

# Geofenced broadcasts via centralized scheduling of device-to-device communications in LTE-Advanced

Giovanni Nardini, Giovanni Stea and Antonio Viridis

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

**Abstract.** Point-to-multipoint device-to-device (P2MP D2D) communications have been standardized in LTE-Advanced (LTE-A) for proximity-based services, such as advertisement and public safety. They can be combined in a multi-hop fashion to achieve geofenced broadcasts in a fast and reliable way, over areas possibly covered by several cells [18]. This allows LTE-A networks to support critical services, like vehicular collision alerts or cyber-physical systems, at a modest cost in terms of consumed resources. In this paper, we argue that previous approaches, which rely on User Equipment (UE) applications to make distributed decisions about message relaying, incur in high per-hop overhead and make crossing cell border difficult. We then propose a novel approach that relies on centralized decisions made at the infrastructure eNodeBs (eNBs) to *schedule* unsolicited D2D grants to the optimal set of UEs that should forward a message at any time. The eNBs can also leverage inter-cell communications through the X2 interface to parallelize relaying over different cells, thus covering larger areas fast. We show that our infrastructure-based approach is computationally feasible and geographically scalable, and prove via simulation that it is faster, more reliable and efficient than UE-based multihop relaying.

**Keywords:** Broadcasting, algorithms, vehicular networks, LTE-Advanced, device-to-device, multihop, resource allocation.

## 1 Introduction

LTE-A already provides reliable ubiquitous connectivity in urban and rural environments. As such, it is the ideal support for location-based broadcast services, such as advertising, smart-city applications, and Distributed Hash Table (DHT) lookup requests in Internet-of-Things deployments [1-3]. Some of these services, for instance vehicular collision alerts, or augmented-reality live games, require low latency and high reliability, as well as the possibility to target an area defined by the application itself (e.g., a set of nearby roads) rather than relying on the layout of the cell coverage. Such an area, normally called a *geofence*, may include several (e.g., small) cells, or possibly exclude part of a (macro) cell.

The current LTE-A standard [4] does not support geofenced broadcasts. In fact, LTE's built-in Multicast/Broadcast SubFrame Network (MBSFN) mechanism, originally devised for broadcast multimedia, is *static*, hence unsuitable to this task: the message transmission format, the target area and the period of broadcast transmissions must all be selected statically. On the other hand, having the base station, called eNodeB (eNB) in the LTE terminology, relay user-defined messages to all the UEs in a target area using *unicast* downlink (DL) transmissions (one per targeted UE) would yield the maximum flexibility, at the price of occupying far too many DL resources.

Recently, device-to-device (D2D) transmissions have been proposed as the building block for broadcast communications [15]-[18]. With D2D, messages sent by a UE can reach physically proximate UEs without traversing the eNB in the standard uplink/downlink two-leg communication mode. D2D transmissions are still controlled by the eNB, which issues grants, either on demand to a requesting UE, or statically, as a pool of resources which UEs can draw from at the price of risking collisions. D2D communications allow the eNB to save power, since it cuts itself out from data-plane communications, and to reuse frequencies. However, the reach of a D2D transmission is limited by the UE transmission power, and well below what is required in the above-mentioned use cases. Multihop D2D transmissions have already been envisaged in recent works, to compensate for this shortcoming. Work [18], in particular, analyzes the time, reliability and overhead of using application-level relaying at single UEs to cover a target broadcast area. The conclusions are that scheduled D2D transmissions can support relatively fast, reliable and efficient broadcast services: for instance, they allow a 20-byte application message to reach 99% of the UEs within a broadcast radius of 1000m in 120ms, traversing five cells, and occupying about one allocated RB per UE on average. This is possible due to the increased efficiency of proximate, short-range transmissions, and the possibility of exploiting spatial frequency reuse. Work [18] exploits relaying *at the application level*, which has several drawbacks: first of all, individual UEs need to make the decision about whether or not to relay a message, and that decision takes time. In fact, for relaying to be efficient, a suppression mechanism (namely, the *Trickle* protocol [9]) must be used. Trickle blocks relaying when more than a threshold of copies of the same message reaches one destination within a predefined timeout: waiting for that timeout to expire, therefore, delays the relaying process. Second, each relaying act must undergo a resource-request handshake between the relaying UE and the eNB, trailing the Trickle timeout, and taking around 10ms. The above delays accumulate at every hop. Third, when the broadcast area includes several cells, messages can only traverse each cell *sequentially*, propagating from one end of it to the other, and then crossing the cell border.

In this paper, we pursue a different approach, advocating instead an *active* role of the network in the relaying process: if an eNB is aware that one of its UEs has started a D2D-based broadcast – hence relaying at nearby UEs will be required – it can compute the probability that the surrounding UEs have received the message at a given time. Based on this information, it can *proactively* issue relaying grants to those UEs whose relaying is necessary, thus dispensing with timeouts and handshakes, making the broadcast diffusion considerably faster. Moreover, it can compute *optimal* relaying sets at each step, to minimize resource consumption and maximize the number of recipients,

while avoiding interference. Furthermore, when it comes to multi-cell broadcast areas, the originating node's eNB can forward the broadcast message to its neighboring eNBs, whose cells are included in the broadcast area, using the X2 wired connection. This allows the broadcast process to be started immediately across possibly many cells, which further reduces its diffusion time. We model the selection of the optimal relaying set at the eNB as a probabilistic set covering problem [19], which can be solved fast enough not to be a bottleneck (i.e., in few milliseconds), and show via simulation that the gains over distributed, UE-driven relaying are remarkable.

