

Modeling unicast device-to-device communications with SimuLTE

Antonio Viridis, Giovanni Nardini, Giovanni Stea

Dipartimento di Ingegneria dell'Informazione, University of Pisa
Largo Lucio Lazzarino 1, I-56122, Pisa, Italy
a.viridis@iet.unipi.it, g.nardini@unipi.it, giovanni.stea@unipi.it

Abstract—In LTE-Advanced (LTE-A), device-to-device (D2D) transmissions allow two peering User Equipments to communicate directly without using the Evolved Node-B as relay. D2D is regarded as one of the enablers to bring LTE-A in the context of vehicular networks, smart cities, or M2M applications. Research on this topic is mostly carried out through link-level simulations. In this work, we describe instead the modeling of D2D into a system-level simulator, namely SimuLTE, which enables us to analyze the performance of applications and higher-layer protocols using D2D transmission. We first describe the modeling within the SimuLTE architecture, then we validate it and analyze the performance of D2D communications with frequency reuse.

Keywords—LTE-Advanced, D2D, resource allocation, system-level simulation.

I. INTRODUCTION

Device-to-device (or *direct*) communications allow two LTE-A User Equipments (UEs) to communicate directly, without using the eNodeB (eNB) as a relay. This can happen either with or without the assistance of the network. Network-unassisted D2D is foreseen for coverage extension or contingency scenarios: in this case, the two UEs occupy frequency resources for their communication autonomously, possibly using a cognitive approach. With network-assisted D2D, which is the object of this paper, the eNB instructs the receiving UE to listen on the same resource blocks (RBs) granted for transmission to the transmitting UE, hence exchanges *control* information with both, but is not involved in *data* exchange. Using network-assisted D2D, neighboring UEs can communicate at a low power. This, in turn, allows the eNB to reuse the same frequency for other D2D pairs – as well as to save the power otherwise involved in supporting the uplink and downlink legs of a relayed transmission between the same endpoints. Network-assisted D2D can support several services, for instance inter-vehicle communications [15], Internet-of-Things (IoT) applications in a fog networking paradigm [8], or TCP-based file exchange [7]. D2D communications can be both *unicast* and *multicast*. Multicast D2D can be used for proximity-based services (e.g., advertisement, alerts, e-gaming), whereas unicast D2D is used for peer-to-peer services, e.g. file transfer, browsing or multimedia.

In the literature, there are indeed many works related to unicast and multicast D2D transmissions, e.g., [4]. However, quoting [5], most of these are based either on analytical tools or on ad-hoc simulators, which can only focus on a subset of the

key aspects of the system. However, the issues of how D2D transmissions affect higher-layer protocol operations (e.g. TCP congestion control) are largely unexplored, the only work we are aware of being our previous works [6],[7]. The above preliminary works show that cross-layer issues between higher-layer protocols and D2D transmissions are significant, and non-intuitive phenomena may occur. In order to be able to look deeper into these cross-layer interactions, a modeling of D2D transmissions into a system-level simulator is required. Moreover, D2D-communications are interference-constrained, thus resource-allocation algorithms have to take these effects into account, for example by embedding either the *protocol* or the *physical* interference models in the algorithm itself [9]. However, this must be validated using a realistic channel model that accounts for interference by computing the Signal-to-Interference-and-Noise-Ratio (SINR) at the receivers.

In this paper, we first describe how we model unicast D2D communications from a system-level perspective and how to integrate such model into the SimuLTE simulator [2]. We will focus mainly on the resource allocation function and on how the interference is computed. Then, we validate the proposed model via simulation and we evaluate the effects of interference on both system-level KPIs, such as the application-level throughput, and on the running time of the simulator itself.

The rest of the paper is organized as follows: Section II reports some background on LTE-Advanced and D2D transmissions. Section III describes the SimuLTE simulator, while Section IV details our D2D model and how the latter is integrated into SimuLTE. We report validation and evaluation results in Section V, and conclude the paper in Section VI.

II. BACKGROUND ON LTE-A

This section provides some background on LTE-A, particularly focusing on its protocol layering and resource allocation, and on the features of D2D communications in LTE-A.

