A low-latency and reliable multihop D2D transmissions scheduling algorithm for guaranteed message dissemination

Giovanni Nardini, Giovanni Stea, Antonio Virdis

Abstract—**In critical alert services (e.g., a collision alert in a vehicular network) a message must reach all the recipients in a prespecified area within a maximum time, usually of few milliseconds, with guaranteed reliability. In this paper, we consider a Device-to-device (D2D)-enabled cellular network where User Equipments (UEs) use D2D transmissions to spread a message in their proximity, according to a centrally computed multihop schedule. Reception of D2D transmissions is probabilistic, with probabilities known a priori, e.g. based on the distance between two UEs. We want a given message to reach all the UEs in the network, within a maximum amount of time, with a pre-specified target probability. We show that the problem of computing: a) the minimal set of UEs that should initially possess the message to be disseminated, and b) the schedule that achieves the above objectives, is integer-non-convex, hence too complex to be solved optimally, and we propose a polynomial heuristic based on iterative decompositions, which always finds a feasible solution in negligible computation time. We analyze the performance of our scheme via simulation, showing that our centralized approach outperforms distributed ones relying on node cooperation, is considerably faster and requires far fewer transmissions.**

Index Terms—**Alert Services, Device-to-Device, Optimization, Resource Allocation.**

I. Introduction

Critical alerts are services whose messages should reach all the intended recipients in an area with a high, quantifiable reliability, and within a pre-specified time, which is usually short (in the order of tens of milliseconds). Examples are vehicular collision alerts [37], malfunctioning alerts in Industry-4.0 manufacturing plants [38], periodic distribution of coordination information for swarming robots or platooning vehicles [39], etc. In all the above cases there are clear requirements on both deadlines and probability of reception.

We consider a cellular network where Device-to-Device (D2D) transmissions can be scheduled by the Base Station (BS) [1]. D2D transmissions are broadcast: UEs that have the message at a certain time may relay it in their proximity. The area where the message must be conveyed is however larger than a single UE's D2D radius. Moreover, D2D transmissions are subject to probabilistic reception, e.g., the further a recipient is from the transmitter, the higher the error probability. We assume that the BS possesses: i) a message that it wants to disseminate, and ii) a graph representation of the network, where the vertices are the UEs and the edges are the reception probabilities. In a 4G/5G cellular network, the BS can in fact estimate the position of the UEs (e.g., connected cars) using location services [2][3], hence it can build the above-mentioned graph given a model of the channel. The BS can send a message to a (small) subset of UEs, using downlink transmissions, thus creating a set of initial transmitters, and it can schedule D2D transmissions to spread that message in the network. Scheduling occurs in a round-based fashion. However, the BS does *not* know the reception outcome of a D2D transmission, i.e., which neighbors of a given UE actually received the message after a scheduled transmission: it can only *estimate* the probability of a UE having the message at a certain round, based on the reception probabilities and the set of previously scheduled D2D transmissions. The problem that we address in this paper is how to centrally compute: a) a minimal set of *initial transmitters*, i.e., UEs that should be seeded with the message by the BS, and b) a schedule of D2D transmissions at each UE, such that every UE in the network will possess the message with a pre-specified target probability α after at most k scheduling rounds.

As far as we are aware, the above problem is new, and fundamentally different from message diffusion problems addressed in a *distributed* way in the context of Mobile/Vehicular Ad Hoc Networks (MANET/VANETs) [24], [28], [29], [34]. We first show that our problem can be formulated as an optimization problem, which is however mixed integer-non-convex, hence $\mathcal{NP}$-hard and generally too hard to solve optimally online, even at small scales. We then present a polynomial-time solution algorithm, which always finds a feasible solution and computes *one round* at a time, by solving a mixed linear-integer problem (MILP). While the latter is still $\mathcal{NP}$-hard, fast polynomial heuristics exist, based on continuous relaxation and rounding. We show that computing the set of relays at each stage takes a small time, comparable to the period of MAC-layer message transmission in a cellular network, even over large graphs (i.e., hundreds of vertices). We analyze the performance of our scheme, as a function of the network topology, the target reliability and the maximum number of hops. We show that it allows a message to reach its intended recipients with a *sublinear* number of transmissions, thus being more efficient than addressing each UE directly from the BS, while still meeting timeliness and reliability guarantees. To the best of our knowledge, ours is the first *centralized* scheme for reliable real-time message dissemination using direct

Giovanni Nardini, Giovanni Stea and Antonio Virdis are with the Dipartimento di Ingegneria dell'Informazione, University of Pisa, Largo L. Lazzarino 1, Pisa, Italy (name.surname@unipi.it).

broadcast transmission. We compare it against two similar schemes [8]-[9], showing that – besides being able to provide guarantees, whereas the others are not – it is also *more efficient* than both at large scales.

The main contributions of our paper are summarized as follows:

- formulation and theoretical analysis of a new problem, i.e., the one of centralized scheduling of messages via D2D multihop transmissions, with both time and reliability guarantees, and a discussion of its hardness;
- a heuristic scheme to solve the above problem in polynomial time, a proof of its correctness and an evaluation of its complexity;
- a proof of the practical feasibility of our scheme, obtained by evaluating the solution time on off-the-shelf hardware in various conditions;
- an extensive performance evaluation of our scheme, showing how it performs with various time and reliability requirements and how it scales with the problem size.

The rest of the paper is organized as follows: we discuss the related work in Section II. Section III introduces the system model and formulates the problem, whereas Section IV describes our algorithm. We evaluate the performance of our scheme in Section V. Finally, Section VI concludes the paper.

II. Related Work

The problem of message diffusion has been studied extensively in Mobile Ad Hoc Networks (MANETs). The problem dealt with in these works is to find a multi-path tree to deliver data from one source to a set of destinations [24], [28], [29]. Since no centralized control is possible in MANETs, routing protocols are inherently distributed, and access to the shared medium is unscheduled, hence prone to collisions. This makes it hard, if possible at all, to disseminate data with strict latency and reliability requirements.

Message diffusion has also been studied in the context of Delay-tolerant Networks (DTNs), i.e. networks where inter-node contacts are sporadic. Works [4]-[7], [30] addressed the problem of disseminating information to a group of mobile users using *opportunistic* communications. Within these works, relaying hops occur because of user encounters due to random mobility. The delay of a link between two users is thus a random variable, whose support has the typical timescales of human-related mobility (i.e., seconds to days), and dissemination deadlines are in the order of hours or days (hence the name *delay-tolerant networks*). In these, it makes sense for a user to *store* a message and rely on mobility to provide relaying opportunities later. In our problem, instead, deadlines are fundamentally different – and much shorter (i.e., up to few tens of milliseconds). In such short timespans, users do not move appreciably, hence the reliability of a link is constant. Therefore, it would make no sense to store a message and wait for future relaying opportunities. Moreover, these works rely on distributed, probabilistic decision making and do not provide *a priori* guarantees.

In [4], authors consider the problem of delivering delay-tolerant traffic from content service providers exploiting opportunistic Bluetooth communications to offload cellular resources. To do this, they select a *target set* of K UEs to be addressed using downlink transmissions from the BS and consider them as starting points for multihop dissemination of content (in the order of megabytes) towards other UEs. The target set is selected to maximize the number of UEs that will receive the data through opportunistic communications within a given deadline (in the order of hours). Parameter K is an *input* to the algorithm, and it determines its effectiveness in a specific scenario. Minor variations on the scenario (e.g., location of UEs) make a given value of K suboptimal. In our work, instead, the set of starting points is an *output*, and its cardinality is chosen to be the minimum possible to guarantee both latency and reliability in a scenario.

Data dissemination from the BS to the UEs is also tackled in [5]. In this work, authors consider two-hop communications only for delivering data to *subscribers* UEs through a set of *helper* UEs. The latter are identified a priori so their optimal selection is not considered in the paper. The focus of this work is mainly on differentiating the type of information to be delivered, which might have different characteristics in terms of size, lifetime and set of UEs interested in receiving it. Authors formulate an optimization problem that aims at maximizing the amount of data delivered to interested UEs before data lifetime expires, by choosing which helper UE will deliver which chunk of data to which subscriber UE and taking into account the limited buffer capacity at helper UEs.

In [6], still in the context of opportunistic networks, authors propose to minimize the number of D2D transmissions required to disseminate data from one UE to all the others, with a probabilistic delay guarantee. They first discuss an offline, centralized heuristic that finds predetermined routes from the source to all destinations. Then, they propose a distributed algorithm where each UE can make online decisions to adapt to the current topology of the network. Work [7] also proposes delivering data from a single source within a time constraint, by selecting a starting set of relays. Each relay then forwards the message to only one of its neighbors and so on. Both relays and next hops are selected according to forwarding probabilities computed based on social interactions among UEs. Work [30] presents a mathematical model of message diffusion in an epidemical DTN.

Work [32] optimizes the *injection* of content into the network, which precedes the D2D-based dissemination phase, leveraging Almost Blank Sub-Frames to coordinate interference among neighboring cells.

Content dissemination has also been studied in contexts where D2D schemes leverage social interactions among users, for example selecting the forwarding paths based on social ties or giving incentives to users that participate to the process [25]. Work [33] solves a joint peer discovery, power control, and channel selection problem for the optimization of content dissemination in a D2D-enabled vehicular network, using social relationships inferred offline, with the objective of maximizing the overall rate, without any specific focus on deadlines. However, our work focuses on critical applications including autonomous driving or coordination of swarming robots, for which human-like social interactions are not applicable. Mobility of UEs is instead considered in [26] as a key factor for D2D dissemination, since it determines the probability of two UEs to be in proximity. In the context of our paper, critical alerts need to be delivered within a short time frame upon their generation. Our proposed algorithm produces a short-term schedule for delivering messages and, as we will show later, it completes in few milliseconds. Since positions of UEs cannot vary as much in such a short time, mobility does not play a role in our algorithm.

Some works on cellular networks discuss multihop D2D routing [13]. Such works focus on establishing multihop routes between two endpoints, either proactively or reactively, assuming *one-to-one* (i.e., unicast) D2D communications. The latter are different from *one-to-many* broadcast D2D communications considered in this paper and are obviously inefficient for our problem. Besides, to the best of our knowledge, unicast D2D transmissions have not made their way in a 3GPP standard so far, whereas *broadcast* D2D transmissions are in both 4G and 5G standards. Work [31] uses multihop D2D transmissions to achieve reliable message delivery in a cooperative vehicular scenario. Reliability and delivery times are *outputs*, rather than constraints, of the above scheme, and this scheme cannot provide a priori guarantees on either. Moreover, the delays it considers are in the order of seconds (in fact, it uses motion prediction techniques to account for the movement of vehicles in the above timeframe), and in the considered scenario it achieves reliabilities close to 90% after 8 s. Our scheme instead works with pre-established *deadlines* and *reliability* constraints, and it achieves (by construction) target reliabilities of 95%-99% in few tens of milliseconds.

Works [8]-[9] deal with message diffusion in LTE-A networks using D2D transmissions. In [8], this is done via a distributed application running at every UE. However, this approach provides no guarantee that the message reaches all UEs in a network, let alone within a maximum time. Moreover, every relaying hop undergoes additional delays, first to allow a Trickle window time to elapse in order to decide whether to relay a message or not, and then to request grants to the BS. Work [9] is the first to formulate the problem of message diffusion over a target area, subject to reliability and time constraints. However, although [9] improves over [8] in terms of delay, reliability and resource consumption, it cannot guarantee either a maximum delay or a target reliability. In fact, it assumes a single starting point as an input, hence all UEs not reachable within k steps from it will not get the message. Moreover, it makes no attempt at minimizing the number of D2D grants – rather, it maximizes the number of new recipients at each hop. The algorithm discussed in this paper, instead, has guaranteed performance also in the presence of partitioned networks, and makes optimal use of resources.

