This paper appeared at IEEE VTC Spring 2016.

Fast and agile lossless mode switching for D2D communications in LTE-Advanced networks

Giovanni Nardini(1), Giovanni Stea(1), Antonio Virdis(1), Dario Sabella(2), Marco Caretti(2)

(1)Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy
g.nardini@ing.unipi.it, giovanni.stea@unipi.it, a.virdis@iet.unipi.it

(2) Telecom Italia Lab, Turin, Italy
{dario.sabella, marco.caretti}@telecomitalia.it

Abstract— **Direct (or D2D) communications allow two UEs to communicate without passing through the eNodeB. However, the two UEs may still need to relay their communication through the eNB from time to time, hence should be able to switch from the direct to the relayed mode seamlessly, without this affecting the QoS. In this paper we show that in conventional systems a mode switching may cause relevant losses, and propose two architectures to mitigate or solve this problem. Our proposals do not require extra signaling or additional functionalities to be added to the network, hence are scalable and inexpensive. We assess their effectiveness through detailed system-level simulations.**

Index Terms—**LTE-A, device-to-device, mode switching**

I. INTRODUCTION

Network-controlled direct (or device-to-device, D2D) communications are currently being investigated and standardized in the framework of LTE-Advanced, and are envisaged as being part of the upcoming 5G systems. Enabling devices to communicate directly, without using the classical two-hop *infrastructure* path having the eNodeB (eNB) as a relay, is expected to reduce latency, enable frequency reuse on a spatial basis, and – possibly – reduce energy consumption at the eNB itself. Typical use cases are high data rate services where the endpoints are in range for direct communications, like file sharing, gaming and social networking [18].

Both one-to-many (i.e., proximity broadcast or multicast) and one-to-one D2D communications are being actively studied by the research community. In one-to-one communications, the two endpoints – being mobile – may not remain in hearing range of each other for the entire duration of the communication. Even if they do, the infrastructure path may still allow higher data rates, or the eNB may simply decide *not* to use the direct path at some point to optimize frequency reuse on a cell-wise scale. For this reason, it is necessary to envisage fast and agile *mode switching* procedures that allow two communicating devices to switch from the direct path, or *sidelink* to the infrastructure path and back, without disrupting the communication or the QoS.

This paper shows that – unless proper countermeasures are taken – mode switching may impair one-to-one D2D communications, inducing a relevant amount of losses. This is because a single hop in LTE-A (both the direct one and either leg of the infrastructure path) requires a PDCP peering, with associated state (e.g., PDU numbering) and ciphering. When switching mode, all traffic below the PDCP layer is unable to reach their destination. Few works so far have addressed the problem of how mode switching takes place [9]-[11]: the proposed solutions often rely on *additional signaling message exchange* between terminals and eNB, which makes them slow and poorly scalable, and do not guarantee that traffic buffered at the RLC is transmitted. We propose two architectural solutions to address the mode-switch problem, which are exempt from the above-mentioned drawbacks: the first one – called *local solution* – only involves the sender, and *mitigates* switching-induced losses without eliminating them completely; the second one – called the *RLC-tunneling* – requires modifications at the eNB *and* the receiver, but it avoids losses completely. We compare the two solutions as for performance, overhead, viability and standard compliance.

The rest of the paper is organized as follows: Section II reports background and hypotheses and presents the problem, while Section III describes the related work. The proposed solutions are described in Section IV, and Section V reports simulation results. We conclude the paper in Section VI.

II. BACKGROUND AND SYSTEM MODEL

Hereafter we provide a minimal background on the LTE-A protocol layering and introduce our working hypotheses.