The LTE-A protocol stack consists of four layers (Fig. 1), henceforth described top to bottom: an IP packet entering an LTE interface first traverses the Packet Data Convergence Protocol (PDCP), where it is cyphered and assigned a sequence number. It is then sent down to the Radio Link Control (RLC) layer, in the form of an RLC SDU, and it is buffered there. The RLC can be configured in one of three modes, transparent (TM), unacknowledged (UM) and acknowledged (AM). UM is

recommended by the standard for D2D, and it performs segmentation/concatenation of RLC SDUs on transmission, and reassembly, duplicate detection and reordering of RLC PDUs on reception. The MAC sits below the RLC and requests to it an RLC PDU of a given size. The RLC complies by dequeuing from its buffer one or more RLC SDUs and combining them as necessary into RLC PDUs. The MAC adds its own header and forms a MAC PDU, also called Transmission Block (TB).

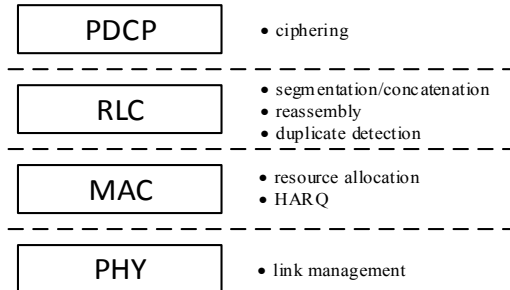


Fig. 1. LTE-A protocol layering and main functions

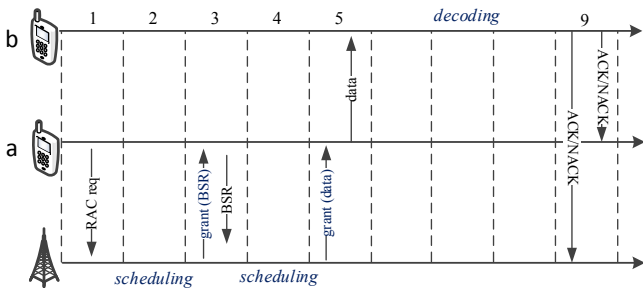


Fig. 2. Procedures for data transmissions

Resource allocation is done by the eNB at the MAC layer. In the downlink (DL), the eNB sends a *subframe*, i.e., a vector of *Resource Blocks* (RBs) containing the MAC PDUs for the UEs, at Transmission Time Intervals (TTIs) of 1ms. An RB carries a different number of bits depending on the modulation that the eNB will use. The latter is selected based on the Channel Quality Indicator (CQI) reported by the UE, which reflects its measured Signal to Interference and Noise Ratio (SINR). A Hybrid ARQ (H-ARQ) scheme, which allows a configurable number of retransmissions, provides reliability at the MAC layer. Downlink H-ARQ is *asynchronous*, i.e., the eNB may schedule retransmissions at any future TTI.

Uplink (UL) subframes carry UE traffic destined to the eNB. The latter issues *transmission grants* to the UEs, specifying which RBs they can use, using what transmission format. Since UL buffers reside at the UEs, UEs must send *Buffer Status Reports* (BSRs) to report their backlog. The latter are transmitted *in band*, possibly trailing a data transmission, whenever the UE is scheduled and has enough space to do so (a BSR can take up to 24 bits). A UE can signal new backlog through an *out-of-band* Random Access Procedure (RAC). RAC requests are responded in-band by the eNB, which schedules the UE in a future TTI. Unanswered RAC requests are re-iterated. Uplink H-ARQ processes are *synchronous*, i.e., the eNB schedules retransmissions with a lag of eight TTI. In both DL and UL, H-ARQ ACKs/

NACKs are sent by the receiver four TTIs later than the transmission they refer to.

A. Unicast device-to-device communications

Device-to-device (D2D) communications, also called *direct* communications, are being currently discussed in the standardization bodies [10]. The ongoing discussion is mainly on *multicast* transmissions, whereas little, if anything, is standardized about *unicast* transmissions. Thus, we stick to what appears to be widely agreed upon in the literature on the subject henceforth. With unicast D2D, UE *a* can send a packet to UE *b* without having it traverse the usual two-hop path through the eNB: instead, the eNB can grant *a* one or more RBs for transmission and instruct *b* to listen on the same RBs for reception. Note that *b* must be equipped with a Single-Carrier Frequency Division Multiplexing (SC-FDMA) receiver for this to happen. The D2D link is often called *sidelink* (SL), to distinguish it from the UL and DL. In a frequency-division duplexing, the SLs are carved out of UL frequency resources, where interference is expected to be less severe [11].