Several works have studied broadcast as a graph theory problem. The two problems that appear to be the most closely related to ours are *broadcast domination* [10] and *k-distance domination* [11]. We briefly explain them here, and then explain the differences. Broadcast domination is a variant of the standard dominating set problem, which considers a graph $\mathcal{G}$ and assigns an integer "transmission power" $p_v \geq 0$ to each vertex v of $\mathcal{G}$, such that every vertex of the graph (a "city") is within a distance p from some vertex having $p_v \geq 1$ (a "broadcast station"). Optimal broadcast domination aims at minimizing the sum of p_v assigned

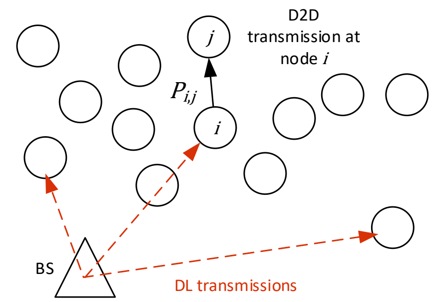

Figure 1 – System model

to the vertices of the graph. This problem can be solved in polynomial time, if p_v is unbounded, and is instead known to be $\mathcal{NP}$-hard otherwise. On the other hand, given an integer $k \geq 1$, a connected graph $\mathcal{G}$, and two sets of vertices $\mathcal{S}$ and $\mathcal{V}$ in $\mathcal{G}$, $\mathcal{S}$ is a distance-k-dominating set of $\mathcal{G}$ if every vertex of $\mathcal{V}$ is at distance at most k from some vertex of $\mathcal{S}$. Optimal k-distance domination seeks to minimize the k-domination number of $\mathcal{G}$, i.e., the minimum cardinality over all distance k-dominating sets. It is known to be $\mathcal{NP}$-hard as well.

Both approaches could, in principle, be used to select the set of starting points for our broadcast coverage (which would however still leave open the problem of computing the entire schedule of D2D grants). However, these problems require a well-defined concept of *distance*. We will show that our probabilistic graph is a clique, where every node is connected to every other, possibly with a very low reception probability. Moreover, the concept of reliability cannot be easily – if at all – superimposed to these frameworks: neither of the above account for the possibility that a node needs to be at k hops from *more than one* starting point, since no starting point *alone* has a k-hop path reliable enough to that node. In our problem, instead, different nodes in the dominating set can join forces to target the same destination node (we will show this via an example in Section III.B).

III. System Model and Problem Formulation

With reference to Figure 1, we consider a network where N UEs are under the control of a *base station* (BS). Transmissions can occur from the BS to the UEs, using the downlink (DL), or between UEs, using D2D. Downlink transmissions are *unicast*, i.e. they have only one recipient, and *reliable*, i.e. the recipient always receives the message correctly. D2D transmissions, instead, are *broadcast*, i.e., can be heard by whichever UE is "close enough", possibly more than one. However, they are *unreliable*: neither a transmitting UE nor the BS knows which recipients, if any, correctly decoded a D2D transmission. The BS knows instead the probability $P_{i,j}$ that recipient j decodes a message transmitted from transmitter i. For instance, $P_{i,j}$ may be a measure of the signal attenuation due to the spatial distance between the UEs, according to the used channel model. We assume that $P_{i,j}$ does not vary with time (we will justify this assumption at the end of this section). The BS schedules all transmissions over *rounds*. In each round, a maximum number of simultaneous D2D transmissions can be scheduled, and they are collision-free. A round begins with the BS issuing *D2D transmission grants* to the UEs that can transmit. UEs that receive a grant *and* possess the message forward it via a D2D transmission.

Under the above hypotheses, the target area can be modeled as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the targeted UEs are vertices, $|\mathcal{V}| = N$, and an edge between two vertices exists if one is reachable from the other through a D2D transmission, as in Figure 1. Edge (i, j) has a probability $P_{i,j}$ associated to it. We allow that $P_{i,j} \neq P_{j,i}$, hence assume the graph to be a directed one. Matrix $\boldsymbol{P} = \|P_{i,j}\|$ contains all the above information.

Our problem is to transmit a message from the BS to the entire set of vertices, exploiting multihop diffusion across edges as much as possible, and to select the minimum set of vertices that should be *seeded* with that message (via DL transmissions) in order to do so. We pose two requirements:

a) the whole process must be completed within k rounds;
b) by the end of the k^{th} round, each UE must possess the message with a probability no smaller than α.

We want to determine which UEs should initially be seeded, and which UEs should relay the message when, i.e., a schedule over time of D2D transmissions.

A schedule can therefore be modeled as:

- a column N-vector $\boldsymbol{X}$, where $x_j = 1$ if UE j is a *starting point*, i.e., the recipient of a DL transmission that seeds the message in the network;
- an $N \times k$ *grant matrix* $\boldsymbol{G}$, where $g_j^{(t)} = w$ means that UE j receives w D2D grants at round t, in which case it will generate w independent D2D transmissions.

The purpose of our scheme is to fill the above vector and matrix, so that - as a primary objective - $sum(\boldsymbol{X})$ is minimal, and - as a secondary objective - $sum(\boldsymbol{G})$ is minimal, and all UEs receive the message with probability at least α after round k. Since DL transmissions are reliable, each UE j such that $x_j = 1$ will possess the message with a probability equal to 1, and therefore in the schedule $g_j^{(1)} > 0\ iff.\ x_j = 1$ (i.e., in the first round, only the initial message holders may receive a D2D grant). Starting from the second round, however, there is no certainty that a UE will possess the message, since reception is probabilistic. Call $m_j^{(t)}$ the probability that UE j has the message after round t, (set $m_j^{(0)} = x_j$). The following iterative formula regulates how probabilities change over the rounds:

$$1 - m_j^{(t)} = \left(1 - m_j^{(t-1)}\right) \cdot \prod_{(i,j)\in\mathcal{E}} \left(1 - m_i^{(t-1)} \cdot P_{i,j}\right)^{g_i^{(t)}}, \quad (1)$$

The left-hand side of the above formula is the probability that node j does not possess the message at round t. This event occurs if: i) node j did not already possess the message at round $t-1$, and ii) all transmissions from nodes that may possess the message at round $t-1$, and receive a grant at round t, are incorrectly received. The right-hand side of the above formula describes the simultaneous occurring of i) and ii), in fact. Expression (1) can be written more concisely as:

$$\boldsymbol{m}^{(t)} = f\left(\boldsymbol{m}^{(t-1)}, \boldsymbol{g}^{(t)}, \boldsymbol{P}\right). \quad (2)$$

Our scheme aims to obtain $m_j^{(k)} \geq \alpha$, $1 \leq j \leq N$.

In the above context, minimizing the number of initial DL transmissions implies relying as much as possible on D2D transmissions to spread the message. The latter reach more UEs per transmission, especially if scheduled carefully, hence make more efficient use of limited spectrum resources.

The following comments are in order: in a cellular network, several DL transmissions can be scheduled in the same scheduling period (called a Transmission Time Interval, TTI), over non-overlapping frequency resources, hence without mutual interference. We assume that network-controlled D2D transmissions are scheduled on a reserved portion of the uplink spectrum [12], without frequency reuse (either internally or in neighboring cells). In D2D transmissions, the BS only issues the grants via control channels, but does not participate in data forwarding, which instead occurs between UEs only [35]. The maximum number of simultaneous non-interfering transmissions depends on both the system bandwidth, the length of the message and the selected transmission format, but it can be expected to be in the order of several tens at least[1]. The number of UEs under the control of a BS can easily reach several hundred (e.g., a congested highway segment, or a large factory hosting a swarm of coordinated robots). TTIs in 4G and 5G are in the order of milliseconds or fractions thereof [36]. However, the time it takes for a recipient to decode a message – call it d – may be longer than that. For instance, it takes 4ms, i.e., four TTIs, for a 4G recipient to acknowledge receipt of a message [40]. Therefore, a round in our problem can be expected to be any multiple of the TTI large enough to allow for decoding.

We assume that the set of UEs to be targeted by a broadcast is known at the BS: this can be realized in practice by having these UEs subscribe to a specific service (e.g., “vehicular alert”) when they associate to the cell [32].

Cellular networks can leverage *location services* to acquire the updated position of the UEs under their control [2],[3]. This allows them to compute the distance between any two UEs, hence the probability of correct reception, given a model of the physical channel. Any position uncertainty due to inaccuracy of the location service can be factored in as a safety margin (i.e., increasing any inter-UE distance by twice the maximum error). UE mobility can be easily accounted for in a similar way. Given the maximum speed at which a UE can move, and a time interval during which our scheme is expected to complete, one can compute the maximum distance that a UE can cover and add it twice to any inter-UE distance. Now, a quick back-of-the-envelope computation shows that a car speeding at 200 km/h only moves by less than 2.5m in 40ms, a negligible distance with respect to the transmission radius of a D2D UE, which is in the region of 100m. Therefore, considering that our aim is to uphold deadlines of few tens of milliseconds, we are entitled to consider $P_{i,j}$ a *constant*, despite UE mobility. This entails neglecting *small-scale* fading effects. Work [31] and the references quoted therein show that this comes with negligible performance degradation.

Finally, we spend a few words to clarify that cellular networks have cell-wide broadcast transmissions, whereby the BS sends the same message to everyone that can hear it. This type of communication is meant for broadcast services (such as TV or radio), with periodic pattern [41]. Broadcast communications from the BS are scheduled in a quasi-static way (e.g., once on x rounds),

[1] In Section V we show that the current LTE-A bandwidth is amply sufficient to transmit the number of messages required by our scheme

and they cannot coexist with other cellular transmissions. Therefore, a low-latency, reactive cell-wide broadcast can only be obtained by setting a small period x, which reduces the available cellular capacity by $1/x$ [9]. Such a service cannot be used for on-demand, fast broadcasts, such as the ones required for sporadic critical events.

IV. Reliable Real-time Broadcast

In this section, we first formulate our problem as an optimization problem, show that it is infeasibly complex, and identify the reasons for such complexity. Then, we present our solution algorithm and analyze its properties. Table VI in Appendix B reports the definitions of all the symbols used in this paper.

A. *Formulation as an optimization problem*

Based on the discussion in the previous section, the problem can be formulated as follows:

$$\begin{array}{lll}\min \sum_{i=1}^{N} x_i + \frac{1}{k\cdot N^2}\cdot \sum_{t=1}^{k}\sum_{i=1}^{N} g_i^{(t)} & & \\ s.t. & & \\ m_i^{(0)} = x_i & \forall i & (i) \\ m_i^{(k)} \geq \alpha & \forall i & (ii) \\ 1 - m_j^{(t)} = & \forall j, \forall t & (iii) \\ \quad = \left(1 - m_j^{(t-1)}\right)\cdot \prod_{(i,j)\in\mathcal{E}} \left(1 - m_i^{(t-1)}\cdot P_{i,j}\right)^{g_i^{(t)}} & & \\ x_i \in \{0,1\} & \forall j & (iv) \\ g_i^{(t)} \in \mathbb{Z}^+ & \forall i, \forall t & (v) \\ m_i^{(t)} \in [0,1] & \forall i, \forall t & (vi)\end{array} \tag{3}$$

The objective function finds a starting set that minimizes $sum(\boldsymbol{X})$, and – for the same $sum(\boldsymbol{X})$ – the matrix $\boldsymbol{G}$ with a minimal $sum(\boldsymbol{G})$. Note that dividing $sum(\boldsymbol{G})$ by $k\cdot N^2$ in the objective function enforces strict priority of the first addendum over the second one.

