their wake-up to the communication and disassociation phases.
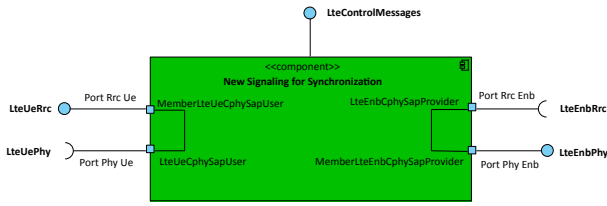
Fig. 2. Detail of the new signalling module.

### A. New signaling for downlink synchronization component

The NB-IOT signaling protocol uses new messages named MIB-NB and SIB1-NB. These messages are different from those used within the LTE standard. In particular, the content, the forwarding period, the physical channel associated with them, and the scheduling type needed to receive the following messages correctly are completely different. Despite the differences, using polymorphism guarantees that the eNB can handle all the messages in the same way, differentiating its behavior according to the specific UE and message.

The new MIB-NB message scheduling rules are listed in the official 3GPP TS 36.331 [15], sec. 5.2.1.2a. The first transmission of a new MIB-NB message must be scheduled in subframe number 0 of a set of 64 radio frames. Each MIB-NB message must be repeated eight times. Moreover, a single MIB-NB message is divided into eight parts and each of them needs to be transmitted at the beginning of each subsequent subframe.

TS 36.331 [15] also contains the specifications of the new SIB1-NB messages periodicities. In this case, the NB-IOT standard requires that, for the first transmission of a SIB1-NB message, the eNB selects the first Physical Cell ID (PCID) received within a radio frame, which is derived from the corresponding cell. A new SIB1-NB message is transmitted every 2560 ms, or rather each 256 radio frames. Within this period, this message can be repeated for a variable number of times, depending on the value provided by the *schedulingInfoSIB1* parameter contained within the last MIB-NB received message. 4, 8 or 16 repetitions are possible, and 4 transport block sizes of 208, 328, 440 and 680 bits are defined.

The RRC level can be described, both from the point of view of the eNB and the UE, through a finite states machine. Each state aims to keep track of the history of the procedures that have been taken to reach it, and the messages that have been received so far. Each state machine evolves according to the received MIB-NB and SIB1-NB messages.

NB-IOT standard requires that an UE has received at least one MIB-NB message *and* one SIB1-NB message before it can proceed with the RACH process and establish a connection with a eNB. Therefore, focal point for the development and the implementation of the new component, shown in figure 2, is the algorithm that manages the change of status of the eNB and the UE when the MIB-NB or SIB1-NB messages are generated or received.

The finite states machine describing the eNB state changes

are coordinated and monitored by the *UeManager* class, whose instance is created specifically from the RRC level of the eNB to handle the connections of an UE, which is capable to interact and invoke methods of an *LteEnbRrc* object.

The evolution of the *UeManager* finite states machine in the NB-IOT case is entirely analogous to that foreseen by the LTE standard, except for some missing states, e.g., handling handover situations, that are not possible in NB-IOT.

The states machine running at the UE RRC layer is very different from that used by normal LTE UEs. As a matter of fact, while for a smartphone is sufficient to receive a MIB or a SIB1 message in order to be able to make a connection via RACH, a NB1 sensor needs to receive both a MIB-NB *and* a SIB1-NB message before it can make a regular connection request. As for the interpretation of any handover triggers, these must be handled as reconfiguration signals, just as in the case of the eNB. At the implementation level, the class that cares about implementing the evolution of the machine to new UE states is *LteUeRrc*.

### B. Validation test of the developed component

After completing the development phase of the new component, it was necessary to verify its integration within the ns-3 code. To do so, after a preliminary phase of debugging, we developed some unit tests to verify the correct encapsulation, forwarding, and receiving procedures execution for the new MIB-NB and SIB1-NB messages.