Note that resource allocation is still performed by the eNB: in particular, with reference to Fig. 2, *a* must request a SL grant in much the same way as it would an UL one, and *b* must be able to ack/nack it and to report the SL CQIs to the eNB. Thus, the eNB is still in charge of the *control plane*, even though it does not participate into data transmission.

There are several open points regarding D2D unicast transmission. One is how to identify the endpoints, possibly using device class, subscription profile, geographical position, interference conditions, flow type etc.. Another problem is how to decide when to switch an ongoing communication from the SL to the two-hop path through the eNB (henceforth, the *infrastructure mode*, *IM*) or back, and whether and how this can be done seamlessly (see, e.g., [6],[7],[12],[13]). Both problems can only be solved by working on the control and management planes (and, possibly, on the higher layers of the LTE protocol stack), i.e., need be tackled through a system-level simulator.

III. SIMULTE ARCHITECTURE

In this section we provide an overview of SimuLTE, with focus on those modeling aspects that will affect the implementation of D2D functionalities. SimuLTE is a system-level simulator, available for download at [17]. It is based on the OMNeT++ [1] framework, which is built around the concept of *module*, a basic modeling unit which communicates via message exchanges and that can be hierarchically organized in *compound* modules. Each module is characterized by a *structure*, defined via `.ned` files, and a *behavior*, implemented via C++ classes.

The core SimuLTE module is the LTE NIC card, based on which one can instantiate nodes with LTE capabilities, specifically UEs and eNBs. These nodes exploit the LTE NIC as an interface, and add as upper layers modules from INET, i.e. a set of OMNeT++-based modules developed by the community, to simulate standard Internet entities. The INET library is also used to implement entities outside the LTE scope, such as ap-

plication servers, that are used as traffic generators/receivers and communicate with the application within the UEs. A high-level representation of the nodes is given in Fig. 3. The NIC card implementation allows one to develop nodes with multiple connectivity capabilities (e.g. LTE and/or Wi-Fi), and fully embodies the modularity paradigm on which the OMNeT++ framework is based.

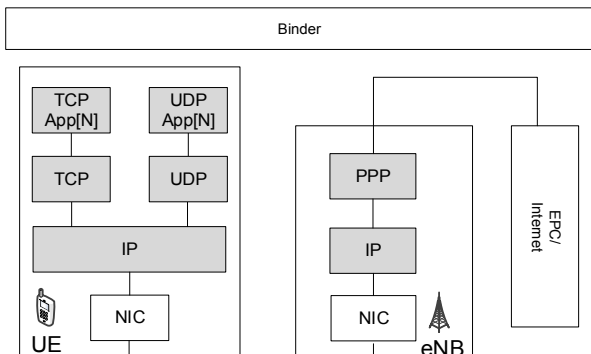


Fig. 3. System overview

The NIC cards in the UE and eNB nodes are organized by layers, namely PDCP, RLC, MAC and PHY, with a one-to-one correspondence with the LTE protocol stack. Each layer includes both *common* and *node-specific* functionalities. In fact, OMNeT++ allows inheritance of both the structure and the behavior of modules. Considering the MAC layer as an example, we have the *MacUe* and *MacEnb* classes, both extending the *MacBase* class with node-specific functions, such as the resource scheduler on the eNB side.

In SimuLTE, data transmission and resource accounting are separate. Resource accounting is done by a central module, the Binder, which monitors the resource blocks (RBs) in the system on a TTI basis. The Binder is thus an oracle module, which has the full visibility of all the nodes in the system, and can be called by every node to obtain shared information. More in detail, it keeps track of which RBs are used by which node (eNBs in DL and UEs in UL), which still allows a correct interference management. The data flow is instead modeled via message exchanges between modules. The correspondence between messages and resources is maintained by the Binder, which associates a certain amount of RBs to each message, based on the length of the MAC PDU and on the modulation and coding scheme employed by the transmitter.

Control channels, such as the Physical Downlink Control Channel (PDCCH), that is used to carry scheduling assignments, are not directly implemented, rather they are modeled using separated messages and the associated resources are again kept consistent by the Binder. This allows us to simulate and evaluate their behavior from a resource perspective, without paying for the added complexity of a full emulation.