Constraints (i) and (ii) are the starting and termination conditions, respectively, and (iii) regulates the update of probabilities.

Problem (3) is a MINLP, with $O(N\cdot k)$ discrete variables, and as many continuous ones. Constraint (iii) is however nonconvex. Note that even transforming it to its logarithmic equivalent, i.e.:

$$\log\left(1 - m_j^{(t)}\right) = \log\left(1 - m_j^{(t-1)}\right) + \sum_{(i,j)\in\mathcal{E}} g_i^{(t)}\cdot \log\left(1 - m_j^{(t-1)}\cdot P_{i,j}\right) \tag{4}$$

brings no benefits, since the rightmost logarithm cannot be simplified further. The above problem cannot be solved in reasonable times even at small scales. For instance, the BARON MINLP solver [14] cannot find the optimum after several minutes of computations when $N = 20$ [15]. Instead, our goal is to compute solutions: i) for larger scales (e.g., one order of magnitude), and ii) within milliseconds at most. In fact, both the above are requirements of current cellular networks.

It is an interesting fact that the toughness of (3) is due to the conjunction of *probabilistic reception* and *multihop*. In fact, removing either makes the problem much simpler. If we assume that graph $\mathcal{G}$ is *deterministic*, i.e., D2D transmissions are either perfectly reliable (in which case an edge exists between two vertexes) or impossible, the above problem becomes a MILP. MILPs are still $\mathcal{NP}$-hard, strictly speaking, but are considerably easier to solve. In Appendix A we show a heuristic that works quite fast in such a deterministic environment and discuss why it is unsuitable for our problem.

As for multihop, instead, we now show that problem (3) is simpler if we assume *one-hop* transmissions, i.e. if $k = 1$, since this will be relevant to our solution approach. The problem of finding the minimum number of initial message recipients, such that every UE will get the message via *one* extra hop, can in fact be formulated as follows:

$$\begin{array}{lll}\min \sum_{i=1}^{N} x_i & & \\ s.t. & & \\ \sum_{(i,j)\in\mathcal{E}} x_i\cdot \log\left(1 - P_{ij}\right) \leq \log(1-\alpha) & \forall j & (i) \\ x_j \in \{0,1\} & \forall j & (ii)\end{array} \tag{5}$$

Problem (5) is a *dominating set problem*, and is integer-linear. Although still $\mathcal{NP}$-hard, it admits fast polynomial heuristic solutions, based on LP-relaxation and rounding of binary variables [16]. Consider now the state of $\mathcal{G}$ after the vertices computed by (5) have been seeded with the message via downlink transmissions, and let $m_j^{(0)} = x_j$. Now, the next decision is who gets a D2D grant at round 1, i.e., $g_j^{(1)}$, knowing $m_j^{(0)}$. While $g_j^{(1)} = x_j$ is certainly a feasible solution, it can be wasteful, since there could be isolated vertices for which $x_j = 1$ (there being no other option than a DL transmission to reach them), and these do not need a D2D grant. However, the problem of finding $g_j^{(1)}$ is again similar to (5), i.e.:

$$\begin{array}{lll}\min \sum_{i=1}^{N} g_i^{(1)} & & \\ s.t. & & \\ \sum_{i\neq j} g_i^{(1)}\cdot \log\left(1 - m_i^{(0)}\cdot P_{i,j}\right) \leq \log(1-\alpha) & \forall j & (i) \\ g_j^{(1)} \in \{0,1\} & \forall j & (ii)\end{array} \tag{6}$$

Solving (5) and (6) at optimality yields a feasible solution to (3) when $k = 1$. Note that $g_j^{(1)} = x_j$ is itself a feasible solution, hence assuming $g_j^{(1)} \in \{0,1\}$ does not make the problem infeasible. In the next subsection, we exploit this insight to design a solution scheme that works in the general case $k \geq 1$.

B. *Heuristic solution algorithm*

We first present our heuristic informally, with the aid of a toy example, and then we formalize it later.

A significant complication with having *both* probabilistic connectivity *and* multihop routing (with maximum path length constraints) simultaneously is that it is inherently difficult to define *hop distances* between two nodes. Strictly speaking, any two vertices h, k in $\mathcal{V}$ have a non-null probability to communicate, hence they are one-hop neighbors connected by an edge. However, short paths are useless if they warrant a negligible probability of correct reception: a longer path, up to k hops, would certainly be preferable, if it is more reliable. Define the *reliability* of path p as $r_p = \prod_{(h,k)\in p} P_{h,k}$. Finding the highest-reliability path of up to k hops is relatively easy, since – although probabilities are not additive – their logarithms are. Therefore, given two vertices i and

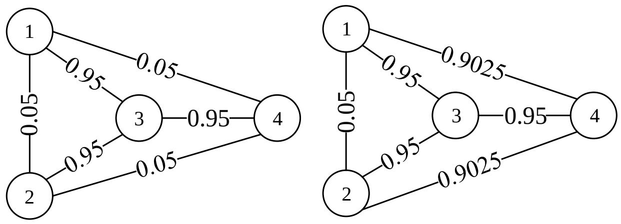

Figure 2 – A graph $\mathcal{G}$ (left) and the corresponding $\mathcal{Q}^{(2)}$ (right).

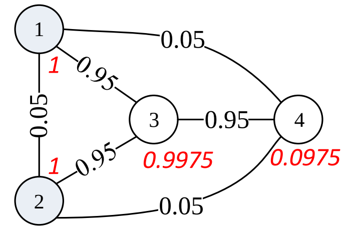

Figure 3 – State of $\mathcal{G}$ after round 1. The numbers in italics besides the vertices are probabilities $\boldsymbol{m}^{(1)}$.

j, the path p connecting them, that has at most k hops and minimizes cost $c_p = \sum_{(h,k)\in p} -\log(P_{h,k})$, can be found by running Bellmann-Ford (BF) algorithm on $\mathcal{G}$ [2]. Call $r_{i,j}^{(k)}$ the reliability of the *k-hop-constrained max-reliability* path between i and j (*k-mr* path for short, henceforth).

Given the *k-mr* path between every pair of vertexes, one can construct a *k-hop reliability closure of* $\mathcal{G}$, $\mathcal{Q}^{(k)} = \{\mathcal{V}, \mathfrak{R}^{(k)}\}$, i.e. a graph where each pair of vertexes (i,j) in $\mathcal{G}$ are directly connected by an edge labeled with reliability $r_{i,j}^{(k)}$. Figure 2 reports a simple $\mathcal{G}$ and its corresponding $\mathcal{Q}^{(2)}$, where both the 2-mr paths from 1 to 4 and from 2 to 4 traverse node 3.

Then, we solve a minimum set covering *on* $\mathcal{Q}^{(k)}$, to find a set of initial starting points, i.e. we solve problem (5) substituting $r_{i,j}^{(k)}$ for P_{ij}. By construction, any feasible solution $\boldsymbol{X}$ of this problem yields a set of starting points from which all the vertexes can be reached *within k hops, with total reliability no smaller than* α. In fact, it is easy to construct a feasible grant matrix $\boldsymbol{G}$ that will complete the schedule. This can be done as follows:

- Column 1, call it $\boldsymbol{g}^{(1)}$, can be set equal to $\boldsymbol{X}$.
- Column t, $\boldsymbol{g}^{(t)}$, with $t > 1$, is populated incrementing by one every entry $g_j^{(2)}$ whenever j appears as the t^{th} hop of a k-mr path[3].

In our toy example, solving (5) on $\mathcal{Q}^{(2)}$, with $\alpha = 0.95$, yields:

$$\boldsymbol{X} = \begin{bmatrix}1\\1\\0\\0\end{bmatrix}. \qquad (7)$$

In fact, vertex 4 cannot be reached from either 1 or 2 *alone* with $\alpha = 0.95$, but it can from *both*. From $\boldsymbol{X}$, the following feasible schedule can be computed via the rules discussed above:

$$\boldsymbol{G} = \begin{bmatrix}1 & 0\\1 & 0\\0 & 2\\0 & 0\end{bmatrix}. \qquad (8)$$

Note that, in the original graph $\mathcal{G}$, messages will also propagate through non-max-rel paths automatically, thanks to the broadcast nature of D2D transmissions. This multipath propagation *increases* the total reliability at the destinations, and is not counted in when solving (5) on $\mathcal{Q}^{(k)}$. This implies that, in general, any feasible solution of the latter (including the optimal one) may get *more* starting points than strictly necessary. Note that this does not happen in our example, where neither starting point can be removed without violating the reliability constraint.

Now, we can solve problem (6) on $\mathcal{Q}^{(k)}$ to obtain the recipients of the first round of D2D grants $g_j^{(1)}$. In our example, it will be $\boldsymbol{g}^{(1)} = \boldsymbol{X}$. We can then predict the effects of a round of grants $\boldsymbol{g}^{(1)}$ by applying (1), i.e. we can compute vector $\boldsymbol{m}^{(1)}$. Figure 3 reports vector $\boldsymbol{m}^{(1)}$ directly in the graph.

Note that using (1) leverages multipath diffusion along *non max-rel paths* too. This has an important consequence, which can be capitalized to design optimized schedules: if $g_i^{(1)} \geq 1$, then for any vertex $h \in \mathcal{V}$, $m_h^{(1)} \geq m_i^{(0)} \cdot P_{i,h}$, where equality holds only if $P_{j,h} = 0$ for $j \neq i$. This inequality holds also when h is the second vertex in a k-mr path. For instance, in Figure 3, $m_3^{(1)} = 0.9975$, which is larger than either $m_1^{(0)} \cdot P_{1,3} = 0.95$ or $m_2^{(0)} \cdot P_{2,3} = 0.95$.

This means that, starting from the next (second) round, the message will be *more likely* to reach its destinations than we had predicted when solving (6) on $\mathcal{Q}^{(k)}$. With reference to the example of Figure 3, the problem that we need to solve now is how to reach 4 with the required reliability (1, 2, and 3 being already over the 0.95 threshold), by scheduling a second round of D2D transmissions. This problem can be formulated as follows:

$$\begin{aligned}
&\min \sum_{i=1}^{N} g_i^{(2)} \\
&s.t. \\
&\sum_{i\neq j} g_i^{(2)} \cdot \log\left(1 - m_i^{(1)} \cdot r_{i,j}^{(1)}\right) \leq \log(1-\alpha) \;\; \forall j \;\; (i) \\
&g_j^{(1)} \in \mathbb{Z}^+ \qquad\qquad\qquad\qquad\qquad\qquad \forall j \;\; (ii)
\end{aligned}, \qquad (9)$$

which is again of the same form as (6). The crucial difference is that now we can rely on 1-*mr* paths, since we have only one last round (and it is obviously $r_{i,j}^{(1)} = P_{i,j}$), hence we are solving the same problem on the 1-hop reliability closure of $\mathcal{G}$, call it $\mathcal{Q}^{(1)}$, where $\mathcal{Q}^{(1)} \equiv \mathcal{G}$. The alert reader can check that solving (9) at optimality yields the following solution:

[2] Note that using Dijkstra's algorithm allows you to find the *maximum reliability path* without any hop-count constraint. However, such a path is very likely to have unnecessarily many hops, hence will be unusable in practice, especially if the network is very dense. In that case, in fact, Dijkstra will systematically violate the k-hop constraint to maximize the reliability, inserting many unnecessary reliability-preserving micro-hops between adjacent UEs.