As shown in Figure 1, IP packets traverse the Packet Data Convergence Protocol (PDCP), where they are ciphered and numbered to form PDCP PDUs. These are immediately sent down to the Radio Link Control (RLC) in the form of RLC SDUs, which are kept in the *RLC buffer*. Each flow has associated one PDCP entity and one RLC entity. Three RLC modes are possible, namely *transparent* (TM), *unacknowledged* (UM) and *acknowledged* (AM). UM - recommended by the standard for D2D communications [12] - performs *segmentation* / *concatenation* of RLC SDUs on transmission, and *reassembly,* duplicate detection and reordering of RLC PDUs on reception. The MAC requests to the RLC an RLC PDU of a given size, and the RLC responds by dequeuing from the RLC buffer an appropriate number of RLC SDUs, fragmenting and concatenating them as necessary into RLC PDUs. The MAC adds a header to form a MAC PDU, also called Transmission Block (TB). MAC-layer transmissions are arranged in subframes and paced at Transmission Time Intervals (TTIs) of 1ms. In the downlink (DL), the eNB allocates a vector of *Resource Blocks* (RBs) to transmissions directed to the User Equipments (UEs) associated to it on each TTI. In the uplink (UL), the eNB issues *transmission grants* for each UE, specifying which RBs they can use, using what transmission format. MAC-level error recovery is provided by a Hybrid ARQ (H-ARQ) scheme, which allows a configurable number of retransmissions.

We consider a cell, served by an eNB, and two D2D-capable UEs in the coverage area of the cell. We assume that the UEs are (electromagnetically) near enough for direct communication to take place, representing the endpoints of a D2D communication. These UEs can communicate either directly, i.e., in *D2D mode* (*DM*), or using the eNB as a relay, i.e., in *Infrastructure mode* (*IM*). In the following, we refer to UE a and UE b as the transmitter and the receiver of the communication flow, respectively. Since we focus on network-controlled D2D, UEs still exchange control information with the eNB even when using DM.

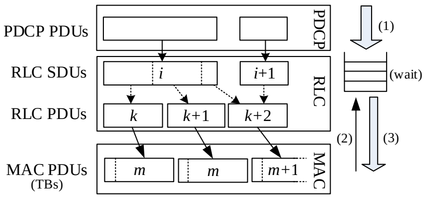

Figure 1 - User data flow through the LTE protocol layers.

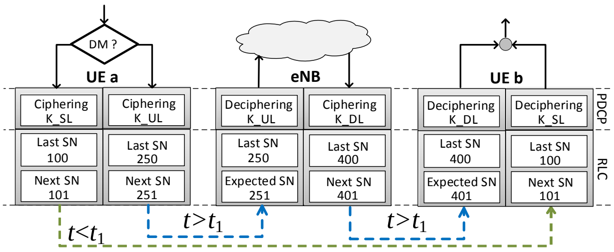

Figure 2 - Effects of mode switch on D2D communications.

We consider a Frequency Division Duplexing (FDD) system, where DM transmissions take place in the UL subframe. This is a common assumption [2], since the UL subframe is likely to be the least loaded one (due to the well-known traffic asymmetry) and allows better overall SINR, especially when both endpoints are far from the eNB. Accordingly, we assume that UEs are equipped with a Single Carrier-Frequency Division Multiple Access (SC-FDMA) receiver [3]. As far as H-ARQ is concerned, we assume that the feedback is sent by the receiver to *both* the transmitter and the eNB. This is necessary: in fact the former needs to know if retransmission is required (in eight TTI, it being UL), while the latter allocates RBs for it to take place.

Due to UEs' mobility and to changes in the environment, an eNB should be able to select *dynamically* whether a D2D communication occurs in DM or IM, according to some metric (e.g., best channel quality). However, this may generate losses, unless some countermeasures are taken. Assume that UE a is communicating with b in DM. Then, a and b have a PDCP peering session established, with its sequence numbers and ciphering. When the communication is switched to IM, *two different* PDCP peerings must be used: one between a and the eNB (i.e., the UL leg), another between the eNB and b (the DL leg). Those two PDCP sessions are independent, hence have different, unrelated sequence numbers and sets of keys for ciphering/deciphering. Since different PDCP entities come with different RLCs, the RLC PDU sequence numbers of the SL path cannot be assumed to be valid on the UL/DL leg of the IM path either. With reference Figure 2, a and b are the endpoints of a DM flow. At time t_1, a's RLC expects to send down a PDU whose sequence number is 101. However, if the mode is switched to IM at t_1, then a will peer with the eNB's PDCP/RLC entities, which will expect a different next RLC sequence number (251, in this example), as well as traffic ciphered with different keys. Because of this, all data in the RLC buffer of the "old" DM connection cannot be sent on the "new" peering, and can only be discarded. Moreover, *fragments* of RLC SDUs that have been already received at b at the time of the mode switch will be discarded as well (when b's RLC reassembly timeout expires), since their missing counterparts, sitting in a's "old" RLC buffer, will be discarded. The same problem occurs, obviously, also when switching from IM to DM, where it is exacerbated by the fact that losses can occur independently on the UL and DL leg of the IM path. We also observe that using a different RLC – notably AM, despite the standard [12] has not identified the need of using AM RLC for D2D communications – would not solve the problem. In fact, although AM allows a sender to know which RLC PDUs have/have not been received at the peer entity, when switching from IM to DM there is no way for a to know what has got to b, since b – being two hops away – is not peering with a at that moment. These losses may be significant and impair the QoS of the flow, whether a multimedia or a TCP-based one, thus their occurrence should be minimized.