Broadcast diffusion problems have been addressed in wireless networks, such as ad hoc or sensor ones (e.g., [12]-[13]), where resource contention is inherently distributed. In LTE-A, instead, resource scheduling is centralized, which makes the problem different. Our broadcast problem also resembles the one of channel assignment and/or link scheduling in Wireless Mesh Networks (WMNs), where centralized decision making is sometimes assumed (see, e.g., [7]-[8]). In WMNs it is normally assumed that nodes are equipped with a small number of radios, which can be tuned to a limited number of channels. With LTE-A, the number of channels (i.e., RBs) is in the order of 50-100, and each UE can listen to them all simultaneously with a single radio. Moreover, the algorithms in the literature usually assume periodic transmissions, long-term, semi-static resource allocations, and unicast point-to-point transmissions. We need instead event-triggered resource allocation, in time-varying topologies, to reach user-defined target areas, and these cannot be achieved using these algorithms.

The rest of the paper is organized as follows: Section 2 provides some background information. Section 3 presents our contribution, which is evaluated in Section 4. Section 5 draws conclusions.

## 2 Background on LTE-A

This section describes the aspects of the LTE-A protocol stack that are connected to the problem at hand, namely the MAC, the resource allocation and point-to-multipoint (P2MP) D2D communications.

The LTE MAC assembles the data from the upper layers into a Transmission Block (TB), and sends it down to the PHY layer for transmission. Each transmission must be allocated some frequency resources. Resource allocation is performed by the eNB's MAC layer on each Transmission Time Interval (TTI) of 1ms. The available bandwidth can be represented as a vector of *Resource Blocks* (RBs), which have to be allocated to backlogged UEs according to some scheduling policy. The number of RBs required to send a TB depends on the selected Modulation and Coding Scheme (MCS). In fact, the MCS defines the number of bits that one RB can carry and is derived from the Signal to Interference and Noise Ratio (SINR) perceived by the UE. In the DL, the eNB sends the TB to a UE on the allocated RBs. In the uplink (UL), the eNB sends a *transmission grant* to the UE, which specifies which RBs the UE can use to carry its TB, using which MCS. In order to inform the eNB about the presence of UL traffic, a UE transmits a *Buffer Status Report* (BSR). However, it can only do it if it is scheduled enough RBs

to contain a BSR. Otherwise, the UE needs to start a *Random Access procedure* (RAC), so that the eNB can issue a transmission grant for the BSR in a future TTI.

Network-controlled *point-to-multipoint* (or *one-to-many*) D2D communications [5], enable direct communication between proximate UEs, without using the traditional two-hop path through the eNB. The new D2D link is also referred to as *sidelink* (SL). In a Frequency Division Duplex (FDD) system, SL communications usually occurs in the UL spectrum, which is likely to be less loaded than the DL one, and requires D2D-enabled UEs to be equipped with Single-Carrier Frequency Division Multiple Access (SC-FDMA) receivers [6]. Resources are allocated by the eNB, using *scheduled resource allocation* (SRA). In SRA, the UE sends a RAC requests to the eNB, followed by a BSR. Then, the eNB schedules resources according to the size of the BSR and communicates its decision to the UE. BSR reporting is similar to that of the UL case. This is shown in Fig. 1, along with its timing. An alternative allocation mode, called Autonomous Resource Scheduling (ARS) can also be used, whereby the eNB statically allocates a pool of resources, and UEs can access them autonomously (thus possibly colliding). ARS has been proved in [18] to be inefficient, hence is not considered further in this paper. D2D transmissions can leverage *frequency reuse* [11]: since they normally occur at reduced power, an eNB can grant the same RBs to more than one sender, provided that their intended recipients are far enough to tolerate the ensuing interference.

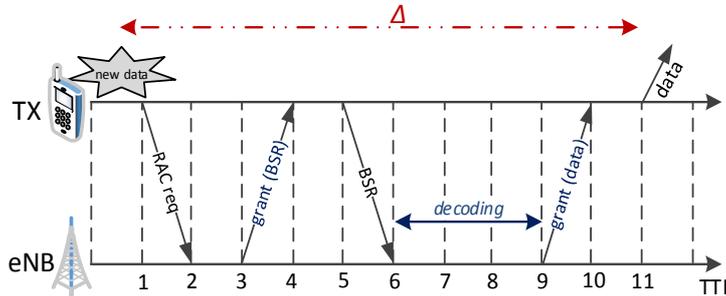


Fig. 1. Scheduled Resource Allocation for D2D transmissions

### 3 Scheduled Multihop D2D Broadcasting

In the following, we consider an LTE-A system where D2D-enabled UEs are located in a multicell network. Each UE runs an application that may generate messages (e.g., vehicular collision alerts) destined to all UEs within an arbitrary *broadcasting area*. The message includes the coordinates of the originating UEs (e.g., taken from GPS) and the broadcast radius (representing more elaborate geofences is left for future study). Our problem is to reach possibly all the UEs in the broadcast area, using P2MP D2D transmissions, relayed by UEs themselves. UEs that receive more than one message in the same TTI will only attempt to decode the one received with the strongest power. Each UE that possesses the message and is within the broadcast area will *wait for a*

*relaying grant*, issued by the eNB, before relaying the message. The transmission power of UEs on the SL is known at the eNB (which may also set it itself for the whole cell using Radio Resource Configuration messages).