[3] Some optimizations are possible, such as leveraging the fact that a k-mr path to a "near" vertex may well form the initial segment of a longer k-mr path to a "far" vertex, in which case the same node can be counted once instead of twice. We leave these optimizations to the interested reader, whom we warn that they are not straightforward, since the set of k-mr paths radiating from a starting point does not form a tree. Here, we are only preoccupied to show that *a feasible* schedule exists. We will discuss *good*, optimized schedules later.

$$\boldsymbol{g}^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{10}$$

With reference to Figure 2, in fact, vertex 4 was meant to be reached through *both* 2-*mr* paths 1-3-4 and 2-3-4. This was implied when selecting *both* 1 and 2 as starting points. Since these two paths are not disjoint, sticking to this initial plan would require giving *two* grants to 3 at round 2 (as per column 2 of (8)). However, given the situation pictured in Figure 3, a single grant at 3 will suffice. This confirms that, while one *could* figure out an entire feasible schedule $\boldsymbol{G}$ right after finding the set of initial points, that schedule would be suboptimal (i.e., include too many grants), since it would neglect multipath diffusion. More efficient schedules can be found by capitalizing multipath diffusion at the first hop through (1).

For a larger value of k, our algorithm will repeat the same basic steps, only going through more iterations. It will first compute the initial starting points by solving set covering (5) on $\mathcal{Q}^{(k)}$. Then, it will iterate for $t = 1$ to k rounds, each time updating the vertex probabilities on $\mathcal{G}$ via (1) and solving (6) on $\mathcal{Q}^{(k-t+1)}$ to compute $\boldsymbol{g}^{(t)}$. More formally, the algorithm is as follows:

Inputs: $k, \alpha, \boldsymbol{P}$.

Step 1:

- For $t = 1$ to k, compute the t-*mr* paths for all pairs of vertices (i, j) and the associated reliabilities $r_{i,j}^{(t)}$.

Step 2:

- Compute vector $\boldsymbol{X}$ by solving the following problem:

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{N} x_i \\ &s.t. \\ &\textstyle\sum_{i \neq j} x_i \cdot \log\left(1 - r_{i,j}^{(k)}\right) \leq \log(1-\alpha) \quad \forall j \quad (i) \\ &x_j \in \{0,1\} \quad \forall j \quad (ii) \end{aligned} \tag{11}$$

- Set $\boldsymbol{m}^{(0)} = \boldsymbol{X}$

Step 3:

For $t = 1$ to k, repeat the following:

- Compute column $\boldsymbol{g}^{(t)}$ of D2D grant matrix $\boldsymbol{G}$ by solving the following problem:

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{N} g_i^{(t)} \\ &s.t. \\ &\textstyle\sum_{i \neq j} g_i^{(t)} \cdot \log\left(1 - m_i^{(t-1)} \cdot r_{i,j}^{(k-t+1)}\right) \quad \forall j \; (i) \\ &\qquad \leq \log(1-\alpha) \\ &g_j^{(t)} \in \mathbb{Z}^+ \quad \forall j \; (ii) \end{aligned} \tag{12}$$

- Apply (1), i.e., set $\boldsymbol{m}^{(t)} = f\left(\boldsymbol{m}^{(t-1)}, \boldsymbol{g}^{(t)}, \boldsymbol{P}\right)$.

C. *Analysis of the solution algorithm*

Note that our algorithm does not guarantee that the k-*mr* path from a starting point i to a destination j will be followed. First, j's reliability may well exceed α without the contribution of the path starting at i. Second, even if that contribution was initially deemed necessary when solving (11), it may well happen that, after computing $m_j^{(t)}$ after a round t, this is found to be no longer the case – recall that our estimates are initially conservative, and they are revised upwards as the broadcast diffusion progresses. Third, a *different* path to j may become preferable, because its initial node is discovered to have a higher probability of possessing the message than anticipated, for the same reasons. This allows our algorithm to capitalize the effects of multipath diffusion of the message, reducing the number of grants in subsequent rounds, without paying the price of incorporating (1) in scheduling decisions.

Hereafter, we prove that our algorithm always finds a feasible schedule, i.e.:

Proposition 1: $\forall j, m_j^{(k)} \geq \alpha$.

Proof: Starting from a feasible solution of (11) and (12), one can compute a feasible schedule of k and $k - t + 1$ rounds respectively. In fact, feasible solutions of (11) and (12) consider only *mr* paths with a feasible number of hops, and propagating grants along those paths allows each destination to be reached with the required reliability (both conditions are embedded in constraints (i) of these problems). In fact, such a feasible schedule is the one where grants are given to all the subsequent vertexes along the max-rel paths that start at vertices for which $x_i = 1$ and $g_j^{(t)} \geq 1$, respectively. Therefore, we only need to prove that:

a) Problem (11) is always feasible;
b) The first iteration of (12), with $t = 1$, is always feasible;
c) If iteration t is feasible, so is iteration $t + 1$.

Points a) and b) are trivial, since $\boldsymbol{X} = \boldsymbol{e}$ is a feasible solution of (11), and $g_i^{(t)} = x_i$ is a feasible solution of (12) when $t = 1$. Point c) follows from the fact that, if $g_i^{(t)} \geq 1$, then $m_h^{(t)} \geq m_i^{(t-1)} \cdot P_{i,h}$, for any node $h \in \mathcal{V}$.

■

We now analyze the complexity of our algorithm. As far as Step 1 is concerned, the *mr* paths of up to k hops can be computed either using BF, or by computing $\boldsymbol{P}^{\boldsymbol{k}}$ using *tropical multiplications* [17]. Both have the same complexity, i.e., $O(k \cdot N^3)$, and both work by iterating on the path length, hence finding k-*mr* paths is as complex as finding all h-*mr* paths, $1 \leq h \leq k$. However, BF is generally faster, especially if $\boldsymbol{P}$ is sparse. Our $\boldsymbol{P}$ is certainly not sparse, unless $\mathcal{E}$ is pruned to remove all the links whose probability can safely be considered irrelevant (e.g., smaller than a threshold β with $\beta \approx 10^{-2}$). Such pruning reduces computation times considerably, at a negligible increase in the solution cost. In fact, on one hand, it is unlikely that schedules will rely on such poor links. On the other, edge-pruning is only needed in Step 1, whereas (1) can be computed using the original $\boldsymbol{P}$, so that even poor links contribute to the reliability. Another advantage of BF is that it is *totally parallelizable*: path computations starting from different nodes can run in parallel, allowing one to reap the full speed-up in a multicore environment.

Both steps 2 and 3 involve solving an ILP of $O(N)$ variables. Solving ILPs at optimality is $\mathcal{NP}$-hard in general. However, good heuristic solutions can be found in polynomial time. The heuristic that we use is based on LP relaxation, i.e., removing the integrality constraints in (11) and (12). LPs can be solved at optimality in polynomial time, with a complexity $O(N^3)$ [18]. The LP optimum is fractional, in general, hence we apply Randomized Rounding (RR) to obtain a logarithmic-factor approximation of the optimal solution to the original ILPs [16]. RR has a cost

Table I – Main simulation parameters

Parameter	Value
Carrier frequency	2 GHz
Bandwidth	20 MHz
D2D Tx Power	20 dBm
Noise figure	5 dB
Cable loss	2 dB
Gaussian Noise	-104.5 dB
Message length	10 bytes
D2D Modulation and coding	CQI 7
Mobility model	Static
Number of independent replicas	10

Table II – Average delivery ratio, 1000m radius and $\alpha = 0.95$

	k					
N	**1**	**2**	**3**	**5**	**7**	**9**
25	0.996	1	1	0.992	0.992	0.992
50	0.998	0.994	1	1	1	0.992
75	1	1	1	0.996	0.999	0.997
100	1	0.999	0.992	1	1	1
125	0.999	0.999	1	1	0.999	0.999
150	0.999	1	0.998	1	0.999	1
175	0.999	1	0.998	0.999	0.999	0.999
200	0.999	1	1	0.999	1	0.999

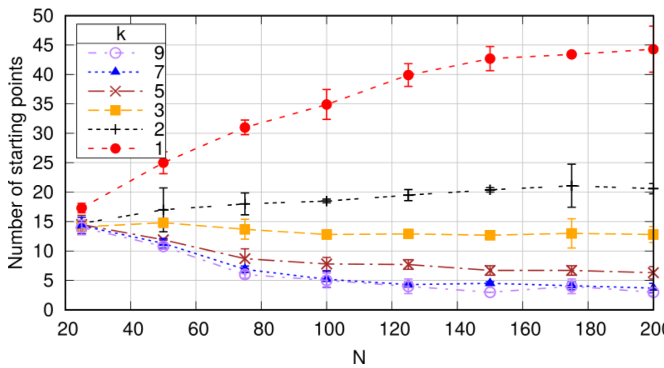

Figure 4 – Number of initial starting points, $\alpha = 0.95$

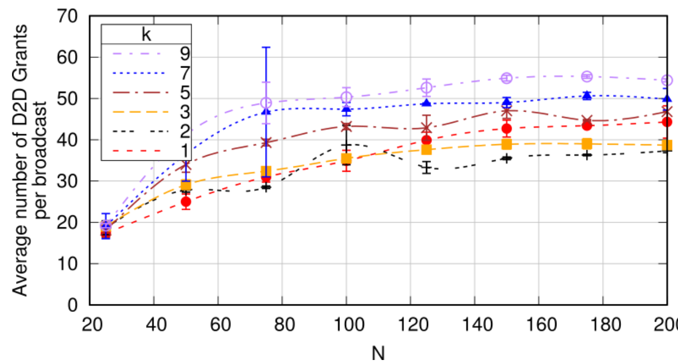

Figure 5 - Total number of grants, $\alpha = 0.95$

$O(N \log(4N))$.

Finally, the last bullet of steps 2 and 3, i.e., updating the vertices' probabilities through (1), is $O(|\mathcal{E}| + N) = O(N^2)$.

From the above, we can state the following:

Proposition 2: *Our algorithm is polynomial, and its complexity is $O(k \cdot N^3)$.*

Proof: The complexity of Step 1 is $O(k \cdot N^3)$. The complexity of Step 2 is $O(N^3)$, and the complexity of one of the *k* iterations of the *for* cycle in step 3 is $O(N^3)$, hence the thesis.

■

Big-O complexity provides a platform-independent way to assess the *scalability* of an algorithm. However, it says little about the orders of magnitude of the actual computation times involved. For this reason, in the next section, we complement the above analysis with an assessment of *computation times* on a desktop computer. Here, we limit ourselves to observing that, assuming that C_1, C_2, C_3 are the computation times for, respectively, Step 1, Step 2, and *a single for iteration* of Step 3 (assuming for simplicity that all iterations take the same time), the only time cost to be paid upfront before starting the message transmission is $C_1 + C_2$, since each problem (12) in Step 3 can be solved in parallel to a message transmission (a DL one when $t = 1$, a D2D one otherwise). Moreover, if $C_3 \leq d$, the computations of Step 3 do not delay the diffusion of the message. We remark that the proposed algorithm remains valid even if graph $\mathcal{G}$ is disconnected, i.e. at least one vertex of the graph is not reachable from any other vertex in at most k hops, with probability larger than α. As previously discussed for problem (5), such vertex is included in the set of starting points $\boldsymbol{X}$ when solving (11).

We terminate this subsection with a consideration on parameter k. It may well be the case that the value of k given as an input is larger than necessary, i.e., that it would be possible to compute a schedule of $h < k$ rounds having the same number of starting points and grants – e.g. because all the k-*mr* paths have at most h hops. To optimize broadcast diffusion, we can cap the *for* cycle in in Step 3 to the *effective number of hops* $k^* \leq k$, where k^* is the maximum length among all the k-mr paths.