III. RELATED WORK

Although mode selection *algorithms* for one-to-one D2D communications have been the subject of some previous works (e.g., [4]-[8], some of which advocate *dynamic* mode selection), to the best of our knowledge very few works so far – significantly, only patents – have dealt with the *protocol requirements* to enable mode switching. Solutions [9] and [10] assume that the PDCP buffers PDUs, and, at a mode switch, the sender, receiver and eNB exchange signaling information to agree on which PDCP PDU number should be transmitted next on the new path. The data plane may be halted while the above signaling occurs. Additional per-flow signaling poses speed and scalability problems, since mode switch may affect several flows simultaneously (e.g., for periodic cell optimizations), at relatively fast timescales. Moreover, putting flows on hold is likely to generate deadline misses for playback multimedia flows and timeouts of TCP connections. Last, adding buffering at the PDCP implies that incoming data do not reach the RLC buffer immediately, hence are not made available to the MAC, and leaves open the problem of defining a handshake with the RLC buffer.

Work [11] proposes tunneling *at the PDCP level* when the D2D connection traverses the IM path: two one-hop "outer"

PDCP connections, one for the UL and one for the DL, act as tunnels for the end-to-end D2D one. PDCP-level tunneling requires that the destination perform deciphering twice.

None of the above works mentions what happens to traffic sitting in the "old" RLC buffer at a mode switch: since it cannot be transmitted on the new path, it can only be discarded, hence the problem of losses still stands unsolved.

IV. MODE SWITCHING

Hereafter, we discuss two solutions to tackle the problem of mode switching losses. The first one, called *local* solution, uses additional data structures *at the sender side only* to retransmit possibly unreceived data. The second one, *RLC tunneling*, relies on the eNB to act as a *relay at the RLC layer* for a flow whose PDCP entities are located *only* on the terminals.

A. *Local solution*

This solution relies on putting a buffer on top of each PDCP entity to hold PDCP *Service* Data Units (PDCP SDUs, e.g. IP packets) *before* they are ciphered. A copy of each PDCP SDU enters the buffer when received from the upper layer, whereas the original PDCP SDU is sent down for processing and transmission in parallel. While fragments of that PDCP SDU traverse lower layers at the sender, we keep trace of which RLC PDUs, and ultimately which MAC-layer TBs, these fragments get included into. The *copy* PDCP SDU is removed from the buffer *only* when all its fragments have been acknowledged at the MAC layer by the receiver. This requires propagating MAC-layer ACKs up to the RLC and the PDCP, and keeping a local map of PDCP SDU/MAC TB associations. At a mode switch, the PDCP SDUs still in the buffer are those that *may* have not been entirely received. Moreover, those SDUs have neither been ciphered nor numbered yet, hence can be seamlessly transferred to the PDCP entity associated to the new mode, which will process – i.e., cipher and number – them anew. The contents of the "old" RLC buffer are instead discarded. Figure 3 shows an example of this solution. The sender UE is in DM and has five SDUs in the RLC's TX buffer and their corresponding PDCP SDUs in the copy buffer. At time $t < t_1$, RLC SDUs 1,2,3 are included into RLC PDUs 101,102 and transmitted using TBs 201,202. Thus, the corresponding PDCP SDUs are removed from the copy buffer. The mode is switched at t_1, thus the remaining SDUs in the copy buffer (4 and 5) are moved to the IM PDCP entity, while the RLC's TX buffer is cleared. PDCP SDUs are numbered according to the new PDCP entity.