Finally, each NIC card has a *ChannelModel* class, which cooperates with the PHY layer and that models the status of the air channel as is perceived by the NIC card itself. It is mainly used to compute the SINR of a signal received by the node, which in turn is used by the PHY layer to compute the CQIs and evaluate

transmission errors. Each *ChannelModel* interacts also with the Binder to know exactly which resources are actually used by every node in the system. This allows us to evaluate SINRs, and consequently CQIs and transmission errors, on a per-RB basis. SimuLTE defines the *ChannelModel* as an interface, i.e. a C++ abstract class with pure virtual functions only, and also provides an implementation of a realistic model, which accounts for path-loss, fading and shadowing. However, such interface can be easily extended by implementing the two functions *getSINR()* and *error()*, used for the above functions.

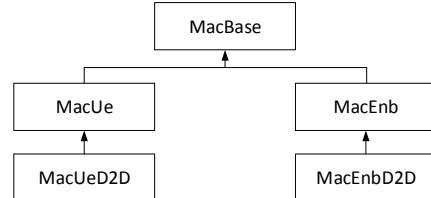


Fig. 4. Example of inheritance

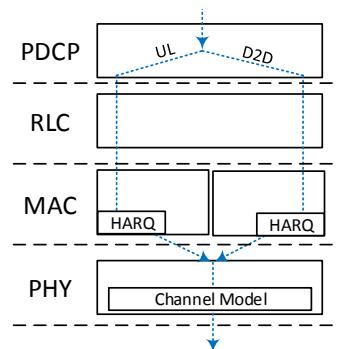


Fig. 5. Data flow from the sender UE perspective

IV. MODELING D2D

This section details how the D2D support is blended into the SimuLTE architecture. This means that most of the D2D operations are realized using pre-existing SimuLTE functions, leveraging inheritance and modularity. D2D-specific functionalities at each layer of the LTE stack are introduced by extending the corresponding modules. For example, with reference to Fig. 4, *MacUeD2D* and *MacEnbD2D* classes inherit the structure of *MacUe* and *MacEnb* classes, respectively. We will describe the modifications that have been implemented to the protocol layers of both UEs and eNBs.

The Binder stores peering relationships in a data structure that contains the set of directly reachable destinations, for each D2D-capable UE. Moreover, we define a new *flow direction*, i.e. D2D, besides the existing DL/UL ones.

From the UE perspective, IP datagrams reach the PDCP layer and the corresponding flow can be associated to either the D2D or the UL directions, depending on whether the destination is in the peering table in the Binder or not. Note that the UL direction is always feasible, whereas the D2D may or may not be, and this may change over time (e.g., due to mobility) and entail *mode switching* [6].

Then, the PDCP layer assigns a Logical Connection Identifier (LCID) to the incoming data flow, according to a 5-tuple defined by source/destination IP address/port, and flow direction. This creates different LCIDs for flows having different transmission directions, allowing lower layers to distinguish UL and D2D flows. Fig. 5 shows the data flow, where datagrams coming from the IP layer are forked to either one of the two branches at the PDCP layer and treated accordingly at lower layers.

As far as the RLC layer is concerned, no additional functionalities are required, thus common operations are performed. At the MAC layer, the UE must notify the presence of new data for a D2D flow by sending a BSR to the eNB. Since the latter needs to know whether the BSR refers to UL or D2D traffic, the BSR comes with a specific LCID. This allows the eNB to issue the right grant for UL or D2D transmission. When the UE receives a D2D grant, it builds a MAC PDU according to the SL CQI, as explained later on, and handles it to the PHY layer. Since each LTE node is identified by a MAC ID, the MAC PDU contains the MAC ID of the destination UE, instead of the eNB's MAC ID, as an UL PDU would. Airtime transmission is simulated by sending direct OMNeT messages to the PHY module of the node identified by the MAC ID in the MAC PDU.

