

# Algorithms to Find Two-Hop Routing Policies in Multiclass Delay Tolerant Networks

Nicola Basilico, Matteo Cesana, *Member, IEEE*, and Nicola Gatti

**Abstract**—Most of the literature on delay tolerant networks (DTNs) focuses on optimal routing policies exploiting *a priori* knowledge about nodes mobility traces. For the case in which no *a priori* knowledge is available (very common in practice), apart from basic epidemic routing, the main approaches focus on controlling two-hop routing policies. However, these latter results commonly employ fluid approximation techniques, which, in principle, do not provide any theoretical bound over the approximation ratio. In our work, we focus on the case without *a priori* mobility knowledge and we provide approximation algorithms with theoretical guarantees that can be applied to cases where the number of hops allowed in the routing process is arbitrary. Our approach is rather flexible allowing us to address heterogeneous mobility patterns and transmission technologies, to consider explicitly the signaling and transmission costs, and to include also nodes discarding packets after a local timeout. We then provide a comprehensive performance evaluation of our algorithms, showing that two-hop routing provides the best tradeoff between delay and energy and that, in this case, they find solutions very close to the optimal ones with a low overhead. Finally, we compare our methods against some state-of-the-art approaches by means of a DTN simulation environment in realistic settings.

**Index Terms**—Delay Tolerant Networks, performance evaluation, routing.

## I. INTRODUCTION

DTNs are sparse and/or highly mobile wireless ad hoc networks with discontinuous connectivity, which may occur due to limits in the wireless radio range, sparsity of mobile nodes, constrained energy resources, attacks, and noise. One central problem in DTNs is the optimal routing of packets from a source towards the desired destinations.

When no *a priori* information is available on the mobility pattern of the nodes, a common technique for overcoming lack of connectivity is to disseminate multiple copies of the packet: this enhances the probability that at least one of them will reach the destination within a temporal deadline. This is referred to as *epidemic-style* forwarding, because, alike the spread of infectious diseases, each time a packet-carrying node encounters a

Manuscript received June 26, 2015; revised October 28, 2015 and February 9, 2016; accepted February 11, 2016. Date of publication February 23, 2016; date of current version June 7, 2016. This work was supported by the Italian Ministry for Education, University and Research (MIUR) through projects MIE (CTN01-00034-594122) and People-Net (PRIN 2009BZM837-003). The associate editor coordinating the review of this paper and approving it for publication was M. Li.

N. Basilico is with the Department of Computer Science, University of Milan, Milan 20135, Italy (e-mail: nicola.basilico@unimi.it).

M. Cesana and N. Gatti are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano 20133, Italy (e-mail: matteo.cesana@polimi.it; nicola.gatti@polimi.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TWC.2016.2532859

new node not having a copy thereof, the carrier may infect this new node by passing on a packet copy. A convenient compromise of energy versus delay compared to epidemic routing is provided by two-hop routing where the infection is limited to contacts between the source and intermediary nodes, that is, the source node passes on the packets to all the mobile nodes she encounters, and the “infected” mobile nodes can deliver the packets they are carrying only to the final destination.

In this paper, we focus on the characterization of two-hop routing policies for DTNs as in [1]–[3], where the problem concerns the decision on whether or not forwarding a given packet to a given mobile node the source is encountering at a given time. We propose an optimization-based framework to derive optimal two-hop routing policies which extends the available approaches in the literature in different directions: (i) we account for mobile nodes which are categorized in distinct multiple classes capturing different mobility patterns and available communication technologies on board; (ii) we account for the cost for neighbor discovery and signaling messages exchange to support packet forwarding further allowing mobile nodes to discard the packets they are carrying upon expiration of a local temporal deadline; (iii) rather than resorting to fluid approximation to derive optimal two-hop routing policies, we propose algorithms for finite mobile node populations with theoretical guarantees; namely we provide an exponential-time (in the number of classes) algorithm to find routing policies that are arbitrarily close to the optimal ones, as well as approximate polynomial-time algorithms; (iv) we extend our algorithms and analysis to the case where the routing policy allows an arbitrary maximum number of hops. Finally, we provide a thorough performance evaluation with realistic settings of the proposed algorithms in terms of approximation ratio, scalability in the number of classes, further evaluating the impact of network parameters onto the optimal routing policies. We also evaluate our routing policies with different maximum numbers of allowed hops and we compare them w.r.t. state-of-the-art routing policies within The ONE Simulator [4].

The paper is organized as follows. Section II reviews the relevant literature. Sections III and IV describe the reference scenario and routing problem. Our approximation algorithms are presented in Section V while extensions to more than two hops is addressed in Section VI. Section VII, evaluates algorithms in synthetic network instances while Section VIII concludes the paper.

## II. RELATED WORK

The main distinctive features of the present work w.r.t. the reference literature are: (i) we take into account the cost for

neighbor discovery and signaling messages exchange to support packet forwarding, further allowing mobile nodes to discard “older” packets upon expiration of a local temporal deadline (implying that, differently from what customarily assumed in the reference literature, the number of copies of a packet is not monotone-increasing); (ii) we do not resort to fluid approximation, but rather propose algorithms to find optimal routing policies for finite mobile node populations with theoretical performance guarantees; (iii) we extend our algorithms to the case where the number of hops allowed in the routing phase is arbitrary; (iv) we compare our algorithms w.r.t. state-of-the-art algorithms by means of one of the main DTN simulators.

We overview here the main body of literature in the fields of optimizing multi-hop routing and two-hop routing in DTNs.