V. Performance Evaluation

In this section, we assess the performance of our algorithm via simulation. We first describe the simulation tool, then we analyze a scenario where UEs are randomly distributed within a circular target area, according to a uniform distribution. In this scenario, we relate the performance of our algorithm to key parameters k and α, discussing computation times for each of the algorithm steps. We also quantify the cost of *not knowing* which UEs have the message on each round, by comparing our algorithm to an *omniscient* version of it. Moreover, we assess the performance of our algorithm in a 3GPP urban scenario for vehicular communications. Finally, we compare our approach with the schemes proposed in [8] and [9].

In order to perform the above analyses, we developed an ad-hoc simulation tool that drops N UEs on a 2D-floorplan and computes matrix $\boldsymbol{P}$. Each entry $P_{i,j}$ is computed as follows: the UEs transmission power is fixed and known at the BS. The reception power at j is computed by subtracting the path loss between i and j, which is a function of their distance, according to models proposed in [19]. The received power is used to compute the Signal to Noise Ratio (SNR). From the latter, using Block Error Rate (BLER) curves, one can derive the probability of correct reception of a single transmission block, and – from the latter – the one for a message occupying more than one block. We use BLER curves taken from SimuLTE [20], a popular and accurate 4G simulator. Unless stated otherwise, the results are obtained by averaging the metrics from ten independent replicas of the simulation.

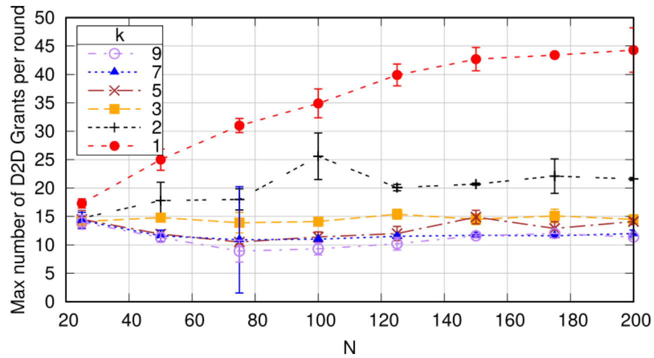

Figure 6 - Maximum number of grants per round, $\alpha = 0.95$

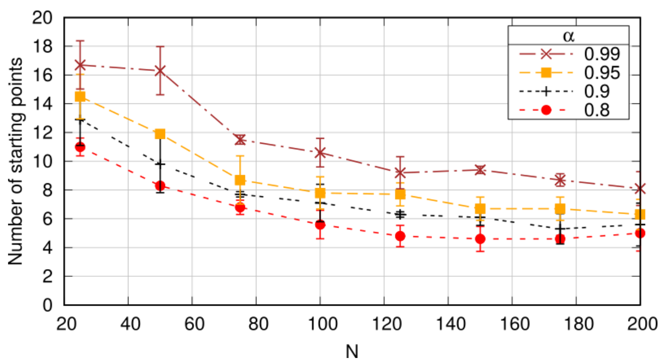

Figure 7 - Number of initial starting points with different values of α

Table III – Average delivery ratio, 1000m radius and k=5

	α			
N	**0.8**	**0.9**	**0.95**	**0.99**
25	0.984	0.992	0.992	1
50	0.99	0.992	1	1
75	0.993	0.996	0.996	1
100	0.994	0.995	1	1
125	0.993	0.998	1	1
150	0.991	0.993	1	1
175	0.994	0.999	0.999	1
200	0.995	0.998	1	0.999

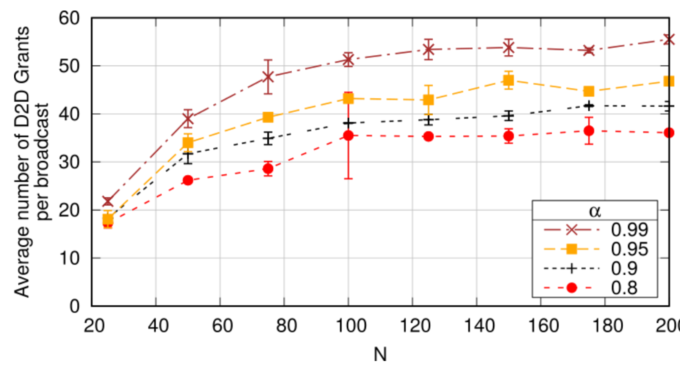

Figure 8 - Total number of D2D grants with different values of α

Graphs report confidence intervals at 95%. The main simulation parameters are shown in Table I. Channel parameters are taken from [19]. The length of messages is set to 10 bytes, assuming that the message includes two 32-bit floating points for the coordinates (i.e. latitude and longitude) of the critical event location, plus one 16-bit integer to store the radius of the target area where the message must be disseminated [8]. CQI is set to 7, as a tradeoff between a conservative modulation with high overhead (e.g. CQI 2) and an efficient modulation with higher loss probability (e.g. CQI 15). This is shown to be the optimal CQI for that message length in [8].

A. Uniform scenario

We deploy UEs uniformly in a circle with 1000m radius. In each experiment, the position of the whole set of UEs is randomly selected. We assess the performance and the computation cost when varying the number of UEs and the algorithm requirements

1) Varying the maximum number of hops

We first vary the maximum number of hops k, while fixing the reliability threshold α at 95%. We vary the number of UEs deployed in the area.

Table II reports the delivery ratio, i.e. the fraction of UEs that received the message by the k-th hop, computed *a posteriori* via simulation. Note that this measure is *not* value $m_j^{(k)}$, which is computed based on *a priori* probabilistic estimates of the delivery at UE i. Confidence intervals are negligible and omitted for ease of reading. All values are above 99%. Figure 4 and Figure 5 show, respectively, the number of starting points and the total number of D2D grants required to complete the broadcast. The number of starting points is relatively large when N is small. This is obvious since the probability of reaching a neighbor increases with the density, hence at low densities many UEs can only be targeted via DL transmissions. When the UE density increases, the number of starting points becomes smaller, especially when $k > 2$, since more UEs can be reached via multihop D2D transmissions. Clearly, this comes at the price of issuing more D2D grants. The curve with $k = 1$ represents the case where only one D2D hop is allowed. In this case, the number of starting points can only increase with the total number of UEs. However, it is clear from both graphs that the total number of transmissions (i.e., DL ones to seed the starting points, plus D2D ones) is smaller than the number of UEs. A network with $N = 200$ UEs can be covered with 54 transmissions (see $k = 3$), i.e. 27% of those that would be required by addressing each UE via the DL. Note that we compute the number of *D2D grants*, which is an upper bound to the number of *D2D transmissions*. In fact, some UEs may be given a grant when they do not have the message (but the BS estimates that they do).

As far as schedule feasibility is concerned, Figure 6 reports the *maximum* number of D2D transmissions per round that our algorithm requires. The maximum number is in the order of few tens, which is well within the capabilities of current 4G and 5G networks. In fact, a 20MHz LTE-A (4G) deployment can already support the simultaneous transmission of 50 messages per round.

2) Varying the reliability threshold

We fix $k = 5$ and evaluate the same metrics when the reliability threshold α changes. Table III reports the delivery ratio at the end of the broadcast for different values of α. We observe that the resulting delivery ratio is above 99% in almost any case, also when α is far lower than this value. This shows that our solution is conservative, and this is due to using broadcast (D2D) communications that allow one to exploit the contribution of non-maxrel paths.

Figure 7 and Figure 8 show the number of starting points and the total number of D2D grants issued during the broadcast. As

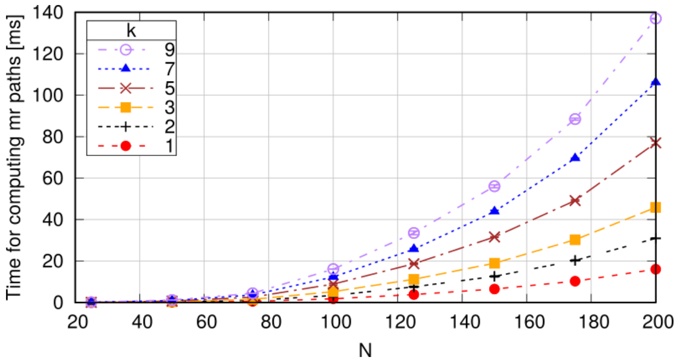

Figure 13 – Average time to compute k-mr paths

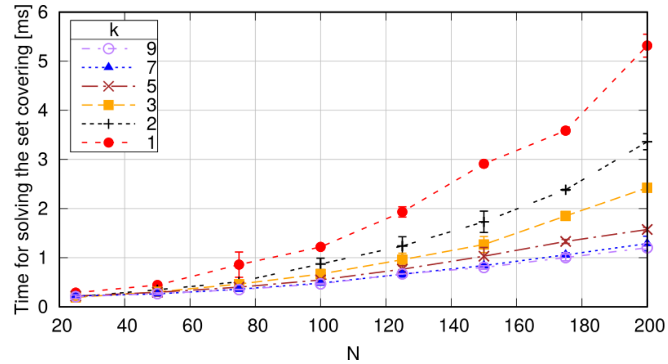

Figure 14 – Average solving time of problems (11)

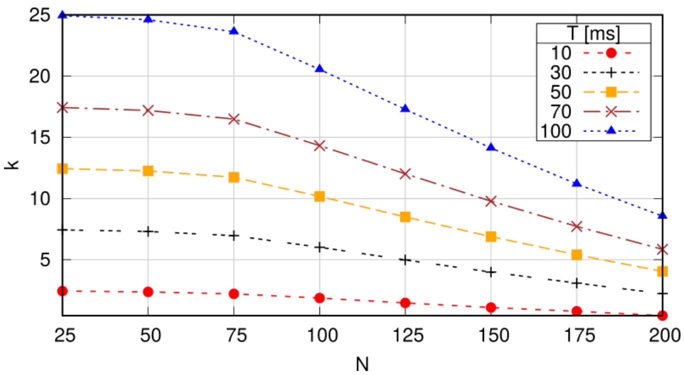

Figure 9 – Feasible regions of N, k for different deadlines T

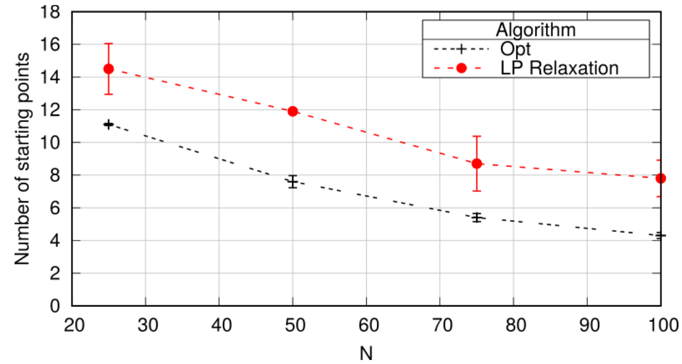

Figure 10 – Initial starting points when solving Step 2 at optimality or heuristically through LP relaxation and rounding

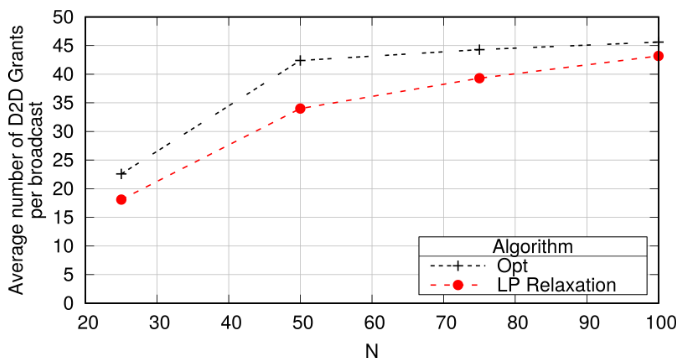

Figure 11 – Total number of D2D grants when solving Step 3 at optimality or heuristically through LP relaxation and rounding

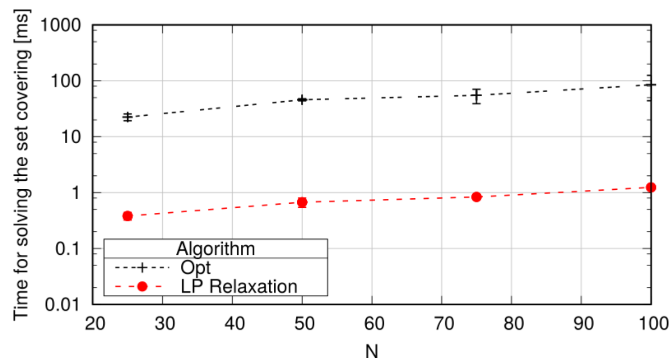

Figure 12 – Average solving time for the set covering problem, optimal formulation vs. LP relaxation

expected, all the curves have the same trend and both the number of starting points and D2D grants increase with α.