The main advantages of this solution are that it only requires (few) modifications to the *sender*, localized and backward-compatible, it does not require additional control messages, and it *guarantees* that *all IP packets are entirely transmitted* by the sender, even across mode switches and despite fragmentation, unlike [9]-[11]. However, it also has some shortcomings:

a) When switching from IM to DM, this solution only allows one to know *what got to the eNB*. There is no way to know if it made it to the receiver as well, nor can there be without additional end-to-end signaling.

b) IP packets may not be delivered in sequence, especially after an IM-DM switch. In fact the above solution requires that the receiver keeps *two distinct PDCP peerings*, one for DM and one for IM, related to the same IP flow. Now, in-sequence delivery is guaranteed only *within* a PDCP connection, but not *among* different ones. Thus, the IP packets sent through the "new", faster DM path may end up overrunning those already sent in the UL of the slower IM path.

c) Due to H-ARQ latencies, it may introduce duplicates, which waste airtime resources and can be harmful to TCP flows. In fact, it may well happen that a DM-IM mode switch occurs *after* an IP packet has been successfully received, but *before* the MAC-layer ACK of the last fragment has made it back to the sender (it takes 4ms for a MAC TB to be ACKed [17]). That IP packet will still be in the copy buffer, hence will be transferred to the "new" PDCP entity and re-sent. Thus, the same IP packet can appear twice at the receiver.

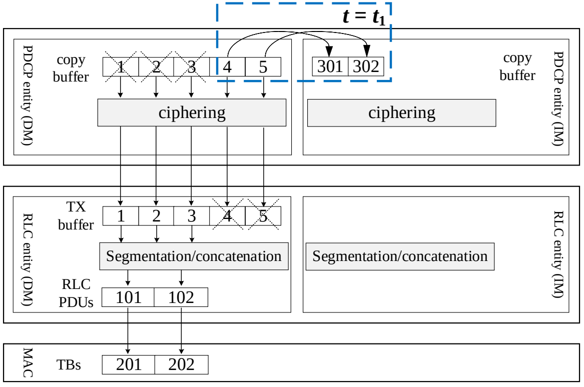

Figure 3 - Local solution.

B. *RLC tunneling*

We argue that the main obstacle to an agile mode switch is that *different PDCP peerings are involved in the DM and IM communications*. In fact, as already explained, what is buffered *at and below* the PDCP level cannot be transferred from one peering to the other, and is therefore lost.

The problem can be circumvented if D2D flows are i) started in DM, and ii) only relayed *at the RLC level* when switched to IM. This way, the only PDCP peering is between the two endpoints, and the eNB just relays RLC PDUs from the former to the latter, as shown in Figure 4. At the sender side, all data from the upper layers traverse the same PDCP entity and are buffered at the same RLC TX entity, regardless of the current communication mode. Therefore, a single sequence numbering and ciphering is maintained. When the sender UE receives a grant from the eNB, its RLC entity delivers a RLC PDU to the MAC layer. The MAC layer takes care of sending the MAC PDU to either the receiver UE or the eNB (respectively, dashed and solid line in Figure 4), depending on the current mode. This is transparent to the sender's RLC, which only has to send down a PDU of the requested size. If the transmission is DM, no changes are required. For IM to work, the eNB needs to know that the received MAC PDU contains a RLC PDU that has to be relayed at RLC, instead of delivered to the PDCP. To this aim, we can exploit one of the *reserved* bits of the MAC header. When that bit is set, the MAC delivers the RLC PDU to a new type of RLC entity, which we call *RLC relay entity*, shown in Figure 4. When the latter receives an RLC PDU from the MAC (i.e., on the UL leg of the IM path), it stores it in a DL TX buffer, *without* performing reordering and/or reassembly, which will be done at the receiving UE only. In other words, it interprets the UL RLC PDU as a DL RLC SDU, which can be segmented and/or concatenated with other PDUs before transmission in the DL leg, i.e. as if it were a chunk of data coming from the PDCP. The UL RLC PDU is encapsulated into a DL RLC TUNNEL PDU for DL transmission. The receiving UE reassembles the RLC TUNNEL SDUs received from the eNB. The reassembled SDUs are in turn the RLC PDUs transmitted by the sender UE (and tunneled through the eNB). Thus, these PDUs are sent to the RLC entity that is responsible for managing the peering with the sender UE, i.e. the same used for DM.