**Multi-Hop Routing.** Besides basic epidemic-style forwarding schemes operating under a zero-information assumption (in Section VII-E we briefly review and compare against them), literature on optimal routing mainly devoted efforts in scenarios where some knowledge about mobility is available and can be exploited. The seminal work [5] studies optimal multi-hop routing strategies when the nodes have a limited buffer further providing an experimental comparison of the different routing policies. In [6], the authors address the DTN routing problem by first proposing an optimization framework to optimally set the routes and then by introducing a gradient-based routing heuristic which exploits the concept of connectivity degree. In [7], the authors cast the routing problem as an optimal stopping rule problem and further propose an Optimal Opportunistic Routing scheme which maximizes the average packet delivery probability. In [8], the authors focus on a multi-hop routing heuristic named Ring Distribution Routing. In all the aforementioned works, routing decisions only leverage topological information such as the contact time and statistics. Differently, recent literature shows that routing performance can be improved if social information on the mobile nodes can be leveraged [9]–[12].

**Optimal Control for Two-Hop Routing.** The seminal work [13] studies optimal static and dynamic control (proved to be threshold based) policies for two-hop DTN when mobile nodes are homogeneous. Furthermore, the authors show that when the parameters are unknown it is still possible to obtain a policy that converges to the optimal one by using some adaptive auto-tuning mechanism. Extensions of such adaptive mechanism are proposed in [14].

Scenarios where mobile nodes belong to multiple distinct classes are studied in [3], showing that the routing policy may be class dependent. The authors resort to fluid approximation to characterize the routing policies under the assumption that the number of copies of the packet is monotonically increasing in time. However, no theoretical guarantee over the quality of the solutions is provided and, in principle, fluid approximation may provide arbitrarily inefficient solutions, see, e.g., [15]. Furthermore, it is not clear whether fluid-approximation approaches can be extended to more than two hops. The authors, in [2], extend the previously mentioned work by considering also non-monotone routing strategies, whereas Chahin *et al.* design optimal control rules to maximize the packet delivery probability under energy budget constraints [16]. The aforementioned work focuses on routing control policies which

assume disjoint traffic generation and routing, that is, the routing process is completely decoupled from the traffic generation one. Alma *et al.* extend this previous work to account for the case in which traffic sources continuously generate traffic during the routing process [17]. The tradeoff between energy consumption and packet delivery probability is studied in [18] in the case where packet replication is allowed at the source to create redundancy in the spread-out information.

A fluid representation of the routing process is adopted also in [19], where a scheduling framework is proposed to let each mobile node locally decide if/when a packet in transmission should be dropped or forwarded under the assumption that the packet forwarding process can be approximated by a time-continuous Markov chain process. In [20], the authors characterize the probability distribution of the packet delivery delay for epidemic and two-hop routing schemes; moreover, they also evaluate the communication cost measured as the number of replica of a given packet at the time the packet has been received by the intended destination.

### III. REFERENCE SCENARIO

We consider an environment populated by one *source* node, one *sink* node and multiple *mobile nodes*. *Sink* and *source* nodes may as well be mobile. A packet is initially held by the source and it must be delivered to the sink no later than  $\tau$  time units through two-hop opportunistic routing. (In Tab. I, we summarize all the symbols used along the paper.) Thus, the source can decide to deliver the packet to any mobile node she gets in touch with, and such mobile node can only deliver the packet to the *sink* in the event of a direct contact. A mobile node carrying a packet from the source discards the packet after a pre-defined time-out (defined below), further refraining from accepting the very same packet again in the future.

Each mobile node belongs to a specific class  $c \in C$ . We denote by  $N_c$  the number of mobile nodes of class  $c$ . Each class encodes the features of its nodes, including their *mobility profile* and *transmission technology*. The mobility profile is characterized by the average speed  $v_c$  and by the class-specific time-out value  $t_c$  (i.e., the time after which the node discards the packet and does not accept it again in the future). We denote by  $\omega \in \Omega$  a transmission technology (e.g., WiFi). Transmission technologies are characterized by a number of parameters that we introduce below. For the sake of simplicity, we assume each node to use a single technology, while the source and the sink can use all the technologies. All the nodes of a class  $c$  use the same technology. Finally, the subset of classes adopting technology  $\omega$  is  $C_\omega \subseteq C$ .

The interactions among nodes of the same class are regulated by the following rules/parameters: (i) two nodes are in contact at a given time if they are within each other’s transmission range (we denote by  $R_\omega$  the transmission range of technology  $\omega$ , and we use, with slight abuse of notation,  $R_c$  in place of  $R_\omega$  when  $c \in C_\omega$ ); (ii) technology-specific neighbor discovery procedures are used to dynamically discover contacts over time; (iii) upon neighbor discovery, technology-specific association procedures are adopted to create peer-to-peer connections among nodes in contact; (iv) once the association

TABLE I  
 NOTATION SUMMARY

Network/topology parameters		Class-dependent node parameters		Scenario parameters	
$\tau$	packet delivery deadline	$N_c$	number of nodes in class $c$	$\Psi$	energy budget
$\Delta$	slot duration	$t_c$	packet's local time to live	$\mu$	forwarding policy
$K$	slot number	$v_c$	average speed	$\mu(k)$	forwarding policy at slot $k$
$L$	radius of the circle	$R_c$	transmission range	$\mu_c$	forwarding policy for class $c$
$\iota$	constant	$\beta_c$	signaling cost	$\mu_{-c}$	forwarding policy for all the classes except $c$
$C$	set of classes	$\rho_c$	transmission cost	$\mu_c(k)$	forwarding policy at slot $k$ for class $c$
$c$	a class in $C$	$\lambda_c$	contact rate	$h_c$	threshold for class $c$
$\Omega$	set of technologies			$F_D(\mu, K)$	probability of delivering the message before time $K\Delta$ given $\mu$
$\omega$	a technology in $\Omega$			$X_{c,k}(\mu)$	random variable of the number of mobile nodes of class $c$ that received the packet by $k$
$C_\omega$	set of classes adopting technology $\omega$			$Y_{c,k}(\mu)$	random variable of the number of mobile nodes of class $c$ that still keep a copy of the packet at time slot $k$
$\beta_\omega$	signaling cost of technology $\omega$			$Q_{c,k,k'}(\mu_c)$	probability that a mobile node of class $c$ does not receive any packet in time slots $k, \dots, k'$ given $\mu_c$
$\rho_\omega$	transmission cost of technology $\omega$			$X_{c,h}^*(s, \mu)$	Laplace-Stieltjes transform of $Y_{c,h}(\mu)$
$R_\omega$	transmission range of technology $\omega$				

phase has been carried out, nodes carrying the reference packet may decide to transfer it to the associated node if the active routing policy so prescribes.