3) Computation cost

We now complement the complexity analysis of Section IV with a measurement of the actual computation times of our algorithm. In particular, we measure the time required to compute the *k-mr* paths and to solve the optimization problems. The computation times of the other parts of the algorithm (e.g., randomized rounding, computing (1)) are negligible. The measurements are taken on a desktop computer equipped with an Intel(R) Core(TM) i7 CPU at 3.60 GHz, 16 GB of RAM and a Kubuntu 16.04 operating system.

Figure 13 shows the time required to compute the all-pairs *q-mr* paths, for all path lengths q such that $1 \le q \le k$, using BF. Such values are obtained by repeatedly executing BF algorithm in a single-threaded process. However, as discussed in Section IV, BF can be fully parallelized. If the computations for N nodes are run on a machine with N_c cores, the total computation time required is the one read on the y axis divided by N_c. For instance, path computation with $k = 9$ and $N = 200$ on an 8-core machine takes around 17 ms.

Figure 14 reports the time required by CPLEX [21] to solve at optimality the LP relaxations of the problems for steps 2 and 3 of our algorithms (the solution times of the two problems are indistinguishable). We observe that the solving times are in the order of few milliseconds, i.e., compatible with decoding times d. Moreover, solving times decrease with k, i.e. allowing multihopping makes these problems *less* costly, because their constraint *(i)* can be met more easily (in other words, a *k-mr* path with $r_{i,j}^{(k)} > \alpha$ for any pair i, j exists with higher probability).

From the above, we can identify the range of feasible values of k given the number of UEs N and the deadline T, factoring in the cost of computations. The total time to complete a broadcast is $D = C_1(N)/N_c + C_2(N) + k \cdot \max\{C_3(N), d\}$, where $C_1(N)$ is taken from Figure 13, and $C_2(N) \cong C_3(N)$ from Figure 14. Based on the latter, Figure 9 shows the extrapolated loci $D = T$ on the (N, k) plane, for $N_c = 2$. The region below each locus includes points (N, k) for which $D < T$, hence feasible from a

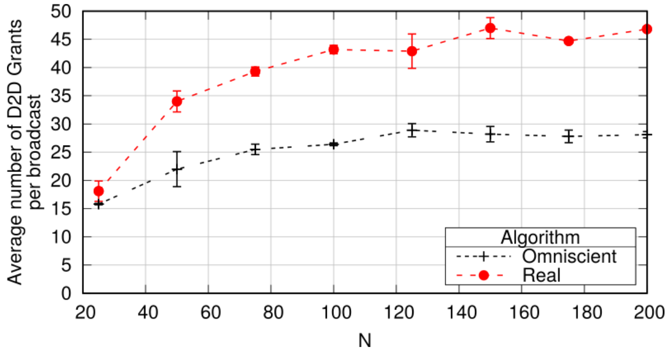

Figure 15 – Total number of D2D grants, real vs. omniscient algorithm

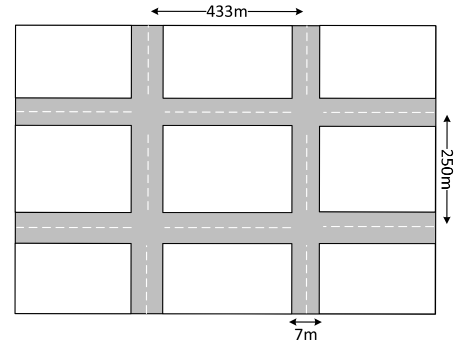

Figure 16 – Urban grid scenario

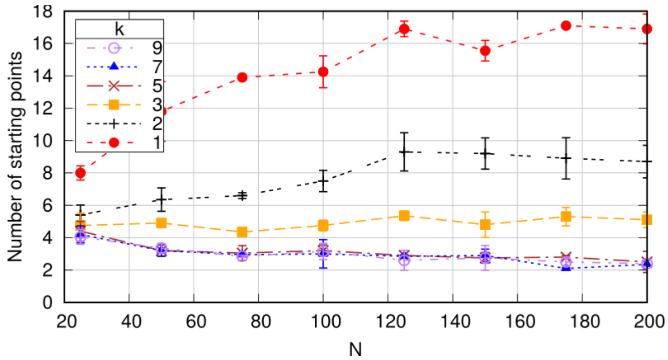

Figure 17 – Number of initial starting points, urban grid scenario

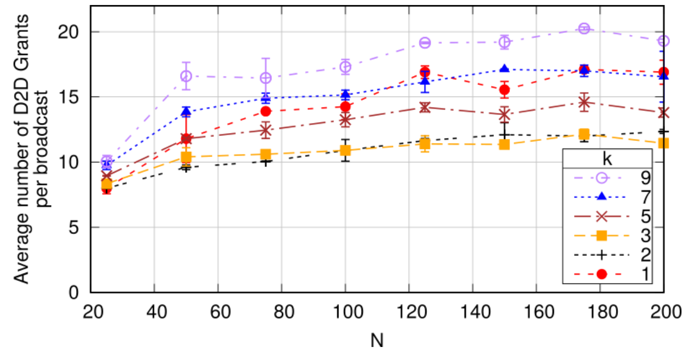

Figure 18 – Total number of D2D grants, urban grid scenario

Table IV – Comparison with an omniscient version of the algorithm

	***A-posteriori* delivery ratio**		***A-priori* fraction of nodes for which $m_i^{(k)} \geq \alpha$**	
N	**Real**	**Omniscient**	**Real**	**Omniscient**
25	0.992	1	1	0.916
50	1	1	1	0.69
75	0.996	1	1	0.692
100	1	1	1	0.634
125	1	1	1	0.671
150	1	1	1	0.566
175	0.999	1	1	0.583
200	0.999	1	1	0.54

latency standpoint. Although the curves in Figure 9 are scenario-dependent, they show that it is actually possible to allow several hops of relaying, even for a large N, unless the deadline is very strict (e.g., 10ms). For instance, if the deadline is $T = 50ms$, in a network with $N = 200$, we can afford up to four relaying hops.

4) Optimality of LP relaxations

We now discuss the optimality of the LP relaxation of problem (11). We fix $k = 5$ and α=0.95. Figure 10 shows the number of starting points selected by the two formulations of the problem, which is also the value of the objective function. The results show that the LP relaxation finds more starting points than the optimal formulation. While this is suboptimal for the operator, it has a mitigating side effect. Figure 11 reports the total number of transmissions, and we observe that fewer D2D grants are used later, *because* more starting points have been selected initially.

In any case, solving the problem at optimality is not feasible at the required timescales, even for small values of N, as shown in Figure 12. In fact, more than 20ms are required when $N = 25$, against less than 0.4ms for the relaxed problem. For $N = 100$, solving (11) at optimality takes 100 times as much as solving its LP relaxation.

5) Assessing the cost of probabilistic assumptions

We now assess the cost of relying on probabilistic assumptions about which UEs have the message during the broadcast. We do this by comparing our algorithm against an *omniscient* version of it. In the latter, message reception is still stochastic, but the BS *knows* which UEs got the message after each round. Note that this is *not* possible in the current 4G and 5G standards, since there is no control channel that reports to the BS the reception of D2D broadcast messages.

We vary N and fix $k = 5$ and $\alpha = 0.95$. The leftmost two columns of Table IV report the percentage of UEs that received the message at the end of the broadcast, measured *a posteriori*. Note that the omniscient version of the algorithm terminates when *all* UEs have the message, i.e., possibly sooner than the k^{th} round. In both cases, almost all UEs correctly receive the message. The rightmost two columns, instead, show the fraction of UEs for which $m_j^{(k)} \geq \alpha$, i.e., for which the *a-priori* estimate is large enough. As expected, with the real algorithm this inequality holds for all UEs. Interestingly, the fraction of UEs that verify it in the omniscient version is *far smaller* than α. In other words, the omniscient version counts in UEs that can be found to possess the message *a posteriori*, but cannot be supposed to have it *a priori* with enough reliability. This confirms that our algorithm assigns grants conservatively. Leveraging *a-posteriori* knowledge on each round allows the omniscient algorithm to issue D2D grants to UEs that would never be scheduled otherwise, hence completing the broadcast sooner and with fewer transmissions (roughly half as many, as shown in Figure 15). Besides providing insight on the workings of our algorithm, the above analysis shows that the latter could also be used in a technology where reception is stochastic, but the outcome of transmissions can be inferred by

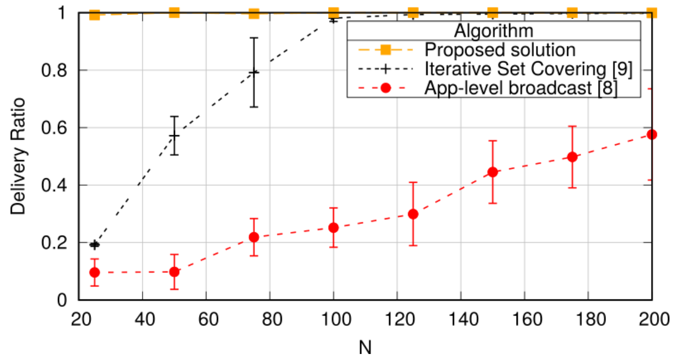

Figure 19 – Delivery ratio, proposed solution vs. [8] and [9]

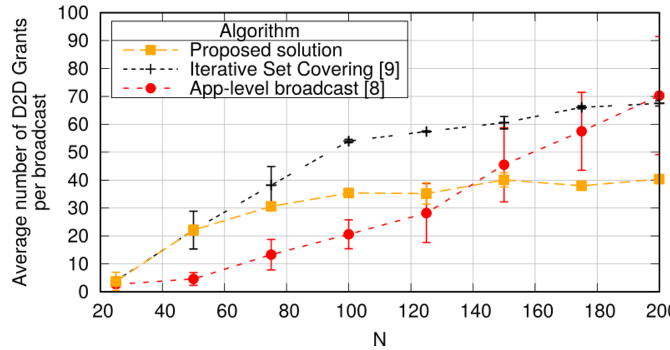

Figure 20 – Total number of D2D grants per broadcast, proposed solution vs. [8] and [9]

the scheduling entity.

B. Urban grid scenario

We now consider an urban scenario taken from the 3GPP specifications [22] and depicted in Figure 16. The latter is composed of four streets forming a grid of size $433m \times 250m$. Streets are $7m$ wide and have two lanes, one per direction. We deploy N UEs (i.e., connected cars) along the streets in a uniform fashion, increasing N to simulate increased traffic density. The graphs show metrics averaged from twenty independent replicas of the simulation, with confidence intervals at 95%.