We assume that the eNB runs a cross-layer application that can receive the messages generated by the UEs, and can instruct the MAC layer to issue recognizable P2MP D2D *relaying grants* to the UEs that can be expected to advance the diffusion of that message in the broadcast area, using a specific MCS. Applications on different eNBs can communicate via the X2 interface. Moreover, we assume that the eNB application knows the position of its associated UEs. This can be accomplished if UEs are endowed with GPS tracking capabilities and report their geographical coordinates to the eNB. Another solution is provided by Mobile-edge Computing (MEC) technology, which is being standardized by ETSI [21]. MEC entails the presence of application servers co-located with the eNB that can perform intensive, context-aware computations. In particular, one of the foreseen features provided by MEC is the real-time tracking of UEs location, based on network measurements [22]. Knowing the D2D transmission power and location of UEs allows an eNB application to compose a *reachability graph* (RG). The latter is an undirected clique between all UEs, and associates to each edge the success probability of a transmission occurring between the endpoints of said edge. In fact, considering a pair of UEs  $(i,j)$ ,  $i$  being the transmitter and  $j$  being the receiver, the eNB can estimate the path loss between them, and use it to compute the power received by  $j$ . Since the eNB knows the MCS used for the transmission and the Block Error Rate (BLER) curve<sup>1</sup>, it can use the above three elements to extract the probability of successful transmission. While UEs *can* be mobile, we show later on that our broadcast diffusion times are short enough that they can be considered static during the broadcast process, and so can the RG.

Assume that an *originating UE* wants to broadcast a message using P2MP D2D transmissions. The basic idea behind our algorithm is that, under the above assumption, the eNB can compute a probabilistic view of which UEs in the first-hop neighborhood of the originator have received the message, and what other UEs within the broadcast area have not. Based on this knowledge, the eNB can find a *minimum set* of relayers that can reach all the UEs that may receive the message next, and iterate the process until all the intended recipients have been reached with sufficient probability. By adopting a stochastic view of the diffusion of the broadcast, the eNB can issue *proactive, unsolicited relaying grants* to UEs, which speeds up the relaying process, by dispensing with the resource-request handshake. By minimizing the set of transmitters, resources are saved and interference is reduced.

Hereafter, we first discuss what happens in the cell where the originating UE resides, and then we show how to adapt the algorithm to a multicell environment.

### 3.1 Algorithm Description

When a UE needs to broadcast a message, e.g. UE2 in Fig. 2, it sends a RAC request to the eNB, followed by a BSR, so as to obtain a P2MP D2D grant for transmitting the

---

<sup>1</sup> A BLER curve maps the received power and the MCS into a reception error probability [20].

message. The eNB can recognize that the BSR is for D2D broadcasting, e.g., because a dedicated value of the Logical Connection ID (LCID) field of the BSR is used, and start the algorithm depicted in Fig. 3. First, it sends *two* grants to the UE: one on the UL, to get the message itself, and another on the SL, to propagate the message to the first-hop neighborhood of the originator. Then, the eNB computes the set  $D$  of UEs under its control that are within the broadcast area, (those within the dashed circle in Fig. 2), by reading the originator's coordinates and the broadcast radius, and comparing the location of the UEs against them.