We assume all the three aforementioned operations performed by nodes to incur in some energy cost. W.l.o.g., let  $\beta_\omega$  represent the energy consumed for performing operations (ii)–(iii) by technology  $\omega$ , and  $\rho_\omega$  represent the energy consumed to transmit the reference packet by technology  $\omega$ . We will refer to  $\beta_\omega$  and  $\rho_\omega$  as to the *signaling* and *transmission* costs, respectively. With slight abuse of notation, we use  $\beta_c$  and  $\rho_c$  in place of  $\beta_\omega$  and  $\rho_\omega$ , respectively, when  $c \in C_\omega$ . All the classes  $c \in C_\omega$  using the same technology  $\omega$  share the signaling costs  $\beta_\omega$ .

The implementation details and parameter values for the routing signaling phase are technology dependent; e.g., referring to WiFi Direct technology, points (ii) and (iii) include the required time and message exchange to perform IEEE 802.11 Channel Scanning, Channel Probing, Group Owner Negotiation and Address Configuration [21]; referring to Bluetooth Low Energy technology, points (ii) and (iii) include the Advertising and Scanning/Initiating phases [22], [23].

We consider a discrete representation of time organized in time slots whose duration is fixed to  $\Delta$  time units and we denote the total number of useful time slots as  $K = \lfloor \tau/\Delta \rfloor$ , where the  $k$ -th time slot corresponds to the time interval  $[k\Delta, (k+1)\Delta)$ .

Transmission opportunities between two nodes are given by contacts taking place when each node is within the communication range of the other one. As we are considering two-hop routing schemes, contacts of interest are limited to those between the source and mobile nodes and between mobile nodes and the sink. In the following, we mainly rely on Markovian models for the packet spreading process, that is, we assume that contacts at the source and at the sink follow a multi-class Poisson distribution; such assumption is largely used in the literature to study the performance of opportunistic routing as it eases up the problem's tractability while maintaining practical insights in the routing design problem [13]–[27]. Recent studies further support such assumption by showing that Poisson distributions well approximate the contact numbers in opportunistic networks with nodes moving according to realistic mobility models provided that the transmission range is large w.r.t. the reference arena and the speed is sufficiently large [28]. In the following, we leverage the formulas derived in [28]

to approximate the pairwise contact creation rate. Namely, in our analysis,  $\lambda_c$  (the *contact rate* of nodes belonging to class  $c$ ) is defined as  $\lambda_c = \frac{8\iota R_c v_c}{\pi L^2}$  where  $\iota$  is a constant set to 1.3693 and  $L$  is the radius of circle in which the nodes move. With Poisson distributions, optimal policies are zero-memory [13]. That is, the best policies from a time slot on do not depend on the contacts happened in the time slots before and therefore optimal policies do not depend on information available at runtime.

When a contact happens between the source and any mobile node that did not receive the packet yet, the source decides to hand over the packet according to a *forwarding policy*  $\mu$  which depends on the current time and the potential recipient's class. Given a time slot  $k$  and a class  $c$ , the policy profile at time  $k$  is  $\mu(k) = (\mu_1(k), \dots, \mu_{|C|}(k))$  where  $\mu_c(k)$  indicates the forward probability at time slot  $k$  for class  $c$ ; we also denote with  $\mu_c$  the entire policy for such class  $c$ . In general, when the packet is forwarded, some energy is spent and the packet's delivery probability is increased. We denote with  $F_D(\mu, K)$  the probability of delivering the message before time  $K\Delta$  given policy profile  $\mu$ . Obviously, such value is prevented from growing indefinitely by an *energy budget*, denoted by  $\Psi$ , on the total spent energy (including both signaling and transmission).

## IV. OPTIMAL ROUTING POLICIES

### A. Problem Formulation

We define  $X_{c,k}(\mu)$  as the random variable expressing the number of mobile nodes of class  $c$  that have received the packet by time slot  $k$ , while  $Y_{c,k}(\mu)$  is a random variable expressing the number of mobile nodes of class  $c$  that still keep a copy of the packet at time slot  $k$ . These variables both depend on  $\mu$  and are, in general, different. Indeed, since a mobile node can both receive and discard a packet before time slot  $k$ , we have that  $Y_{c,k}(\mu) \leq X_{c,k}(\mu)$ . Furthermore, we denote by  $Q_{c,k,k'}(\mu_c)$  the probability that a mobile node of class  $c$  does not receive any packet in time slots  $k, \dots, k'$  as function of  $\mu_c$  which can be expressed as:

$$Q_{c,k,k'}(\mu_c) = e^{-\lambda_c \Delta \sum_{i=k}^{k'} \mu_c(i)}. \quad (1)$$



The expected number of mobile nodes of class  $c$  that receive a packet in time slots  $0, \dots, k$  is:

$$\mathbb{E}[X_{c,k}(\boldsymbol{\mu})] = N_c \cdot (1 - Q_{c,0,k}(\boldsymbol{\mu}_c)). \quad (2)$$