This solution has several advantages: first of all, since it operates at the lowest layer of the LTE stack where buffering occurs (i.e., the RLC), it *cannot* lose PDUs as the others do. Second, it does not require additional signaling, nor halting of the connections at mode switch. Third, its overhead is limited to a 2-byte extra RLC header in the DL leg only, and it relieves the eNB of the cumbersome tasks of PDCP ciphering/deciphering. Fourth, unlike [9]-[11] and the *local* solution, it avoids RLC-level reassembly at the eNB. In fact, fragments of the same PDCP PDU (i.e., RLC PDUs) get relayed to the receiver, which is the *only* entity where RLC reassembly occurs. This minimizes the end-to-end latency. The only drawback is that it requires modifications to the eNB and receiver. However, these do not require *new* functionalities, but exploit existing ones for a new purpose.

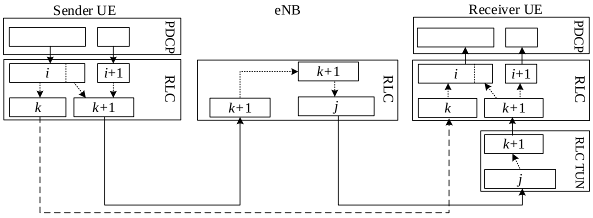

Figure 4 - RLC tunneling.

V. PERFORMANCE EVALUATION

In this section, we show the performance of the architectural solutions presented in Section IV. Our evaluation is carried out using SimuLTE [13]-[14], where we implemented one-to-one direct communications. Figure 5 reports the simulation scenario. We consider one pair of D2D-capable UEs and one eNB, whose antenna radiates the signal with an omnidirectional pattern. At the beginning of the simulation, the two UEs are 300m away from the eNB. They swing back and forth in a straight line at a speed of 3m/s, their distance varying between 30m and 160m. Such a path allows the UEs to experience the whole range of CQIs on the SL, whereas the UL CQI stays equal to nine. The eNB periodically (once per second) selects the mode with the highest CQI, hence the communication will periodically bounce between DM and IM. To highlight the effects of mode switch, we disabled fading and inter-cell interference, so that each observed phenomenon (e.g., a packet loss) can be ascribed to mode switching *only*. Simulation parameters are reported in Table 1.

UE1 sends a 1GB-file to UE2 using TCP as a transport protocol, and the transfer lasts for the whole simulation time. Note that TCP is sensitive to losses, which reduce its congestion window (CW) and the flow throughput. We compare our solutions with a baseline where the RLC buffer is cleared at the time of mode switch and no recovery mechanism (within the LTE stack) is used. Figure 6 shows the congestion window of the sender TCP, which allows one to grasp when losses occur. Markers on the *x*-axis represents the instants when the mode is switched. We observe that, with the baseline, the CW drops at *every* mode switch, whereas this happens less frequently with the *local* solution, and – significantly - only when switching from IM to DM. This is due to the fact that data buffered at the eNB are sent to UE2 (in the DL leg) simultaneously to (or after) new data sent by UE1 (using DM), hence are received out of sequence. Non-occasional out-of-sequence reception, in turn, triggers congestion avoidance mechanisms as do losses. With RLC tunneling, instead, losses are avoided and the CW increases with time, as it should. Recall that the effective sending rate is, in any case, bounded by the minimum between the CW and the *receiver* flow-control window, which is set to 64KB in our scenario.