Figure 17 shows the number of starting points for different values of k. Note that a direct comparison with the uniform scenario would be unfair, since the target area is smaller here. We observe that few starting points are required. This is because UEs' positions are constrained by the streets configuration, making it is less likely that a UE is isolated. Moreover, Figure 18 shows that the total number of D2D grants required for completing the broadcasting is small, too. In fact, the algorithm can select the starting points wisely, i.e. among UEs located close to intersections, from which it is easy to reach all the other UEs. Overall, our scheme allows one to reach 200 UEs with 12 transmissions (counting in both DL and D2D ones), i.e. 6% of those that would be required using DL transmissions only, when $k = 3$.

C. Comparison with other schemes

Finally, we compare our work against the schemes in [8] and [9]. Work [8] describes a *distributed* scheme that can run on 4G networks. Each UE that receives the message acts as a relay, using the Trickle suppression mechanism [23] to avoid network flooding. Collision-free scheduling is guaranteed by the BS. The latter, however, does not make schedules: UEs that want to relay the message just request a grant to it, and the BS responds by scheduling D2D grants according to the available resources. Our previous work [9] is instead the first to involve the BS into the scheduling process: it is proposed therein that the BS iteratively selects UEs with a "high" probability of having the message, computes their one-hop neighborhood, and computes the minimum set of relay UEs that can cover it.

We run the comparison in a scenario where UEs are uniformly distributed within a 1000m-radius circle. Both [8] and [9] deal with data dissemination starting from a single source. Thus, in order to perform the fairest possible comparison, we force one UE to be located exactly in the center of the circle and consider it as the starting point for the broadcasting in schemes [8] and [9].

We fix $k = 5$ and $\alpha = 0.95$. Since in [8] the maximum number of relaying hops is carried in the message as a Time-to-live (TTL) field, we set the latter equal to k [8]. The TTL is set by the source and decreased by one at every relaying hop. UEs that receive a message with a TTL=1 do not relay it.

Figure 19 shows the delivery ratio against N. Schemes [8] and [9] are sensitive to the UE density, and with a small N they fail to reach all the UEs. On the other hand, our algorithm guarantees that all UEs receive the message, regardless of the UE density. In fact, isolated UEs can be targeted by DL transmissions from the BS. Figure 20 reports the total number of D2D grants. The distributed scheduling proposed in [8] uses fewer transmissions when N is small because it can reach only a small fraction of UEs, hence relaying cannot be performed. As soon as the UE density increases and more UEs receive the message, the number of D2D grants overcomes the one obtained with our algorithm. This is because relay suppression is delegated to the applications running on the UEs, which only possess a local view of the status of the broadcast. Scheme [9] uses the same number of D2D grants as our algorithm at low densities, where the delivery ratio is small. However, our algorithm outperforms [9] when delivery ratios get comparable, due to the greedy nature of the latter, which tries to maximize the number of receiving UEs at each hop.

VI. CONCLUSIONS

In this paper we proposed a centralized scheme to diffuse a message in a D2D-enabled cellular network, by scheduling broadcast D2D transmissions, so that the message reaches all its intended targets with a pre-specified probability and within a given time. Our scheme minimizes the number of initial UEs to be seeded via the DL, and the number of D2D grants required. To the best of our knowledge, this is the first and only scheme that achieves this objective. We showed that the problem is mixed-integer-non-linear and non-convex, hence unsolvable in practice, and proposed a polynomial approximation which is based on computing maximum-reliability paths and solving per-hop problems iteratively to find the sets of grant recipients. We showed that the problem is computationally feasible, i.e. can be solved online on off-the-shelf hardware at reasonably large scales. Moreover, we showed that our scheme requires a small number of initial starting points and a small, sublinear number of D2D grants, making it considerably more efficient that targeting UEs individually.

At the time of writing, we are investigating if the original non-convex problem admits tighter approximations, possibly at the price of different orders of magnitude in the computation time. This will help us to benchmark the optimality of our scheme.

Acknowledgments

This work was partially supported by: i) the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence), and ii) the University of Pisa, through grant "*Analisi di reti complesse: dalla teoria alle applicazioni*" - PRA 2020.

References

[1] A. Asadi, Q. 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, doi: 10.1109/COMST.2014.2319555.

[2] 3GPP TS 38.305 v15.5.0, "NG Radio Access Network (NG-RAN); Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN", December 2019

[3] ETSI GS MEC 013 v2.1.1 (2019-09) "Multi-access Edge Computing (MEC); Location API", September 2019

[4] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao and A. Srinivasan, "Mobile Data Offloading through Opportunistic Communications and Social Participation," in IEEE Transactions on Mobile Computing, vol. 11, no. 5, pp. 821-834, May 2012. doi: 10.1109/TMC.2011.101

[5] Y. Li, M. Qian, D. Jin, P. Hui, Z. Wang and S. Chen, "Multiple Mobile Data Offloading Through Disruption Tolerant Networks," in IEEE Transactions on Mobile Computing, vol. 13, no. 7, pp. 1579-1596, July 2014. doi: 10.1109/TMC.2013.61

[6] Y. Liu, A. M. A. E. Bashar, Fan Li, Y. Wang and Kun Liu, "Multi-copy data dissemination with probabilistic delay constraint in mobile opportunistic device-to-device networks", in proceedings of IEEE WoWMoM 2016, Coimbra (PT), 2016, pp. 1-9. doi: 10.1109/WoWMoM.2016.7523548

[7] W. Gao, Q. Li, B. Zhao and G. Cao, "Social-Aware Multicast in Disruption-Tolerant Networks," in IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1553-1566, Oct. 2012. doi: 10.1109/TNET.2012.2183643

[8] G. Nardini, G. Stea, A. Virdis, "A Fast and Reliable Broadcast Service for LTE-Advanced Exploiting Multihop Device-to-Device Transmissions", in Future Internet 2017, 9, 89. doi: 10.3390/fi9040089

[9] G. Nardini, G. Stea, A. Virdis, "Geofenced broadcasts via centralized scheduling of device-to-device communications in LTE-Advanced", in Springer Communications in Computer and Information Science (CCIS), vol. 825, ISBN 978-3-319-91631-6, 2018

[10] P. Heggernes and D. Lokshtanov, "Optimal broadcast domination in polynomial time", in Discrete Mathematics, 306 (24): 3267–3280, 2006.

[11] M. A. Henning, Ortrud R. Oellermann, and Henda C. Swart, "Bounds on distance domination parameters", in Journal of Combinatorics, Information and System Sciences, 16(1):11–18, 1991

[12] B. Kaufman and B. Aazhang, "Cellular networks with an overlaid device to device network", in proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2008, pp. 1537-1541, doi: 10.1109/ACSSC.2008.5074679.

[13] F. S. Shaikh and R. Wismüller, "Routing in Multi-Hop Cellular Device-to-Device (D2D) Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2622-2657, 4th quarter 2018. doi: 10.1109/COMST.2018.2848108

[14] NEOS solvers website, https://neos-server.org/neos/solvers/index.html, accessed May 4, 2019

[15] M. Cacciola, A. Frangioni, L. Galli, G. Stea, "A Lagrangian approach to Chance Constrained Routing with Local Broadcast", in proceedings of 18th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Ischia, Italy, June 15-17, 2020

[16] V. Vazirani, "Approximation algorithms", Springer Verlag. ISBN 978-3-540-65367-7.

[17] B. A. Carré, "An algebra for network routing problems", in IMA Journal of Applied Mathematics 7 (1971), 273-294

[18] K. M. Anstreicher, "Linear Programming in $O(n^3/log(n) \cdot L)$ operations", in SIAM Journal of Optimization, vol. 9, no. 4, pp. 803-812, 1999

[19] 3GPP TR 36.814 v9.0.0, "Further advancements for E-UTRA physical layer aspects (Release 9)", March 2010

[20] A. Virdis, G. Stea, G. Nardini, "Simulating LTE/LTE-Advanced Networks with SimuLTE", in: Obaidat, M.S., Kacprzyk, J., Ören, T., Filipe, J., (eds.) "Simulation and Modeling Methodologies, Technologies and Applications", Springer, 2016

[21] ILOG CPLEX Software, http://www.ilog.com

[22] 3GPP TR 36.885 v14.0.0, "Study on LTE-based V2X services (Release 14)", June 2016

[23] P. Levis, N. Patel, D. Culler, S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks", in proceedings of 1st USENIX/ACM Symposium NSDI, pp. 15-28, 2004

[24] L. Junhai, Y. Danxia, X. Liu and F. Mingyu, "A survey of multicast routing protocols for mobile Ad-Hoc networks," in IEEE Communications Surveys & Tutorials, vol. 11, no. 1, pp. 78-91, First Quarter 2009. doi: 10.1109/SURV.2009.090107

[25] M. Ahmed, et al., "A Survey on Socially Aware Device-to-Device Communications," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2169-2197, thirdquarter 2018

[26] M. Waqas et al., "Mobility-Aware Device-to-Device Communications: Principles, Practice and Challenges," in IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1863-1886, third quarter 2020, doi: 10.1109/COMST.2019.2923708.

[27] J.W. Chinneck, "An effective polynomial-time heuristic for the minimum-cardinality IIS set-covering problem", in Annals of Mathematics and Artificial Intelligence, vol. 17, 1996, pp. 127-144, DOI: 10.1007/BF02284627

[28] M. Masoudifar, "A review and performance comparison of QoS multicast routing protocols for MANETs", in Ad Hoc Networks, Volume 7, Issue 6, 2009, Pages 1150-1155, ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2008.10.004.

[29] B. Yang, Z. Wu, Y. Shen, X. Jiang, S. Shen, "On delay performance study for cooperative multicast MANETs", in Ad Hoc Networks, Volume 102, 2020, 102117, ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2020.102117.

[30] C. Sá de Abreu, R. Moreira Salles, "Modeling message diffusion in epidemical DTN", in Ad Hoc Networks, Volume 16, 2014, Pages 197-209, ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2013.12.013.

[31] Z. Zhou, H. Yu, C. Xu, Y. Zhang, S. Mumtaz and J. Rodriguez, "Dependable Content Distribution in D2D-Based Cooperative Vehicular Networks: A Big Data-Integrated Coalition Game Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 953-964, March 2018, doi: 10.1109/TITS.2017.2771519.

[32] V. Sciancalepore, V. Mancuso, A. Banchs, S. Zaks and A. Capone, "Enhanced Content Update Dissemination Through D2D in 5G Cellular Networks," in IEEE Transactions on Wireless Communications, vol. 15, no. 11, pp. 7517-7530, Nov. 2016, doi: 10.1109/TWC.2016.2604300.

[33] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz and J. Rodriguez, "Social Big-Data-Based Content Dissemination in Internet of Vehicles," in IEEE Transactions on Industrial Informatics, vol. 14, no. 2, pp. 768-777, Feb. 2018, doi: 10.1109/TII.2017.2733001.

[34] A. Baiocchi, P. Salvo, F. Cuomo and I. Rubin, "Understanding Spurious Message Forwarding in VANET Beaconless Dissemination Protocols: An Analytical Approach," in IEEE Transactions on Vehicular Technology, vol. 65, no. 4, pp. 2243-2258, April 2016, doi: 10.1109/TVT.2015.2422753.

[35] P. Gandotra, R. K. Jha, "Device-to-Device Communication in Cellular Networks: A Survey", in Journal of Network and Computer Applications, vol. 71, 2016, pp. 99-117, ISSN 1084-8045, doi: https://doi.org/10.1016/j.jnca.2016.06.004.