Thus, the expected number of mobile nodes of class  $c$  that have the packet at time slot  $k$  is:

$$\mathbb{E}[Y_{c,k}(\boldsymbol{\mu}_c)] = N_c \cdot (1 - Q_{c,\max\{0,k-t_c\},k}(\boldsymbol{\mu}_c)). \quad (3)$$

The probability that a packet is delivered to the sink by time  $k\Delta$  can be defined as:

$$F_D(\boldsymbol{\mu}, k) = 1 - \prod_{c \in C} \prod_{h=0}^{k-1} X_{c,h}^*(\lambda_c \Delta, \boldsymbol{\mu}), \quad (4)$$

being

$$X_{c,h}^*(s, \boldsymbol{\mu}) = \mathbb{E}[e^{-sY_{c,h}(\boldsymbol{\mu})}], \quad (5)$$

where  $X^*$  is the Laplace–Stieltjes transform (see [13] for details). Note that Eq. (4) inherently uses the Markovian assumption on the contact inter-arrival time, which allows to consider variables  $Y_{c,h}$  to be independent w.r.t. the temporal index  $h$  (i.e., the number of mobile nodes holding the packet is memoryless over time). The constraint on the consumed energy is:

$$\underbrace{\sum_{c \in C} \rho_c N_c (1 - Q_{c,0,K}(\boldsymbol{\mu}_c))}_{\text{transmission costs}} + \underbrace{\sum_{\omega \in \Omega} \sum_{k=0}^{K-1} \beta_\omega \cdot \left(1 - \prod_{c \in C_\omega} (1 - \mu_c(k))\right)}_{\text{signaling costs}} \leq \Psi. \quad (6)$$

The left term of the inequality adds up the expected transmission costs with the expected signaling costs for class  $c$ , given a policy profile  $\boldsymbol{\mu}$ . In particular, transmission costs are obtained by multiplying  $\rho_c$  by the expected number of nodes that will receive the packet from slot 0 to slot  $K$ ; on the other side, a signaling cost of  $\beta_\omega$  is paid for each time slot  $k$  with the probability that at least one class  $c \in C_\omega$  will forward the packet, i.e.,  $1 - \prod_{c \in C_\omega} (1 - \mu_c(k))$ .

The problem of finding the optimal routing policy can be formally defined as follows: find policy  $\boldsymbol{\mu}^*$  that maximizes  $F_D(\boldsymbol{\mu}, K)$  subject to the budget constraint reported in Eq. (6).

## B. Problem Properties

We now show some theoretical properties that we will exploit in the following.

**Property IV.1.** *Optimal policies either completely consume the budget or prescribe that all the classes transmit for all the slots.*

*Proof:* It is easy to see that  $F_D(\boldsymbol{\mu}, K)$  is monotonically increasing in  $\sum_{h=0}^{K-1} \mu_c(h)$  and that, as a consequence, transmitting for a larger (expected) number of slots cannot result in a lower delivery probability. ■

Similarly to what proposed in [13], we define a threshold-based policy  $\mu_c$  as:

$$\mu_c(k) = \begin{cases} 1 & k < \lfloor h_c \rfloor \\ \{h_c\} & k = \lfloor h_c \rfloor, \\ 0 & k > \lfloor h_c \rfloor \end{cases}$$

where  $h_c$  is the threshold of class  $c$  and  $\{h_c\}$  is the fractional part of  $h_c$ . The following property, which was holding in the single-class case, keeps valid also when multiple classes are allowed.

**Property IV.2.** *Optimal policies are threshold based.*

*Proof:* The delivery delay c.d.f. is  $F_D(\boldsymbol{\mu}, K) = 1 - \prod_c \Gamma_c(\boldsymbol{\mu}, s)$ , where  $\Gamma_c(\boldsymbol{\mu}, s) = \prod_{h=0}^{K-1} X_{c,h}^*(s, \boldsymbol{\mu})$  and  $s = \lambda_c \Delta$ . Let us denote with  $\boldsymbol{\mu}_c$  a non-threshold policy for class  $c$  and with  $\hat{\boldsymbol{\mu}}_c$  a policy obtained by shifting to the left all the non-empty slots of  $\boldsymbol{\mu}_c$  and by rounding them so that  $\hat{\boldsymbol{\mu}}_c$  matches the definition of threshold policy introduced above. For any  $(\boldsymbol{\mu}_c, \hat{\boldsymbol{\mu}}_c)$  obtained in this way we have that  $\Gamma_c(\boldsymbol{\mu}_c, s) \geq \hat{\Gamma}_c(\boldsymbol{\mu}_c, s)$  and therefore, defined  $\Gamma_{-c}(\boldsymbol{\mu}, s) = \prod_{c' \neq c} \prod_{h=0}^{K-1} X_{c',h}^*(s, \boldsymbol{\mu})$ , we have  $1 - \Gamma_c(\boldsymbol{\mu}, s) \cdot \Gamma_{-c}(\boldsymbol{\mu}, s) \leq 1 - \hat{\Gamma}_c(\boldsymbol{\mu}_c, s) \cdot \Gamma_{-c}(\boldsymbol{\mu}, s)$ , that is, for any given joint policy  $\boldsymbol{\mu}$ , if we substitute the marginal policy of a class  $c$  with its threshold version the probability of delivery within  $K$  time slots will not decrease. ■