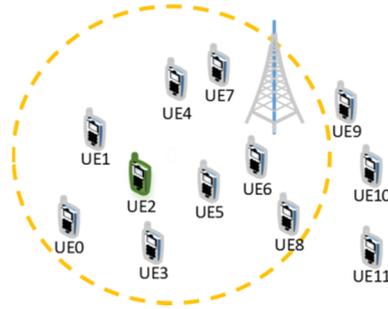


Fig. 2. - Example scenario

Then, it obtains the probability that a transmission from  $i$  is correctly decoded by  $j$ , by combining the BLER curve of the MCS used for D2D transmissions the transmission power on the SL, and the attenuation between  $i$  and  $j$ . This way, the eNB estimates the probability that each  $UE$  in  $D$  has the message transmitted by the originator. With reference to Fig. 3, the eNB computes two sets: a *Transmission Set (TS)*, i.e. the set of UEs that have the message with probability  $P_i > \alpha$ , and a *Receiving Set (RS)*, i.e. the set of UEs that may receive the message with probability  $P_j > \alpha$  from at least one UE in TS. Value  $\alpha$  is a probability threshold value that defines the reliability of the broadcasting (e.g. 0.9 or 0.95). These sets can be computed in  $O(D)$  on the RG. The eNB can then iterate the following algorithm: given the TS and related RS, find the *Candidate Relaying Set (CRS)*, i.e. the *minimum* subset of TS that should transmit so that each UE  $j$  in RS will receive with probability  $P_j > \alpha$ , and schedule a relaying grant to UEs in the CRS only. The above algorithm terminates when all UEs in  $D$  have the message with the required probability. The probability  $P_j$  that a UE  $j$  in RS receives a message is  $P_j = 1 - \prod_{i \in TS} (1 - P_i P_{ij} x_i)$  where  $x_i$  is equal to 1 if UE  $i$  is included in the CRS,  $P_i$  is the probability that UE  $i$  has the message, and  $P_{ij}$  is the probability that the transmission from  $i$  to  $j$  is successfully decoded. Note that the above formula allows that UE  $j$  receives the message more than once from different senders  $i$ , on non-overlapping transmissions. The issue with the above formulation is that, under *constrained resources* (i.e., a maximum number of RBs), reaching *all* the RS may not be possible, since it may require more RBs than available. For this reason, and due to the fact that frequency reuse is difficult to model, we *first* compute the CRS assuming *infinite resources* and *non overlapping transmissions* (i.e., no frequency reuse), and then take into

account resource constraints and frequency reuse later. Computing the resource-unconstrained CRS entails solving a probabilistic variant of the set covering problem (SCP) [19]. This problem is NP-hard, but (as we show in Section 4) it is still solvable in few milliseconds in fairly large networks, hence it does not constitute a performance bottleneck. The formulation is as follows:

$$\begin{aligned} \min \quad & \sum_{i \in TS} x_i \\ \text{s.t.} \quad & \sum_{i \in TS} x_i \log(1 - P_i P_{ij}) \leq \log(1 - \alpha) \quad \forall j \in RS \quad (i), \\ & x_i \in \{0, 1\} \quad \forall i \in TS \quad (ii) \end{aligned} \quad (1)$$

where (i) is the linearized version of the above-mentioned reception probability constraint, making it a Mixed Integer-Linear problem. Once (1) has been solved, we apply frequency reuse to the solution, to obtain as compact as possible an allocation. We do this by fitting several transmissions on the same RBs, taking care not to increase the interference so much that the decoding probability falls below the threshold  $\alpha$ . This is done using the scheduling algorithm whose pseudocode is shown in Fig. 4.

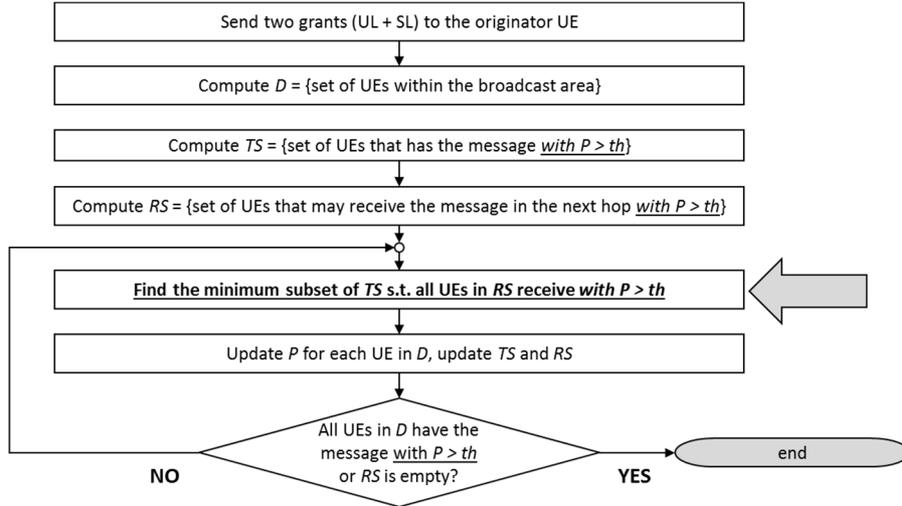


Fig. 3. - Flowchart of the algorithm

In the pseudocode, we denote with `rbAlloc` a map that stores the set of UEs allocated on each RB  $k$ , and `ueAlloc` is a vector that stores the RB allocated to UE  $i$ . For ease of reading, we assume that each UE is allocated one RB (all UEs transmit the same message with the same MCS, hence this assumption comes without loss of generality). Starting with serialized UE allocations (lines 1-6), the algorithm iterates through the UEs, (lines 9-26), tentatively moving their allocation over some other UE's. In doing this, it updates the decoding probabilities taking into account the added interference (lines 15-17), and checks whether they are still above the threshold (line 18). If so, the