Figures 7, 8, 9 describe in more detail the flow of TCP segments from UE1 to UE2 across an IM-DM mode switch, for the baseline, the local and the RLC tunneling solution respectively. The figures show a marker for each TCP segment sent by UE1, and a smaller one at the same quota to mark the instant when that segment is received by UE2. In Figure 7, we observe that a burst of losses occurs and the subsequent segments, received out-of-sequence, trigger a retransmission. Since the number of unacknowledged segments is larger than the sending window, UE1 cannot retransmit until a timeout expires. In other words, UE1 is stalled for as much as 1s. Figure 8 highlights that, with our local solution, *some* segments are received out-of-sequence after switching to DM. In fact, DM is faster than IM and new transmissions from UE1 reach UE2 *before* DL transmissions of data buffered at the eNB, as already mentioned. Again, unordered receptions generate duplicate ACKs, which in turn trigger the TCP congestion avoidance mechanism and retransmissions from UE1. However, now UE1 does not wait for the timeout to expire, as TCP segments are not lost. Figure 9 shows that the RLC tunneling solution makes the mode switch completely transparent to TCP. These results directly map to a difference in the application-level throughput, as shown in Figure 10. The local solution and the RLC tunneling achieve a throughput higher than that of the baseline, by 10% and 16% respectively.

VI. CONCLUSIONS

This paper presented two proposals to reduce the impact of mode-switching-induced losses in D2D communications. We showed that the problem arises from the fact that the standard mandates that buffering occurs at the RLC level, i.e. below the level at which connections are established and switched. Our

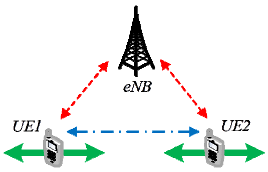

Figure 5 – Simulation scenario

TABLE 1 - SIMULATION PARAMETERS

Parameter	Value
Carrier frequency	2 GHz
Bandwidth	10 MHz (50 RBs)
Mobility model	Linear (OMNeT++ model)
UEs' speed	3 m/s
Path loss model	Urban Macro [16]
eNB Tx Power	40 dBm
UE Tx Power	20 dBm
eNB antenna gain	18 dB
Noise figure	5 dB
Cable loss	2 dB
Simulation time	500 s
# of replicas	10

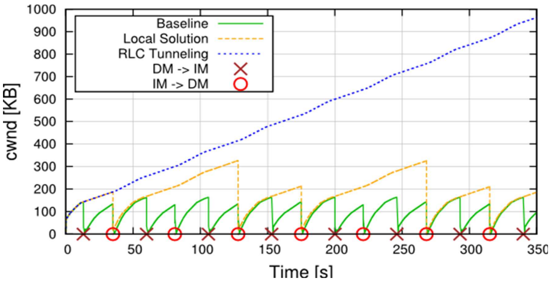

Figure 6 – TCP Congestion window

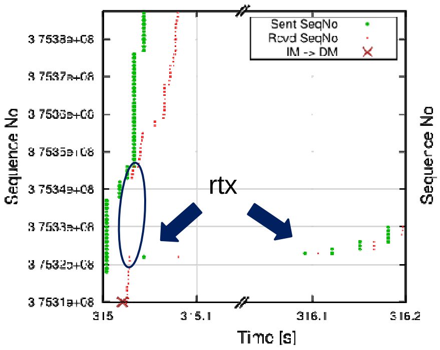

Figure 7 – Baseline

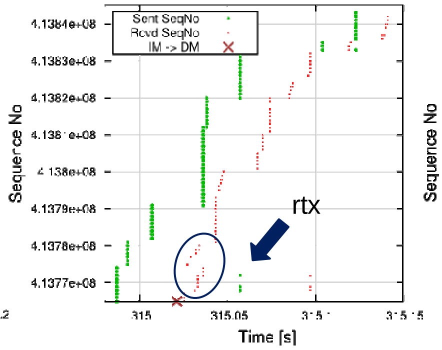

Figure 8 – Local Solution

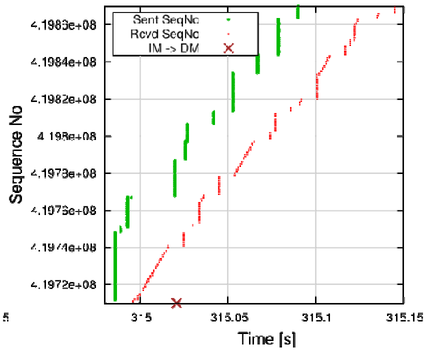

Figure 9 – RLC Tunneling

first *local* solution employs a copy buffer of PDCP SDUs at the sender and a map to track whether or not they have been entirely transmitted. Our second *RLC-tunneling* solution requires that relaying at the eNB only occurs at the RLC level, so that the RLC buffers do not have to be flushed at a mode switch. Both solutions improve the QoS, as testified by simulations involving TCP flows – which are known to be sensitive to losses.