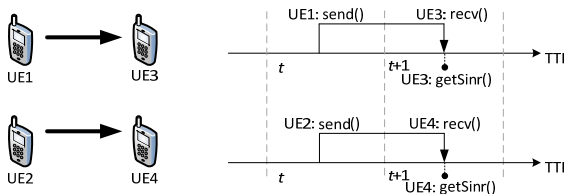


Fig. 6. Transmission and reception of direct messages

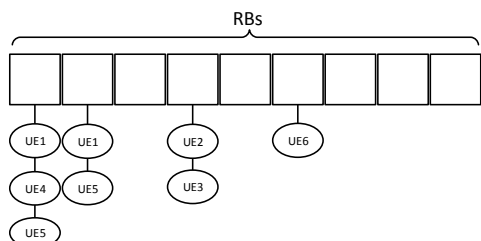


Fig. 7. Representation of RBs with frequency reuse

At the receiver side, in order to compute the interference correctly, procedures for both transmission decoding and CQI reporting need to be modified. Let us consider the sequence of OMNeT events related to a PHY transmission. With reference to Fig. 6, UE1 and UE2 transmit data to their peers at TTI t via direct OMNeT messages. At TTI $t+1$, UE3 and UE4 receive the corresponding OMNeT message and compute if the latter has been received successfully, via the `getSinr()` function. To do this, UE3 (UE4) needs to know who else is transmitting at TTI t on the same RBs, in this case UE2 (UE1).

On the other hand, a CQI measurement occurring at TTI $t+1$, is based on the status of the channel at the very same TTI. Thus, it is necessary for the UE to store the map of used RBs and keep it for at least two TTIs. This way, it is possible to check the RB occupancy status for each UE during both the current and previ-

ous TTI, for both CQI reporting and decoding purposes respectively. The `getSinr()` function implemented by the channel model of the UEs is extended such that, for each RB associated with the transmission from UE i to UE j , the SINR is

$$SINR(i, j) = \frac{P_{rx}^{ij}}{N + \sum_{k \neq i} P_{rx}^{kj}},$$

where P_{rx}^{ij} is the power received from UE i , P_{rx}^{kj} is the power received from the k -th interfering UE and N is the Gaussian noise. The interference term at the denominator is computed by checking which UEs (besides eNBs) used the same RB at the time of the transmission. To do this, we scan the list of all (D2D-capable) UEs in the system and check whether the UE used that RB during the current TTI (for CQI measurement) or the previous one (for decoding of transmissions). The UE list is maintained at the Binder and is built during the initialization phase of each UE, which stores the pointer to the corresponding PHY layer module into the list.

Once a transmission has been performed, an H-ARQ feedback (ACK/NACK) must be sent. This is modeled through a control message flowing from the receiving UE to the sender UE. However, the eNB must have knowledge of that feedback in order to schedule possible retransmissions: recall that the UL H-ARQ is synchronous, hence if the D2D transmission fails, the eNB must re-schedule it eight TTI later. To this aim, we enhance the eNB with a data structure that mirrors the *status* of each D2D H-ARQ buffer on the sender (i.e., whether idle or waiting for retransmission). This data structure is updated on each TTI via direct method call, which simulates the fact that the eNB is able to overhear the H-ARQ *feedback* sent by the receivers of D2D flows. Note that this does not entail that the eNB overhears the *data transmission* on the SL (which is false), and it is however a necessary condition for network-controlled resource allocation under the current H-ARQ standard.

From the eNB's point of view, the main MAC-level operation is resource allocation, which is enhanced to enable the scheduling of both UL and D2D flows in the same TTI. To do this, the eNB exploits i) the knowledge of H-ARQ buffers of D2D UEs to schedule retransmissions, and ii) the CQIs reported by the UEs for the D2D links. Moreover, the scheduling function of SimuLTE is overloaded, so that simultaneous allocation of the same RB to multiple D2D flows is allowed. This is accomplished by exploiting a data structure containing the set of UEs allocated for each RB, as exemplified in Fig. 7. Based on the information contained in that structure, the eNB issues transmission grants to the UEs. The algorithm according to which the data structure of Fig. 7 is user-defined. For instance, paper [3] defines one where interference is modeled via a binary *protocol model*, i.e. flows are either interfering or not, and only non-interfering flows can share the same RBs.