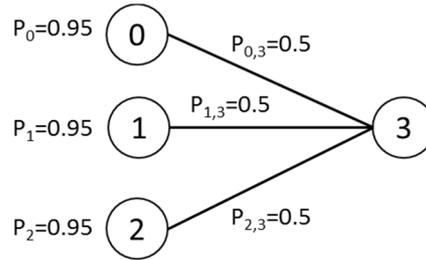
new allocation is enforced (lines 19-21). The complexity of the scheduling is  $O(|CRS|^2 \cdot B)$ , where  $|CRS|$  is the number of UEs in the CRS and  $B$  is the total number of RBs. Note that the  $|CRS|$  can be expected to be small (and, significantly, smaller than  $|D|$ ), since it is the outcome of minimization problem (1).

```

1. // initialization
2. sort UEs in CRS according to a given policy
3. for each k in B
4. | add UE k to rbAlloc[k]
5. | ueAlloc[k] = k
6. end for
7. // allocation
8. for each UE i in CRS
9. | tempRbAlloc = rbAlloc
10. | prev = ueAlloc[i]
11. | remove i from tempRbAlloc[prev]
12. | for each k in tempRbAlloc
13. | | add i to tempRbAlloc [k] // try UE i on RB k
14. | | for each UE j in CRS // eval. probabilities
15. | | compute P_j according to tempRbAlloc
16. | | end for
17. | | if P_j > alpha, for each UE j // update alloc.
18. | | | remove i from rbAlloc [prev]
19. | | | add i to rbAlloc [k]
20. | | | go to the next UE
21. | | else
22. | | | remove i from tempRbAlloc[k]
23. | | end if
24. | end for
25. end for

```

**Fig. 4.** - Pseudo-code for the heuristic scheduling algorithm

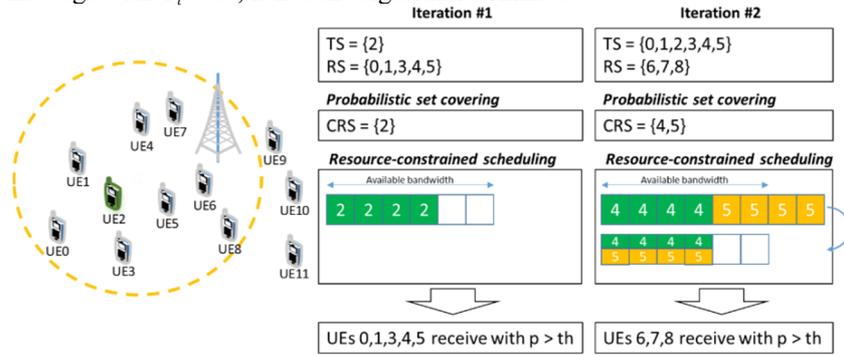


**Fig. 5.** - Example of unreachable UE in the RG

After the scheduling, the part of the allocation exceeding the available RBs, if any, is discarded. The rest of the UEs in the schedule are sent relaying grants by the eNB. At the end of an iteration, the eNB computes the new probabilities  $P_i$  for each UE in  $D$ . A new iteration is repeated after four TTIs, which is the standard time limit to decode a message, using the updated probabilities. The algorithm terminates if either  $P_i > \alpha$  for each UE in  $D$ , or the updated RS is empty. In the latter case, it means that some UE

in  $D$  could not be reached by the message with  $P_j > \alpha$ . An example of RG representing this condition is shown in Fig. 5, where  $\alpha=0.95$  and  $P_i$  and  $P_{ij}$  are those shown in the figure. The probability of reaching UE3 is given by  $P_3 = 1 - \prod_{i=0}^2 P_i \cdot P_{i3} = 1 - (0.95 \cdot 0.5)^3 = 0.893$ , which is smaller than  $\alpha$ .

An example of message broadcasting using the proposed approach is reported in Fig. 6: UE2 begins the procedure and it is clearly the only UE within the TS. UEs within the green circle are those in  $D$ , i.e., those that should receive the message. In the first iteration, UEs 0,1,3,4 and 5 are close to UE2 and are included in the RS at the first step. The trivial solution of the probabilistic SCP is  $\text{CRS}=\{2\}$ , which is allocated by the heuristic in RBs 0-3. After the transmission, UEs in the RS have the message with  $P_i > \alpha$ . In the second iteration of the algorithm, see again Fig. 6, the eNB runs the probabilistic SCP and finds that UEs 4 and 5 can cover all the UEs in the RS. Then, the scheduler exploits frequency reuse to allocate UEs 4 and 5 in the same RBs. Note that, if frequency reuse is not enforced, either UE 4 or 5 cannot be allocated, since the sum of their allocations exceeds the available RBs. After the transmission, all UEs in  $D$  have the message with  $P_i > \alpha$ , hence the algorithm terminates.



**Fig. 6.** - Example of broadcasting

The described algorithm allows the eNB to guide the broadcasting on the SL with the minimum amount of transmissions, which means reducing the allocated resources and the interference. Moreover, the resource request handshake is required *at the first hop* only (where it is unavoidable), and subsequent hops can occur as fast as the decoding allows. However, since the algorithm is based on probabilistic models, it is subject to the following problems:

- The eNB might schedule UEs that have not received the message, hence propagating errors in estimating the TS and the CRS;
- The eNB might *not* schedule UEs that have the message and could easily reach other UEs (for example, because their probability is slightly below the threshold). This may increase broadcast latency.