**Property IV.3.** *Optimal policies can prescribe non-integer thresholds for all the classes.*

*Proof:* Consider, e.g., a two-class scenario:  $K = 20$ ,  $\Delta = 100$ ,  $\Psi = 0.7$ ,  $N_1 = 1$ ,  $N_2 = 2$ ,  $\lambda_1 = 21 \times 10^{-5}$ ,  $\lambda_2 = 20 \times 10^{-5}$ ,  $t_1 = t_2 = K$ . We approximate the optimal policy profile by discretizing the values of  $h_c$  with a fine grid of step 0.01. In addition to these points, we consider all the points in which the threshold of one class is integer and the threshold of the other class is calculated in such a way the budget is completely consumed. We evaluate the objective function at all these points and select the maximum. The approximately optimal policy is  $h_1 = 7.87$ ,  $h_2 = 15.91$ . Hence, at the optimum, a fractional part is assigned to both classes. ■

**Property IV.4.** *The optimization problem with objective function given by Eq. (4) and constraint given by Eq. (6) is nonlinear and nonconvex.*

*Proof:* Nonlinearity is trivial. Nonconvexity is proved by showing the nonconvexity of the feasibility region by computing the Hessian matrix of the budget constraint in Eq. (6) to which we will refer here with  $u$  (notice that we restrict our attention on threshold policies). It can be easily seen that all the eigenvalues of the Hessian matrix (of the form  $-N_{|c|} \lambda_{|c|}^2 \Delta^2 e^{-\lambda_{|c|} \Delta h_{|c|}}$ ) are strictly negative for every policy profile  $\boldsymbol{\mu}$  and therefore the feasibility region is non convex. ■

Some of the above properties show that the optimization problem is hard and that no exact algorithm is possible. The adoption of non-convex programming techniques cannot assure to approximate the global optimum. Hence, we focus on the problem of developing approximation algorithms and of studying their theoretical and empirical approximation errors.

**Algorithm 1.**  $\epsilon$ -grid search

---

```

1:  $F^* \leftarrow 0$ 
2: if  $\mu$  s.t.  $h_c = \frac{K-1}{\epsilon}$  for all  $c$  is feasible then
3:    $\mu^* = \mu$ 
4: else
5:   for  $c \in C$  do
6:     for every  $\mu = (\mu_c, \mu_{-c})$  s.t.  $\mu_{-c}$  is integer and budget
        $\Psi$  is entirely consumed do
7:       if  $F_D(\mu, \frac{K-1}{\epsilon}) > F^*$  then
8:          $\mu^* \leftarrow \mu$ 
9:          $F^* \leftarrow F_D(\mu, \frac{K-1}{\epsilon})$ 

```

---

## V. APPROXIMATION ALGORITHMS

## A. Non-Polynomial-Time Approximation Scheme

We start by defining a non-polynomial-time algorithm returning a solution arbitrarily close to the optimal one. To this extent, we over-constrain the optimization problem, allowing only a single class to have a fractional threshold in its routing policy. This additional constraint is likely to worsen the solution quality (see Property IV.3) but it allows us to provide a combinatorial version of the optimization problem. Indeed, once all the classes except one have been assigned integer policies, the potentially fractional policy of the remaining class is univocally determined either by the policy that consumes all the remaining budget or the one that transmits until the last useful time slot (see Property IV.1). To have more precision, we split each slot of length  $\Delta$  in sub-units of length  $\epsilon\Delta$  where, for simplicity,  $\frac{1}{\epsilon} \in \mathbb{N}$ .

We solve optimally this over-constrained problem by enumeration. We enumerate all the feasible threshold policies and we select the best one (see Property IV.2). We report in Algorithm 1 the necessary steps. At Step 1, the algorithm initializes  $F^*$ —the variable with the value of the best found solution—to be zero. Let us denote  $\mu_{-c}$  the forwarding policy of all the classes except class  $c$ . If the budget cannot be entirely consumed, then the optimal policy is to assign  $h_c = (K-1)/\epsilon$  to each class  $c$  (Steps 2–3). Otherwise, the algorithm enumerates all the classes  $c$ , and for each class  $c$  it enumerates all the policies  $\mu = (\mu_c, \mu_{-c})$  s.t.  $\mu_{-c}$  is integer and budget  $\Psi$  is entirely consumed (Steps 5–6). Finally, we keep trace of the best policy found so far. We now describe an efficient scheme to enumerate all and only the feasible policies  $\mu = (\mu_c, \mu_{-c})$  s.t.  $\mu_{-c}$  is integer and budget  $\Psi$  is entirely consumed in Step 6. First, we build a lexicographic (strict) order  $\succ_L$  over  $C_{-c} = C \setminus c$  and we scan lexicographically the classes in  $C_{-c}$ . Then, for each  $c' \in C_{-c}$  we determine the range  $I_{c'}$  of feasible values for  $h_{c'}$  on the basis of the policies assigned to the classes  $c''$  preceding  $c'$  in the lexicographic order (i.e.,  $c'' \succ_L c'$ ):

$$I_{c'}(\mu_{-c'}) = \left\{ \max \left\{ 0, \min \left\{ \frac{K-1}{\epsilon}, \lceil r_{c'}(\bar{\mu}_{-c'}) \rceil \right\} \right\}, \dots, \min \left\{ \frac{K-1}{\epsilon}, \lfloor r_{c'}(\underline{\mu}_{-c'}) \rfloor \right\} \right\},$$

$$\bar{\mu}_{-c'} = \begin{cases} \bar{\mu}_{c''} = \mu_{c''} & c'' \succ_L c' \\ \bar{\mu}_{c''} : h_{c''} = \frac{K-1}{\epsilon} & c' \succ_L c'' \\ \bar{\mu}_c : h_c = \frac{K-1}{\epsilon} & \end{cases},$$

$$\underline{\mu}_{-c'} = \begin{cases} \underline{\mu}_{c''} = \mu_{c''} & c'' \succ_L c' \\ \underline{\mu}_{c''} : h_{c''} = 0 & c' \succ_L c'' \\ \underline{\mu}_c : h_c = 0 & \end{cases},$$

$\bar{\mu}_{-c'}$  is the policy in which the policies of all the classes  $c''$  with  $c'' \succ_L c'$  are given by  $\mu$  and for all the other classes  $c''$  with  $c' \succ_L c''$  the threshold is the maximum one (i.e.,  $\frac{K-1}{\epsilon}$ ) and therefore the energy budget that can be used for class  $c'$  is minimized, while  $\underline{\mu}_{-c'}$  is the policy in which the policies of all the classes  $c''$  with  $c'' \succ c'$  are given by  $\mu$  and for all the other classes  $c''$  with  $c' \succ c''$  the threshold is the minimum one (i.e., 0) and therefore the energy budget that can be used for class  $c'$  is maximized.