V. VALIDATION AND PERFORMANCE EVALUATION

In this section, we validate our model and carry out simulations to assess the effects of interference in D2D communications. We consider a system with one eNB and a single pair of

TABLE I - MAIN SIMULATION PARAMETERS

| Parameter | Value |
|-------------------|---------------------|
| Carrier frequency | 2 GHz |
| Bandwidth | 5 MHz (25 RBs) |
| Path loss model | ITU Urban Macro [7] |
| Fading model | Jakes |
| UE Tx Power | 26 dB |
| Noise figure | 5 dB |
| Cable loss | 2 dB |
| Simulation time | 50 s |

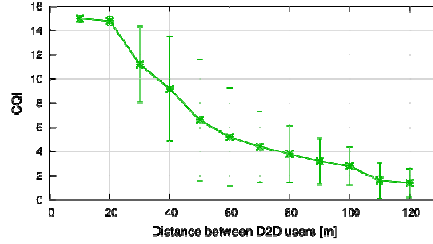


Fig. 8. Reported CQI vs distance

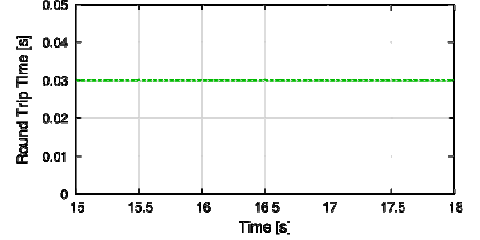


Fig. 9. Round Trip Time of the TCP connection

UEs communicating using D2D, placed at an increasing distance to each other. UEs transmit with a power of 26 dB and the channel is affected by fading. Main simulation parameters are summarized in Table I. Fig. 8 shows the reported CQI for the D2D link. Channel quality is good at short distance (less than 30m), then it rapidly decreases until D2D communication becomes unadvisable, at distance greater than 100-120m. In that case, it would be beneficial to switch the transmission to the traditional two-hop path through the eNB.

As a further validation, we now consider the UEs at a fixed distance of 20m and have them establish a TCP connection. Fig. 9 shows the Round-trip Time, which stays constant at 30ms. This value is the one resulting from the handshake reported in Fig. 10. At $t=0$, the sender UE has data ready for transmission and issues a RAC request to the eNB. The latter replies with an UL grant of one RB that the UE exploits to send the BSR. Since the BSR uses resources from the data plane, the eNB requires three TTIs to decode it. At $t=10$, it sends a SL grant to the UE, which finally transmits data to its peer at $t=11$. After decoding, the receiving UE delivers the data at the TCP layer, which generates the corresponding ACK, at $t=15$. Transmission of the ACK follows the same sequence, thus it requires another 15ms to get back to the sender TCP layer. Note that in SimuLTE decoding is only required for data plane transmissions.

To evaluate the impact of interference on resource allocation, we setup a scenario with one eNB and a varying number of UE pairs, whose endpoints establish a D2D flow. The simulation scenario is reported in Fig. 11. UEs are randomly deployed at a maximum distance of 100m from the eNB and no further than 25m from their respective peering UE. Each transmitting UE sends 100-byte packets each 20ms to its peer, resulting in a 40 kbps-Constant Bit Rate (CBR) traffic. The CQI reporting period is set to 10ms. We assess the impact of the interference on both computational complexity and system performance. We do this by comparing two allocation schemes: the well-known MaxC/I scheduler, which schedules D2D connections according to their reported CQIs, and allocates *different* RBs to different UEs (i.e., does not exploit frequency reuse), and the *Best-Fit* scheduler described in [3]. In the latter, frequency reuse among D2D flows is allowed, unless the interference produced is higher than a certain threshold. More in detail, the eNB knows the position and power of the UEs, and uses that information to build a *conflict graph*, whose nodes are D2D flows and whose edges are *conflicts*, i.e. pairs of flows that cannot be scheduled on the same RBs (i.e. when the received power from the interfering

flow is higher than a threshold of -50dB). Then, resource allocation is done by ranking D2D flows by decreasing CQI, and allocating each one of them on RBs where no conflict exists, using a *best-fit* approach. UL flows cannot share RBs with D2D ones.