However, a tradeoff can be struck between the two issues by tuning threshold  $\alpha$ . A larger  $\alpha$  reduces the first problem, at the cost of additional transmissions and increased delay. A smaller  $\alpha$  allow the eNB to compute larger TSs, increasing the risk of wasting

resources by scheduling UEs that do not actually have the message. We will assess the impact of  $\alpha$  in the performance evaluation section.

### 3.2 Target Areas Consisting of Multiple Cells

We now consider the case of broadcasting a message over a multiple cells, each one serve by different eNB. We first observe that the algorithm described above would still work in a multicell environment, without any modification. With reference to Fig. 7, left, D2D transmissions can be received also from UEs served by different eNBs. The receiving UEs can thus perform their own RAC request to their serving eNB, which starts an instance of the algorithm in its cell. The new instance is completely independent from that running in the neighboring cell. However, this is inefficient, since it entails summing up delays of neighboring cells. Since the eNB application in a cell is fully aware of the target area, it can *immediately* contact its peer applications on all neighboring eNBs whose cells intersect the target area, using the X2 interface, allowing them to start their own D2D-based internal broadcast much sooner. With reference to Fig. 7, right, eNB1 gets the message from UE0, starts the algorithm and, at the same time, forwards the message to eNB2 using the X2 interface. eNB2 sends the message to one (or more) UE(s) under its coverage (UE4 in the example) and, in turn, starts the algorithm. This yields remarkable improvements in vehicular scenarios, where the broadcast area typically covers a stretch of road served by several eNBs.

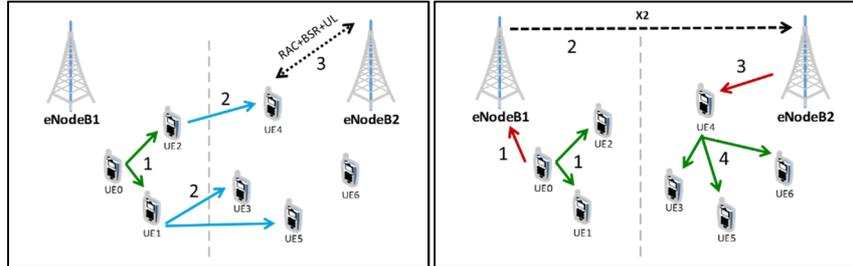


Fig. 7. Broadcasting in a multicell using cross-cell-border (left) or X2 (right) communications

Starting a broadcast within a cell requires that the eNB (which has the message) finds one or more candidate UEs to start the multihop relaying, addressing them with a DL transmission (i.e. the red arrow originating from eNodeB2 in Fig. 7, right). While the eNB may do this according to any suitable policy, we choose to select the UE  $C$  having the *minimum maximum shortest path* to the other UEs within the broadcast area in its cell, i.e.:  $C = \arg \min_{i \in D} \left\{ \max_{j \in D, j \neq i} \{ SP_{i,j} \} \right\}$ , where  $SP_{i,j}$  is the length of the shortest path between UEs  $i$  and  $j$ .

## 4 Performance Evaluation

In this section we evaluate the behavior of the proposed scheduled broadcasting algorithm in various conditions. For this purpose, we use SimuLTE [10], an OMNeT++-

based system-level simulator of LTE networks. We first analyze the performance using various values of the threshold  $\alpha$ . We simulate a single-cell environment (scenario #1) where one UE broadcasts a message to 100 UEs located in a radius of 500m, as we show in Fig. 8. Then we evaluate the performance of the proposed methods for the dissemination of a message in multi-cell environment (scenario #2), composed of five eNBs at a distance of 400m, as we show in Fig. 9. The main simulation parameters are listed in Table 1. 95% confidence intervals are reported in the graphs.

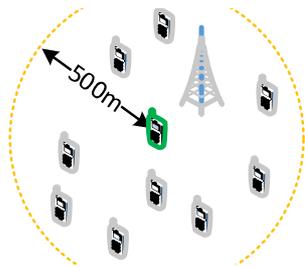


Fig. 8. Simulation scenario #1

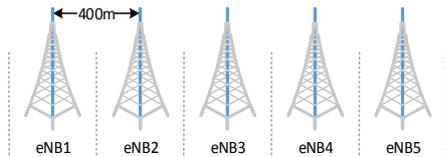


Fig. 9. Simulation scenario #2

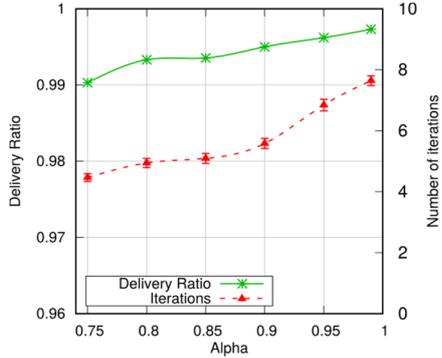
Table 1. Main simulation parameters

Parameter	Value
Carrier frequency	2 GHz
Bandwidth	10 MHz (50 RBs)
Path loss model	ITU Urban Macro
eNB Tx Power	46 dBm
UE Tx Power (UL)	30 dBm
UE Tx Power (SL)	15 dBm
eNB Antenna gain	18 dB
Noise figure	5 dB
Cable loss	2 dB
Mobility model	Stationary
Simulation time	100 seconds