[36] N. Patriciello, S. Lagen, L. Giupponi and B. Bojovic, "5G New Radio Numerologies and their Impact on the End-To-End Latency," in procceedings of the 23rd IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2018, pp. 1-6, doi: 10.1109/CAMAD.2018.8514979.

[37] M. Ali, A.W. Malik, A.U. Rahman, S. Iqbal, M.M. Hamayun, "Position-based emergency message dissemination for Internet of vehicles", in International Journal of Distributed Sensor Networks. July 2019. doi:10.1177/1550147719861585

[38] C. Sauer, E. Lyczkowski, M. Sliskovic and M. Schmidt, "Real-time Alarm Dissemination in Mobile Industrial Networks", in proceedings of the 22nd IEEE International Conference on Industrial Technology (ICIT), 2021, pp. 1152-1156, doi: 10.1109/ICIT46573.2021.9453568.

[39] E. G. Cabral-Pacheco, S. Villarreal-Reyes, A. Galaviz-Mosqueda, S. Villarreal-Reyes, R. Rivera-Rodriguez and A. E. Perez-Ramos, "Performance Analysis of Multi-Hop Broadcast Protocols for Distributed UAV

Formation Control Applications", in IEEE Access, vol. 7, pp. 113548-113577, 2019, doi: 10.1109/ACCESS.2019.2935307.
[40] 3GPP TS 36.213 v16.7.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 16) ", September 2021
[41] J. J. Gimenez *et al*., "5G New Radio for Terrestrial Broadcast: A Forward-Looking Approach for NR-MBMS", in IEEE Transactions on Broadcasting, vol. 65, no. 2, pp. 356-368, June 2019, doi: 10.1109/TBC.2019.2912117.

Giovanni Nardini is Assistant Professor at the University of Pisa, where he obtained his MSc degree in Computer System Engineering in 2013, and his PhD in Information Engineering in 2017. His research interests include resource allocation algorithms in 4G and 5G networks, multi-access edge computing, simulation and performance evaluation of computer networks. In these fields he has coauthored six patents and more than 40 papers. He has been involved in EU-funded and industrial research projects.

Giovanni Stea is Associate Professor at the Department of Information Engineering of the University of Pisa, Italy, where he also got his PhD in 2003. His current research interests include Quality of Service and resource allocation in wireline and wireless networks, performance evaluation through simulation and analytical techniques, traffic engineering. In these fields he has coauthored more than 100 peer-reviewed papers and 16 patents. He has been involved in national and EU research projects, and he has led joint research projects with industrial partners. He has served as a member of the technical and/or organization committees for several international conferences, including SIGCOMM, WoWMoM, and VALUETOOLS, and he is serving on the editorial board of the Wireless Networks journal.

Antonio Virdis is Assistant Professor at the University of Pisa, where he obtained his MSc degree in Computer System Engineering in 2011, and his PhD in Information Engineering in 2015. His research interests include Quality of Service, scheduling and resource allocation in wireless networks, network simulation and performance evaluation. He has been involved in national, EU-funded and industry-funded research projects. He coauthored seven patents and more than 50 peer-reviewed papers.

VII. Appendix

A. Deterministic formulation

We show here that the problem of computing the optimal schedule in a *deterministic* graph is simpler. In the latter, the available D2D links are *reliable*, i.e. transmissions through them are always successful. The problem is as follows:

$$
\begin{array}{lll}
\min \sum_{i=1}^{N} x_i^{(1)} + \frac{1}{N} \cdot \sum_{t=1}^{k} \sum_{i=1}^{N} g_i^{(t)} & & \\
s.t. & & \\
g_i^{(t)} \le x_i^{(t)} & \forall i, \forall t & (i) \\
x_i^{(k)} = 1 & \forall i & (ii) \\
x_i^{(t)} \ge x_i^{(t-1)} & \forall i, \forall t > 1 & (iii) \\
x_i^{(t)} \ge g_j^{(t-1)} \cdot a_{j,i} & \forall i, \forall j, \forall t > 1 & (iv) \\
x_i^{(t)} \le x_i^{(t-1)} + \sum_{j=1}^{N} g_j^{(t-1)} \cdot a_{j,i} & \forall i, \forall t > 1 & (v) \\
x_i^{(t)} \in \{0,1\} & \forall i, \forall t & (vi) \\
g_i^{(t)} \in \{0,1\} & \forall i, \forall t & (vii)
\end{array}
\tag{13}
$$

Constraint (i) states that only a UE which has the message can be given a grant. Constraint (ii) is the termination condition, i.e. all nodes have the message at round k. Constraint (iii) states that a node that has the message after a round will continue to have it afterwards. Constraint (iv)-(v) state that a node receives the message if and only if at least one of its one-hop neighbors is given a grant in the previous round. Coefficient $a_{j,i}$ is the corresponding element in the binary *adjacency matrix* $\boldsymbol{A}$, which states which edges are in the graph. Note that $g_i^{(t)} \in \{0,1\}$, since there is no need to give more than *one* grant per node under deterministic reception. More to the point, each node will receive at most one grant *in the whole schedule*. However, a constraint such as $\sum_{t=1}^{k} g_i^{(t)} \le 1$ is redundant, since the optimal solution will obviously verify it. For the same reason, the multiplying coefficient of the second summation in the objective function can be $1/N$ and the number of initial grants will still have priority over the number of D2D grants.

Problem (13) is an ILP with $O(N \cdot k)$ binary variables. ILPs are $\mathcal{NP}$-hard, in general, and can be solved using the branch-and-bound algorithm. Polynomial-time heuristics based on LP-relaxation and rounding are also available [16], and they may be fast enough for our context. Here we present a simple heuristic solution algorithm, which computes the schedule one round at a time. The algorithm is as follows:

a) First, determine a minimal set of starting points, and target it with DL transmissions.
b) Then, on each round, compute the minimum set of D2D grants that covers the entire one-hop neighborhood of current message holders.

Problem a), i.e. identifying the minimum set S so that every other node can be reached in k hops, is the following:

$$
\begin{array}{l}
\min \boldsymbol{e}^T \cdot \boldsymbol{X} \\
s.t. \\
\boldsymbol{A}^k \cdot \boldsymbol{X} \ge \boldsymbol{e} \\
x_i \in \{0,1\}, \forall i
\end{array}
\tag{14}
$$

where $\boldsymbol{e}$ is a vector of 1s. The entries of $\boldsymbol{A}^k$ are in fact the number of different paths between two nodes. This is a *minimum set covering* problem, an ILP for which efficient polynomial heuristics can be found [27]. To solve problem b), set $t = 1$ and iterate through the following steps:

1. let $W^{(t)}$ be the set of nodes that have the message at round t, and call $Q^{(t)}$ the one-hop neighborhood of $W^{(t)}$. Exclude from $Q^{(t)}$ all the nodes already in $W^{(t)}$.

2. Construct from graph $\mathcal{G}$ the bipartite subgraph $\mathcal{H}^{(t)}$ having nodes in $W^{(t)}$ as the first layer, and the nodes in $Q^{(t)}$ as the second layer, and only arcs from a node in $W^{(t)}$ to a node in $Q^{(t)}$, and solve another set-covering problem on $\mathcal{H}^{(t)}$, i.e. find the minimum set of D2D grants to be given to nodes in $W^{(t)}$ so that all nodes in $Q^{(t)}$ are reached.
3. Remove nodes in $W^{(t)}$ from graph $\mathcal{G}$.
4. Increase t and repeat until $t = k$.

It is obvious that the above algorithm covers the whole *k*-hop neighborhood of the initial set *S*. Since the latter is computed as the set of nodes whose k-hop neighborhood is the entire set of nodes, then the above algorithm converges to a solution of (13). The following considerations are in order:

With respect to the objective of (13), the above heuristic finds a solution with the same number of initial starting points (first sum in the objective) by definition, and possibly *more* grants (second double summation in the objective). This is due to its *greedy* approach, according to which a neighbor must be reached if it can be reached at all. The optimal solution of (13) may instead defer reaching a node to a later round, when that node would be reached "for free" due to the fact that one of its neighbors must be given a grant. On the other hand, with the above algorithm every node is reached *as soon as possible*, which is not guaranteed by the optimum of (13). The iterative step of the above algorithm involves solving at optimality a set covering over a *smaller* set of nodes (i.e., the bipartite subgraph $\mathcal{H}^{(t)}$). This makes it faster to solve. All in all, the above heuristic involves solving $O(k)$ ILPs with $O(N)$ binary variables each. These problems may even be solvable at optimality in reasonable times, at least for small values of N, or they may be solved in polynomial time through LP-relaxation and rounding at larger scales.

Given the above, the alert reader may wonder if transforming our original probabilistic problem into a deterministic one via edge-weight rounding, i.e., a mapping $P_{i,j} \to \{0,1\}$, could be a viable solution strategy. Unfortunately, the answer is negative. In fact, edge-weight rounding makes it impossible to assess the reliability of a path, hence one would be able to guarantee a maximum number of hops, but not a target reliability. A rounding that preserves reliability *a priori* is the following:

$$a_{i,j} = 1 \ iff. \ P_{i,j} \geq \sqrt[k]{\alpha} \qquad (15)$$

The above rounding, however, removes most of the edges from the graph, reducing its connectivity and easily making it disconnected. As an example, Table V shows the minimum link probabilities that would survive the edge pruning of (15). It is clear that a graph with so few links will require a considerably larger number of initial DL grants, i.e. to reach isolated nodes. It is also bizarre, to say the least, that increasing k has the adversarial effect of increasing the cost of a schedule, since it makes the graph sparser, hence requiring more starting points. Increasing k should instead allow one to capitalize the gain of multihop relaying, thus *reducing* the number of starting points.

Table V – Minimum per-link probabilities that survive rounding as per equation (15)

		k						
		2	**3**	**4**	**5**	**6**	**7**	**8**
α	**0.9**	0.949	0.965	0.974	0.979	0.983	0.985	0.987
	0.95	0.975	0.983	0.987	0.990	0.991	0.993	0.994
	0.99	0.995	0.997	0.997	0.998	0.998	0.999	0.999

B. Symbols used in the paper

Table VI reports the symbols used in the paper, along with their meaning.

Table VI – Symbols used in the paper

Symbol	***Meaning***
N	Number of UEs
$P_{i,j}$	probability that recipient j decodes a message transmitted from transmitter i.
$\boldsymbol{P}$	Matrix including values $P_{i,j}$.
$\mathcal{G}$	Graph of the network.
$\mathcal{V}$	Set of vertices of graph $\mathcal{G}$.
$\mathcal{E}$	Set of edges of graph $\mathcal{G}$.
k	Maximum number of rounds.
α	Target reliability (e.g., 95%).
$\boldsymbol{X}$	Column N-vector of binary elements x_j
x_j	x_j is one if node j is a starting point for the algorithm, and zero otherwise.
$\boldsymbol{G}$	$N \times k$ grant matrix. Each column $\boldsymbol{g}^{(t)}$ represents the grants given by the algorithm at round t. $g_j^{(t)} = w$ if node j receives w grants at round t.
$\boldsymbol{g}^{(t)}$	Column t of matrix $\boldsymbol{G}$.
$g_j^{(t)}$	Number of grants received by node j at round t.
$\boldsymbol{m}^{(t)}$	Column N-vector of elements $m_j^{(t)}$.
$m_j^{(t)}$	The probability that node j possesses the message at round t.
d	The time it takes for a node to decode a message.
r_p	Reliability of path p.
$r_{i,j}^{(k)}$	reliability of the k-*hop-constrained max-reliability* path between i and j (*k-mr* path).
$\mathcal{Q}^{(k)}$	k-hop reliability closure of $\mathcal{G}$.