Unit tests aim to verify the correctness of the individual algorithms, the interfaces, etc. Testing an algorithm or a component is usually performed through the code execution of known inputs, and the verification that the output is the expected one. Interfaces verification aims at controlling that the component is robust against bad or unexpected behaviors introduced by other software components.

In ns-3, tests are of both types, and they cover the most critical code sections. Individual unit tests are instantiated using the *TestCase* class, and each code variation is verified against the existing tests (or also new ones, in case of new models/components). The model followed by ns-3 is to perform: "a test for each feature" and "a test for each method" [16]. In our work, we have used this approach.

We performed tests about MIB-NB and SIB1-NB messages exchange between the RRC level of the eNB and the UE. The UML sequence diagram in figure 3 summarizes a MIB-NB message creation and send process by the eNB, and it has been used as a guideline for the test development. Similar UML diagrams have been developed for MIB-NB reception and SIB1-NB transmission and reception.

The tests highlighted a correct encapsulation of the messages, including the MIB-NB splitting and repetitions. Moreover, we verified that the state machines in the UE and eNB evolution over time was correct, including the timeout events related with the missing reception of messages.

Moreover, we tested also the correct message exchanges between RRC and Physical layer, including the multiplexing and demultiplexing in the new NB-IOT channels.
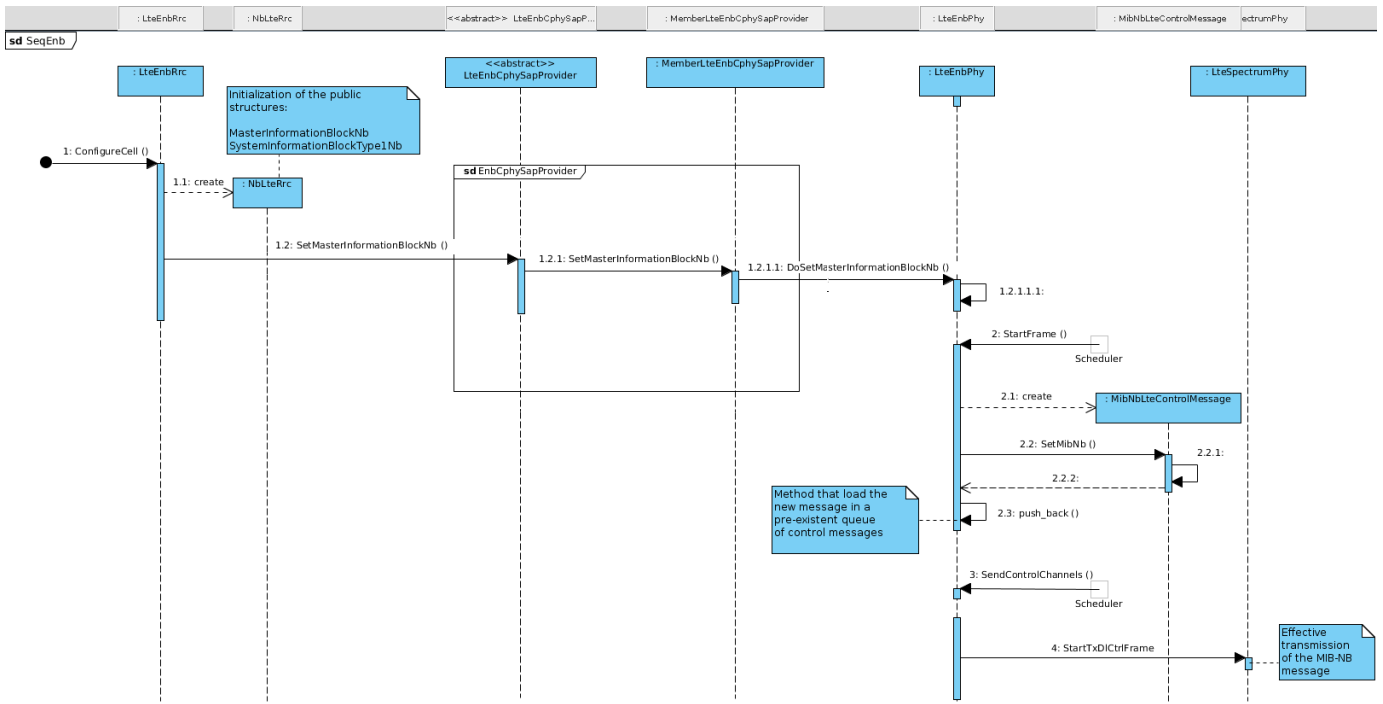
Fig. 3. UML Sequence Diagram: operation flow to generate and send a MIB-NB message by the eNB.