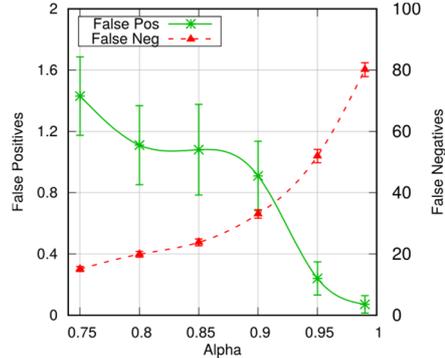
Starting with scenario #1, Fig. 10 shows the tradeoff between the number of iterations required to complete the broadcasting and its reliability, defined as the percentage of UEs in the target area that have received the message. We observe that the delivery ratio stays above 99% in any case. As expected, a larger  $\alpha$  implies more reliable dissemination at the cost of increasing convergence time, which approaches eight iterations when  $\alpha = 0.99$ . Fig. 11 reports the occurrence of *false positives* (i.e., relaying grants given to UEs that do not have the message) and *false negatives* (i.e., UEs that have the message but are considered to be below threshold by the eNB). We note that larger  $\alpha$  reduces the risk of false positives, allowing the network to minimize resource waste. On the other hand, false negatives significantly increase with  $\alpha$ , meaning that the algorithm's estimates are fairly conservative. In the following, we set  $\alpha = 0.95$ .

As for algorithmic complexity, Fig. 12 reports the time required to solve optimization problem (1) at every iteration of the algorithm. Since the problem complexity depends on the cardinality of both TS and RS, solving times are higher at intermediate iterations, when both TS and RS includes many UEs and the solver needs to explore a larger number of solutions. The figure shows that the time is always lower than 10ms. These times are computed on off-the-shelf hardware equipped with an Intel(R) Core(TM) i7 CPU at 3.60 GHz, 16 GB of RAM and a Linux Kubuntu 16.04 OS, using

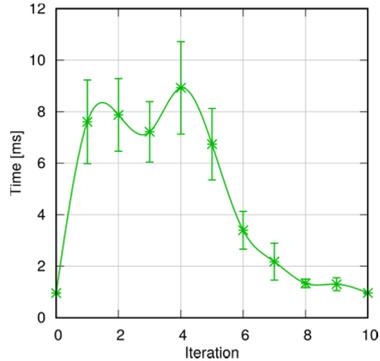
the CPLEX general-purpose solver. We believe that these times can be reduced by using dedicated hardware and/or by devising problem-driven solution algorithms. Note that we could also stop computations after a time limit, and use the best solution found by the branch-and-cut algorithm used by the solver, as a heuristic solution.



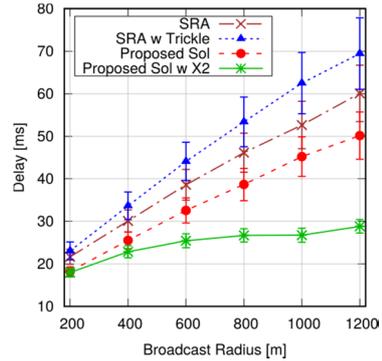
**Fig. 10.** Average delivery ratio and number of iterations, with varying  $\alpha$



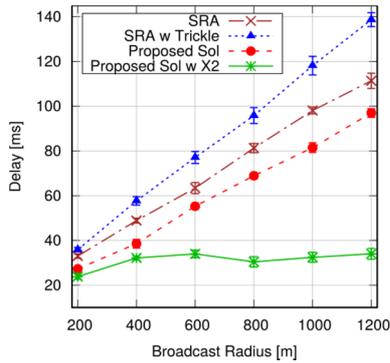
**Fig. 11.** Average number of false positive/negatives per broadcast, with varying  $\alpha$



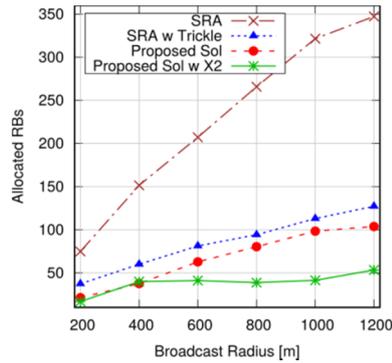
**Fig. 12.** Solving time of the SCP variant for finding the CRS



**Fig. 13.** - Average application-level delay for increasing broadcast radius



**Fig. 14.** - 95<sup>th</sup> percentile of the application-level delay for increasing broadcast radius



**Fig. 15.** - Average amount of allocated RBs required for completing a broadcast

We now consider scenario #2. Fig. 13 shows the average application-level delay experienced by UEs within the broadcast area. We compare our scheme against distributed, UE-requested relaying [18], with or without using Trickle and with SRA as a network resource allocation strategy. SRA with Trickle performs the worst, because of the message suppression mechanism. The benefits of the proposed approach with X2 are evident when the broadcast radius increases. In this case, the delay has an upper bound because all the involved eNBs start the broadcasting roughly at the same time instead of waiting for the message to traverse cell borders through UE relaying. For time-critical services, a more relevant metric is the 95<sup>th</sup> percentile of the application-level delay, which is reported in Fig. 14. The proposed solution with inter-eNB X2 covers the broadcast area in less than 40 ms. For shorter-range broadcasts, (e.g. 200m), only 20-25 ms are necessary. Fig. 15 reports the average number of RBs occupied by the broadcast of a single message. SRA consumes much more RBs than the other solutions. Adding the Trickle suppression mechanism reduces the amount of allocated RBs, at the price of increased latency, as explained above. On the other hand, our approach further reduces the utilization of RBs. It is worth observing that adding X2 communication adds further benefits: this is due to the fact that the originating UE in each cell is chosen by the eNB to minimize the maximum shortest path.