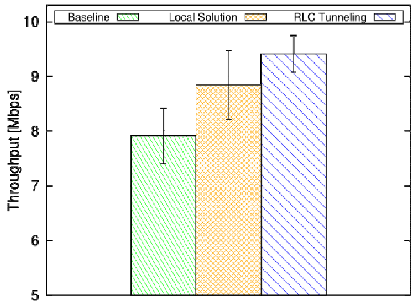

Figure 10 – App-layer throughput

VII. ACKNOWLEDGEMENTS

The subject matter of this paper includes description of results of a joint research project carried out by Telecom Italia and the University of Pisa. Telecom Italia reserves all proprietary rights in any process, procedure, algorithm, article of manufacture, or other result of said project herein described.

REFERENCES

[1] G. Fodor *et al.*, "Design aspects of network assisted device-to-device communications," IEEE Comm. Mag., 50(3), pp.170-177, 2012.

[2] C. H. Yu, O. Tirkkonen, K. Doppler, C. Ribeiro, "On the Performance of Device-to-Device Underlay Communication with Simple Power Control," Proc. of IEEE VTC Spring 2009, pp. 1-5, 26-29 Apr. 2009.

[3] X. Lin, J. Andrews, A. Ghosh, R. Ratasuk, "An overview of 3GPP device-to-device proximity services," IEEE Comm. Mag., 52:(4), pp.40-48, 2014

[4] K. Doppler, Chia-Hao Yu, C.B. Ribeiro, P. Janis, "Mode Selection for Device-To-Device Communication Underlaying an LTE-Advanced Network," Proc. of WCNC 2010, pp.1-6, 18-21 Apr. 2010.

[5] 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 Feb. 2012, Kuala Lumpur, Malaysia.

[6] M. Belleschi, G. Fodor, A. Abrardo, "Performance analysis of a distributed resource allocation scheme for D2D communications," IEEE GLOBECOM 2011 workshops, pp.358-362, 5-9 Dec. 2011.

[7] X. Xiao, X. Tao, J. Lu, "A QoS-Aware Power Optimization Scheme in OFDMA Systems with Integrated Device-to-Device (D2D) Communications," Proc. of VTC Fall 2011, pp.1-5, 5-8 Sept. 2011.

[8] S. Wen, X. Zhu, X. Zhang, D. Yang, "QoS-aware mode selection and resource allocation scheme for Device-to-Device (D2D) communication in cellular networks," Proc. of ICC 2013, pp.101-105, 9-13 June 2013.

[9] US8213360 B2, "Apparatus and method for flexible switching between device-to-device communication mode and cellular communication mode".

[10] US 2014/0160950, "Methods and apparatuses for facilitating D2D bearer switching".

[11] WO2014068464 A2, "Method and apparatus for packet tunneling".

[12] 3GPP - TS 36.843 v12.0.1, "Study on LTE Device to Device Proximity Services: Radio aspects (Release 12)", March 2014.

[13] SimuLTE webpage. http://www.simulte.com.

[14] A. Virdis, G. Stea, G. Nardini, "Simulating LTE/LTE-Advanced Networks with SimuLTE", Advances in Intelligent Systems and Computing, vol.402, pp.83-105, Springer, January 2016.

[15] OMNeT++, http://www.omnetpp.org

[16] 3GPP TR 36.814 v9.0.0, "Further advancements for E-UTRA physical layer aspects (Release 9)," March 2010.

[17] 3GPP TR 36.213 v12.5.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," March 2015.

[18] A. Asadi *et al.*, "A survey on device-to-device communication in cellular networks," IEEE Comm. Surveys and Tutorials 16:(4), pp.1801-1819, 2014.