Function  $r_{c'}(\mu_{-c'})$  returns the maximum threshold  $h_{c'}$  for class  $c'$  given the policy  $\mu_{-c'}$  of all the other classes subject to the energy budget constraint and can be computed as follows:

- initially we compute in closed form:

$$r_{c'}(\mu_{-c'}) = -\frac{\log \left( 1 + \frac{A - \Psi + B}{\rho_{c'} N_{c'}} \right)}{\lambda_{c'} \Delta \epsilon}$$

where

$$A = \sum_{c'' : c'' \neq c'} \rho_{c''} N_{c''} \left( 1 - e^{-\lambda_{c''} \Delta \epsilon \sum_{k=0}^{\frac{K-1}{\epsilon}} \mu_{c''}(k)} \right),$$

$$B = \sum_{\omega \in \Omega} \sum_{k=0}^{\frac{K-1}{\epsilon}} \beta_{\omega} \cdot \left( 1 - \prod_{c'' \in C_{\omega} : c'' \neq c'} (1 - \mu_{c''}(k)) \right),$$

the above  $r_{c'}(\mu_{-c'})$  is the maximum threshold  $h_{c'}$  for  $c'$  in the case in which  $h_{c'}$  does not affect the signaling costs due to the technology used by class  $c'$ ; this happens when  $\lceil \epsilon r_{c'}(\mu_{-c'}) \rceil \leq \max_{c'' : c'' \in C_{\omega}, c'' \neq c'} \{ \lfloor \epsilon h_{c''} \rfloor \}$  where  $c' \in C_{\omega}$ , that is, when some class adopting the same technology of class  $c'$  transmits for the whole slot of duration  $\Delta$  that is the last slot in which class  $c'$  transmits;

- if, solving the above closed form, we obtain  $\lceil \epsilon r_{c'}(\mu_{-c'}) \rceil > \max_{c'' : c'' \in C_{\omega}, c'' \neq c'} \{ \lfloor \epsilon h_{c''} \rfloor \}$  where  $c' \in C_{\omega}$ , then the policy of class  $c'$  affects the signaling costs due to the technology adopted by  $c'$ . Thus, we can compute  $r_{c'}(\mu_{-c'})$  by solving the following equation:

$$\rho_{c'} N_{c'} (1 - e^{-\lambda_{c'} \epsilon \Delta r_{c'}(\mu_{-c'})}) + \sum_{\omega \in \Omega : c' \notin \omega} \sum_{k=0}^{\frac{K-1}{\epsilon}} \beta_{\omega} \cdot \left( 1 - \prod_{c'' \in C_{\omega}} (1 - \mu_{c''}(k)) \right) + \sum_{k=0}^{\frac{K-1}{\epsilon}} \beta_{\omega} \cdot \left( 1 - \prod_{c'' \in C_{\omega} : c'' \in \omega, c'' \neq c'} (1 - \mu_{c''}(k)) \cdot (1 - r_{c'}(\mu_{-c'})) \right) + A - \Psi = 0,$$

by means of the Newton algorithm. Notice that the above function has only one variable,  $r_{c'}(\boldsymbol{\mu}_{-c'})$ , and it is strictly monotonically increasing. Therefore the Newton algorithm can be used to obtain an arbitrarily accurate approximation of the exact solution.

Once the previous steps are done, for every element in  $I_{c'}(\boldsymbol{\mu}_{-c'})$ , we assign it to  $h_{c'}$  and go to the next class according to the lexicographic order. Finally, once the policies of all the classes  $c' \in C_{-c}$  have been assigned, the policy of  $c$  is easily given by  $h_c = r_c(\boldsymbol{\mu}_{-c})$ .

**Theorem V.1.** *The above scheme enumerates all and only the feasible policies consuming exactly the budget in which at most one  $h_c$  is fractional.*

*Proof:* We need to prove that: (i) all the policies except  $\boldsymbol{\mu}_c$  are integer, (ii) the budget is exactly consumed, and (iii) all and only the feasible policies are enumerated. The first two points are trivial by construction (given that the policy of  $c$  is the only potentially non-integer and is computed as the policy that consumes the budget given the policies of all the other classes). To prove the third point, we observe that  $I$  is always a well-defined range. Indeed,  $\lfloor r_{c'}(\boldsymbol{\mu}_{-c'}) \rfloor$  returns the largest  $h_{c'}$  that consumes exactly the remaining budget given the budget consumed by all the classes preceding  $c'$  in the lexicographic order. Assigning a policy larger than  $\min \left\{ \frac{K-1}{\epsilon}, \lfloor r_{c'}(\boldsymbol{\mu}_{-c'}) \rfloor \right\}$  violates the budget constraint or violates the deadline  $\tau$ . If the policies assigned to the previous classes are feasible, then  $\lfloor r_{c'}(\boldsymbol{\mu}_{-c'}) \rfloor$  is always non-negative. As well,  $\lfloor r_{c'}(\boldsymbol{\mu}_{-c'}) \rfloor$  returns the smallest  $h_{c'}$  that consumes exactly the remaining budget given the budget consumed by all the classes preceding  $c'$  in the lexicographic order and assuming that the classes that succeed transmit all the slots. Assigning a policy smaller than  $\max \left\{ 0, \min \left\{ \frac{K-1}{\epsilon}, \lfloor r_{c'}(\boldsymbol{\mu}_{-c'}) \rfloor \right\} \right\}$  does not allow one to consume entirely the budget. Thus, by construction, for each policy assigned to class  $c'$  belonging to  $I$ , it is always possible to find a feasible policy for the succeeding classes. ■