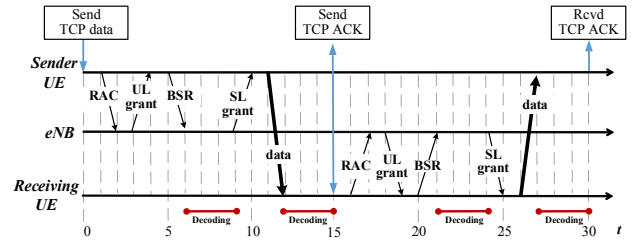


Fig. 10. Analysis of Round Trip Time

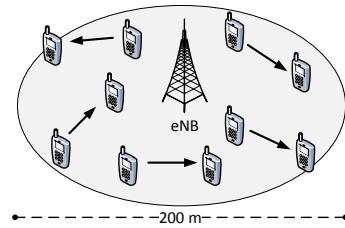


Fig. 11. Evaluation Scenario

Fig. 12 shows the execution time of the simulation. These values are obtained by running the system on an Intel(R) Core(TM) i7 CPU at 2.80 GHz, with 8 GB of RAM, a Linux Kubuntu 12.04 operating system, and OMNeT++ version 4.6. As expected, computing interference requires additional CPU time, which depends on the number of flows in the network. Recall that evaluating the received SINR in the presence of interference requires cycling through the list of all UEs to check whether they transmitted in the same RBs.

Fig. 13 shows the average UE throughput at the application level, whereas Fig. 14 reports the number of allocated RBs in the UL subframe. We observe that interference does cause throughput degradation, which is however limited to less than 10%, indicating that a protocol model with the above settings is effective in limiting the interference. However, on the other hand, frequency reuse allows the eNB to allocate *fewer* RBs (about 20% less), leaving more space to accommodate additional traffic, i.e. increasing the overall efficiency of the system.

In order to verify how interference affects the performance of an algorithm that resolves conflicts using the protocol model, we simulate the same scenario as above with heavier traffic load, i.e. UEs send CBR traffic at 400 kbps. We use the best-fit allocation with different settings of the conflict threshold. Clear-

ly, frequency reuse increases with the threshold values. Fig. 15 shows the overall sum of UE throughput at the application level. As expected, the cell reaches saturation faster with low threshold values, since the same amount of traffic requires more RBs. However, lower frequency reuse also implies lower interference. Thus, at low loads (40-50 flows), a threshold of -70dB warrants better performance than one of -90dB.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we described the modeling of unicast D2D communications within SimuLTE. Since SimuLTE is implemented following the modular paradigm of OMNeT++, adding unicast D2D communications requires few, localized changes to the software architecture, mainly consisting in specializing new classes through inheritance. We have validated the model by showing that message exchange sequences and CQI reporting match the expectations. Moreover, we have assessed the impact of frequency reuse on the system throughput and efficiency.

SimuLTE is being extended at the time of writing. Among the features being considered for a future release we can list: support for *multicast* D2D operations; support for dynamic mode switching, as described in [6]; an easy-to-use software interface to optimization solvers (e.g., CPLEX), so that resource allocation problems can be formulated as optimization problems and solved by calling the solver API; making SimuLTE D2D support interoperable with the VEINS vehicular network simulator [16], to test D2D communications in a vehicular scenario.

REFERENCES

- [1] Varga, A. and Hornig, R. (2008), "An overview of the OMNeT++ simulation environment", in Proc. SIMUTools '08, Marseille, France, March 2008.
- [2] A. Viridis, G. Stea, G. Nardini, "Simulating LTE/LTE-Advanced Networks with SimuLTE", DOI 10.1007/978-3-319-26470-7_5, in: Advances in Intelligent Systems and Computing, Vol 402, pp. 83-105, Springer, 15 January 2016
- [3] G. Nardini, G. Stea, A. Viridis, D. Sabella, M. Caretti, "Resource allocation for network-controlled device-to-device communications in LTE-Advanced", Wireless Networks, 2017
- [4] J. Gu, H. Yoon, J. Lee, S. J. Bae, M. Chung, "A resource allocation scheme for device-to-device communications using LTE-A uplink resources", Pervasive and Mobile Computing, Volume 18, April 2015, Pages 104-117, ISSN 1574-1192,
- [5] A. Asadi, Qing Wang and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1801-1819, Fourthquarter 2014.
- [6] G. Nardini, G. Stea, A. Viridis, D. Sabella, M. Caretti, "Fast and agile lossless mode switching for D2D communications in LTE-Advanced networks", IEEE VTC Spring 2016, Nanjing, PRC, 15-18 May 2016
- [7] G. Nardini, G. Stea, A. Viridis, "Performance evaluation of TCP-based traffic over direct communications in LTE-Advanced", IEEE VTC Spring 2016, Nanjing, PRC, 15-18 May 2016
- [8] C. Vallati, A. Viridis, E. Mingozzi, G. Stea, "Exploiting LTE D2D Communications in M2M Fog Platforms: Deployment and Practical Issues", ResearchGate.net proc. of IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14-16 Dec. 2015, pp. 585-590
- [9] Yi Shi, Y. T. Hou, Jia Liu, S. Kompella, "Bridging the Gap between Protocol and Physical Models for Wireless Networks," in IEEE Trans. on Mobile Computing, vol. 12, no. 7, pp. 1404-1416, July 2013.
- [10] 3GPP TR 36.814 v9.0.0, "Further advancements for E-UTRA physical layer aspects (Release 9)," March 2010.
- [11] 3GPP - TS 36.843 v12.0.1, "Study on LTE Device-to-device Proximity Services: Radio aspects (Release 12)", March 2014