Therefore, it can be stated that we obtained the expected results, and that the tests performed were successful.

## V. CONCLUSIONS

In this paper we presented the main issues in developing a NB-IOT network simulator. We highlighted the differences between NB-IOT and LTE, and how these can be mapped in the actual ns-3 LTE model.

The module development is tracked in the ns-3 wiki page (https://www.nsnam.org/wiki/NB-IOT). We plan to improve the model, also with contributions from the ns-3 and NB-IOT developers community. Moreover, we expect to add the *consolidated* NB-IOT features into the main ns-3 development branch, where each feature will be considered *consolidated* when all the standard ns-3 code best practices are met, i.e., presence of unit testing, Sphinx and Doxygen code documentation, and code reviewed by the community.

## REFERENCES

[1] International Data Corporation (IDC). Worldwide Semiannual Internet of Things Spending Guide, 2017H1. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS42209117

[2] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation," 3rd Generation Partnership Project (3GPP), TS 36.211, Mar. 2017.

[3] "ns-3." [Online]. Available: http://www.nsnam.org

[4] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 Project Goals," in *Proceeding from the 2006 Workshop on Ns-2: The IP Network Simulator*, ser. WNS2 '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1190455.1190468

[5] L. Giupponi, T. Henderson, B. Bojovic, and M. Miozzo, "Simulating lte and wi-fi coexistence in unlicensed spectrum with ns-3," *arXiv preprint arXiv:1604.06826*, 2016.

[6] "MATLAB." [Online]. Available: http://www.mathworks.com/products/matlab.html

[7] "OMNeT++." [Online]. Available: http://www.omnetpp.org

[8] M. Rupp, S. Schwarz, and M. Taranetz, *The Vienna LTE-Advanced Simulators: Up and Downlink, Link and System Level Simulation*, ser. Signals and Communication Technology. Springer Singapore, 2016.

[9] A. Virdis, G. Stea, and G. Nardini, *Simulating LTE/LTE-Advanced Networks with SimuLTE*. Cham: Springer International Publishing, 2015, pp. 83–105. [Online]. Available: https://doi.org/10.1007/978-3-319-26470-7_5

[10] "LTE-EPC Network simulAtor." [Online]. Available: http://networks.cttc.es/mobile-networks/software-tools/lena/

[11] 3GPP, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," 3rd Generation Partnership Project (3GPP), TS 24.301, Aug. 2017.

[12] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities," 3rd Generation Partnership Project (3GPP), TS 36.306, Aug. 2017.

[13] A. Calvi, "I sistemi di comunicazione per lo smart metering di seconda generazione." [Online]. Available: http://www.autorita.energia.it/allegati/eventi/151211CalviTelecom.pdf

[14] D. R. J. Schlienz, "Narrowband internet of things," 2016. [Online]. Available: https://www.rohde-schwarz.com/us/applications/narrowband-internet-of-things-white-paper_230854-314242.html

[15] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.331, Aug. 2017.

[16] ns-3 project. Testing framework, ns-3 manual. [Online]. Available: http://www.nsnam.org/docs/manual/html/test-framework.html