The number of policies enumerated by Algorithm 1 is exponential in  $C$ , being  $O\left(\left(\frac{K-1}{\epsilon}\right)^{|C|-1}\right)$ . We state the following result on the optimality loss of the solution found by Algorithm 1.

**Theorem V.2.** *Let  $\tilde{F}_D$  be the value of the solution returned by Algorithm 1 and  $F_D^*$  the value of the optimal solution, then we have*

$$\frac{\tilde{F}_D}{F_D^*} \geq \frac{1 - \left(\frac{1}{2}\right)^{\frac{K-1}{\epsilon}}}{1 - \left(\frac{1}{2}\right)^{|C|\frac{K-1}{\epsilon}}}.$$

*Proof:* Call  $\boldsymbol{\mu}^*$  the optimal policy profile and call  $\tilde{\boldsymbol{\mu}}_c$  the policy profile in which  $\tilde{h}_{c'} = \lfloor h_{c'}^* \rfloor$  for all  $c' \neq c$  and  $\tilde{h}_c = h_c^*$ . Obviously,  $F_D^* \geq F_D(\tilde{\boldsymbol{\mu}}_c, \frac{K-1}{\epsilon})$  and  $F_D^* \geq \tilde{F}_D \geq \max_{c'} \{F_D(\tilde{\boldsymbol{\mu}}_c, \frac{K-1}{\epsilon})\}$ . This is because  $\tilde{\boldsymbol{\mu}}_c$  is a feasible policy profile in which at most one policy is fractional that is not assured to consume exactly the budget. We can write a lower bound for  $F_D(\tilde{\boldsymbol{\mu}}_c, \frac{K-1}{\epsilon})$  as:

$$\begin{aligned} F_D\left(\tilde{\boldsymbol{\mu}}_c, \frac{K-1}{\epsilon}\right) &= \\ &= 1 - \prod_{c' \in C} \prod_{k=0}^{\frac{K-1}{\epsilon}} X_{c',k}^*(\lambda_{c'} \in \Delta, \tilde{\boldsymbol{\mu}}_c) \geq 1 - \prod_{k=0}^{\frac{K-1}{\epsilon}} X_{c,k}^*(\lambda_c \in \Delta, \tilde{\boldsymbol{\mu}}_c) \end{aligned}$$

By using such lower bound over  $F_D(\tilde{\boldsymbol{\mu}}_c, \frac{K-1}{\epsilon})$ , we can write:

$$\frac{\tilde{F}_D}{F_D^*} \geq \max_c \left\{ \frac{1 - \prod_{k=0}^{\frac{K-1}{\epsilon}} X_{c,k}^*(\lambda_c \in \Delta, \boldsymbol{\mu}^*)}{1 - \prod_{c' \in C} \prod_{k=0}^{\frac{K-1}{\epsilon}} X_{c',k}^*(\lambda_{c'} \in \Delta, \boldsymbol{\mu}^*)} \right\}$$

since, given  $\tilde{\boldsymbol{\mu}}_c$  and  $\boldsymbol{\mu}^*$ , we have  $\tilde{h}_c = h_c^*$ . Thus, we are interested in:

$$\min_c \max_c \left\{ \frac{1 - \prod_{k=0}^{\frac{K-1}{\epsilon}} X_{c,k}^*(\lambda_c \in \Delta, \boldsymbol{\mu}^*)}{1 - \prod_{c' \in C} \prod_{k=0}^{\frac{K-1}{\epsilon}} X_{c',k}^*(\lambda_{c'} \in \Delta, \boldsymbol{\mu}^*)} \right\}$$

where the minimization is over all the parameters. Although the definition of  $X^*$  is intricate, a bound can be derived disregarding the exponential nature of all the  $X^*$  and considering them as arbitrary values in  $[0, 1]$ . In this case, for reasons of symmetry, the values that minimize the maximum ratio prescribe  $X_{c,k}^* = \frac{1}{2}$  for all  $c$ . This leads to the bound stated in the theorem. ■

Notice that the theoretical lower bound does not depend on whether the signaling costs are present. The worst case is when  $K = 2$  and  $|C| \rightarrow \infty$ , obtaining a ratio of  $1 - \frac{1}{2}^{\frac{1}{\epsilon}}$ . However, it can be observed that the worst case ratio goes to one exponentially in  $\frac{1}{\epsilon}$ . Thus we can obtain a good approximation ratio with a small value of  $\frac{1}{\epsilon}$ , e.g., the theoretical lower bound over the approximation ratio is about  $1 - 10^{-4}$  when  $\frac{1}{\epsilon} = 10$ . Algorithm 1 is an approximation scheme (AS), given that the ratio goes to one as  $\epsilon$  goes to zero.

## B. Polynomial-Time Approximation Algorithm

In this section, we discuss a heuristic approach to approximate the optimal policy in polynomial time. We restrict our attention to threshold policies and we devise an iterative method that incrementally constructs the policy  $\boldsymbol{\mu}$  by increasing, at each iteration, the threshold for some class. Algorithm 2, reports the formal steps.