- [12] X. Lin, J. Andrews, A. Ghosh, R. Ratasuk, "An overview of 3GPP device-to-device proximity services," IEEE Communications Magazine, vol.52, no.4, pp.40,48, April 2014, doi: 10.1109/MCOM.2014.6807945
- [13] K. Doppler, Chia-Hao Yu, C.B. Ribeiro, P. Janis, "Mode Selection for Device-To-Device Communication Underlying an LTE-Advanced Network," Proc. of WCNC 2010, pp.1-6, 18-21 April 2010
- [14] J. Gu, S.J. Bae, B.G. Choi, M.Y. Chung, "Mode Selection Scheme Considering Transmission Power for Improving Performance of Device-to-Device Communication in Cellular Networks", Proc. ICUIMC 2012, 20-22 February 2012, Kuala Lumpur, Malaysia
- [15] G. Piro *et al.* "D2D in LTE vehicular networking: system model and upper bound performance", ICUMT 2015, October 2015
- [16] C. Sommer, R. German, F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," IEEE Trans. on Mobile Computing, vol.10, no.1, pp.3-15, Jan. 2011.
- [17] SimuLTE website: <http://simulte.com>

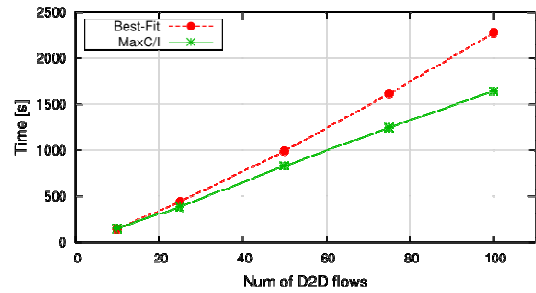


Fig. 12. Execution time w and w/o frequency reuse

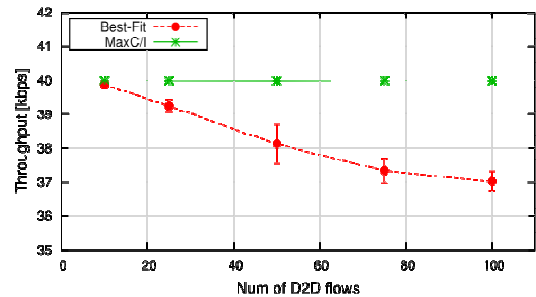


Fig. 13. UE throughput w and w/o frequency reuse

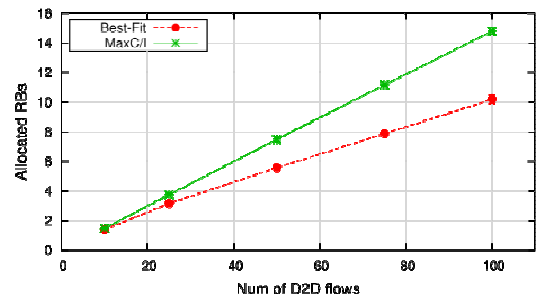


Fig. 14. Average Resource Blocks allocated with and without frequency reuse

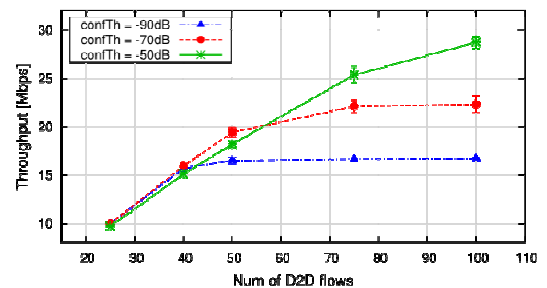


Fig. 15. Overall sum of app.-level UE throughput vs. conflict thresholds