## 5. Conclusions and Future Work

In this paper, we proposed a method for D2D-based message broadcasting over multiple cells. Our scheme relies on endowing the eNB with a resource allocation algorithm that estimates which UEs have/have not the message, using probabilistic models, and schedules unsolicited relaying grants to the minimum number of UEs required to propagate the broadcast. We also described how to exploit X2 communication to speed up the broadcast process in a multicell environment. Simulative results show that the proposed solution allows a message to be disseminated in a fast and resource-efficient way over large broadcast areas, e.g. covering more than 99% of UEs within a 1200m radius from the originator in less than 40ms, using less than one RB per reached UE on average. Future work includes gathering context information from the network to better characterize the broadcast area, e.g. targeting only the user that are interested on a specific content, or notifying an alert only to cars moving along the road involved in the accident. Moreover, the alert reader will have noticed that the same solution adopted to speed up inter-cell diffusion (i.e., a unicast relaying by the eNB on the DL) could, in principle, be used (sparingly) also *within* a cell, to reduce the number of hops, or to reach otherwise unreachable target UEs. We are actively investigating this tradeoff at the time of writing.

## References

1. L. Lei, Z. Zhong, C. Lin, X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," IEEE Wireless Comm., no. 3, pp. 96-104, June 2012

2. C. Vallati, A. Viridis, E. Mingozzi, G. Stea, "Exploiting LTE D2D Communications in M2M Fog Platforms: Deployment and Practical Issues", Proc. of IEEE WF-IoT 2015, Milan, Italy, pp. 585-590
3. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, "Internet of Things for Smart Cities", in IEEE Internet of Things Journal, vol. 1, n. 1, pp. 22-32, February 2014.
4. 3GPP - TS 23.303 v15.0.0, "Proximity-based services (ProSe); (Release 15)", June 2017
5. 3GPP - TS 36.843 v12.0.1, "Study on LTE Device-to-device Proximity Services: Radio aspects (Release 12)", March 2014
6. 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
7. P. Cappanera, L. Lenzini, A. Lori, G. Stea, G. Vaglini, "Optimal joint routing and link scheduling for real-time traffic in TDMA Wireless Mesh Networks", Computer Networks, 57(11), 2013, pp. 2301-2312
8. R. Draves, J. Padhye, B. Zill "Routing in multi-radio, multihop wireless mesh networks", Proc. ACM Mobicom'04, Philadelphia, US pp. 114-128
9. P. Levis, N. Patel, D. Culler and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks", Proc. 1st USENIX/ACM Symp. NSDI, pp. 15-28, 2004
10. 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
11. 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, DOI: 10.1007/s11276-016-1193-3
12. B. Williams, T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," Proc. of MOBIHOC '02, Lausanne, CH, 2002
13. P. Kyasanur, *et al.*, "Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks," Proc. IEEE MASS 2006, Vancouver, BC, pp. 91-100.
14. A. Cimmino et al. "The Role of Small Cell Technology in Future Smart City Applications", Trans. on Emerging Telecom. Tech. 25 (1), 11-20
15. J. M. B. da Silva Jr, G. Fodor, T.F. Maciel, "Performance Analysis of Network-Assisted Two-Hop D2D Communications", Proc. Globecom 2014, Austin (TX), 8-12 Dec. 2014
16. Siyi Wang, et al., "Outage Probability for Multi-Hop D2D Communications With Shortest Path Routing", IEEE Comm. Letters, 19(11), Nov. 2015
17. G. Rigazzi, et al., "Multi-hop D2D networking and resource management scheme for M2M communications over LTE-A systems", IEEE IWCMC '14, Nicosia (CY), 4-8 Aug. 2014
18. G. Nardini, G. Stea, A. Viridis, "A fast and reliable broadcast service for LTE-Advanced exploiting multihop device-to-device transmissions", Future Internet, 2017, 9(4), 89
19. V. Chvatal, "Greedy Heuristics for the Set-Covering Problem", Mathematics of operations research, 4 (3), 233-235, 1979.
20. C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, M. Rupp, "Simulating the Long Term Evolution physical layer", European Signal Processing Conference, Glasgow, 2009.
21. ETSI GS MEC 003 v1.1.1, "Mobile Edge Computing (MEC); Reference architecture", March 2016.
22. ETSI GS MEC 002 v1.1.1, "Mobile Edge Computing (MEC); Technical requirements", March 2016.