---

### Algorithm 2. $i$ -greedy construction

---

- 1:  $h_1, \dots, h_{|C|} \leftarrow 0$
  - 2:  $F^* \leftarrow 0$
  - 3: **while** Constraint in Eq. (6) is satisfied **do**
  - 4:   **for** every class  $c$  **do**
  - 5:      $\hat{h}_c \leftarrow h_c + \min\{1, r_c(\boldsymbol{\mu}_{-c}) - h_c\}$
  - 6:      $\delta_c \leftarrow G_i(F^*, h_1, \dots, \hat{h}_c, \dots, h_{|C|})$
  - 7:      $h_{c^*} \leftarrow \hat{h}_{c^*}$ , where  $c^* = \arg \max_{c \in C} \{\delta_c\}$
  - 8:    $F^* \leftarrow F^* + \delta_{c^*}$
- 

Since we focus only on threshold policies, the algorithm assumes that  $\boldsymbol{\mu}$  is uniquely determined by class thresholds. So, each time a modification is made to a threshold  $h_c$  we implicitly

assume that  $\mu$  changed accordingly and *vice versa*. Algorithm 2 works on the same discrete-time representation we introduced above, where each time slot has a temporal length of  $\epsilon\Delta$ . It starts from an initial empty policy where thresholds are set to 0 for each class (Step 1). Then, at each iteration, it evaluates with an objective function  $G_i$  each of a set of local modifications of  $\mu$  (Steps 5–6) and then applies the one maximizing such function (Steps 8–9). A local modification of  $\mu$  involves a single class  $c$  and is represented by an increment of  $h_c$  of the quantity  $\min\{1, r_c(\mu_{-c}) - h_c\}$ , that is, one more slot or, in case the residual budget would not be enough, the fractional part that would deplete the budget. In general, the objective function  $G_i$  is defined as a *marginal gain* between the current solution value and the one obtained after a modification would be applied. This is the reason why, besides class thresholds,  $G_i$  takes as argument the value  $F^*$ . Such variable represents the current solution value computed by iteratively adding up the marginal gains of those modifications selected by the algorithm (Step 9). We propose two different instantiations of the objective function that we will call  $G_1$  and  $G_2$  and we will speak of a first and a second version of the algorithm, respectively. In each of the two versions the computation of  $G_i$  will be doable in  $O(|C|)$  time, thus the worst case time complexity of Algorithm 2 is  $O(\frac{K-1}{\epsilon}|C|^3)$ .

*First Version, Locally Optimizing  $F_D$* : In the first version of Algorithm 2, we define  $G_1$  in order to obtain the maximization of the marginal gain of  $F_D$ , i.e., the delivery probability. Formally, we have that at Step 6 it holds that  $G_i = G_1$  where  $G_1(F^*, h_1, \dots, h_{|C|}) = F_D(\mu, (K-1)/\epsilon) - F^*$ . (Recall that  $\mu$  is assumed to be the unique policy obtained from thresholds  $h_1, \dots, h_{|C|}$ ). Under such definition,  $\delta_c$  represents the benefit, in terms of delivery probability, that an additional (integer or fractional) time slot for class  $c$  would introduce at the current iteration. By exploiting a result presented in [29] we are able to provide a bound on the solution quality obtained with this version of the greedy algorithm. The result we will exploit can be formalized as follows (see [29] for a complete derivation).

**Theorem V.3 [From [29]]** *Given a ground set  $\Theta$ , a set function  $\Phi : 2^\Theta \rightarrow \mathbb{R}$ , and a positive integer  $W \in \mathbb{N}^+$ , let us consider the problem of finding  $S^* = \arg \max_{S \subseteq \Theta, |S| \leq W} \{\Phi(S)\}$ . Then if  $\Phi$  is submodular, we have that for every integer  $0 \leq l \leq W$ ,  $\Phi(S_l) \geq (1 - e^{-l/W})\Phi(S^*)$ , where  $S_l \in \Theta$  is the set built after  $l$  iterations of the following greedy element-selection rule*

$$S_i = \begin{cases} \emptyset & \text{if } i=0 \\ S_{i-1} \cup \arg \max_{s \in \Theta} \Phi(S_{i-1} \cup \{s\}) & \text{else} \end{cases} \quad (7)$$

Theorem V.3 states that greedily maximizing a submodular set function introduces a bounded suboptimality. Eventually, the bound converges to  $(1 - \frac{1}{e})$  ( $\approx 0.63$ ) when  $l = W$ , that is, when the maximum number of selections allowed by the cardinality constraint is made.

In order to apply the above result to Algorithm 2, we need to show that the problem of finding an optimal integer policy can be expressed as the maximization of a submodular set function subject to a cardinality constraint. This similarity can be shown by mapping our problem to the following set-based formal interpretation. Let us assume that each element in the ground set

$\theta \in \Theta$  is a pair  $(c, k)$  where  $c \in C$  and  $k \in \{1, \dots, \frac{K-1}{\epsilon}\}$ . That is, each element of  $S$  prescribes that a specific class transmits in a specific slot. Then, every subset  $S \subseteq \Theta$  can be uniquely associated with an integer (not necessarily threshold-based) policy  $\mu^S$ . Indeed, a unique correspondence between  $S$  and  $\mu^S$  can be obtained by the following construction rule:

$$\mu_c^S(k) = \begin{cases} 1 & (c, k) \in S \\ 0 & \text{else} \end{cases}$$

Therefore, the objective function for a policy  $\mu^S$  can be rewritten as a set function by operating the following simple assignment:  $\Phi(S) = F_D(\mu^S, (K-1)/\epsilon)$ .

The second necessary step is to derive a cardinality constraint to define the problem's feasibility region. In our problem, the feasibility of a policy is determined by the budget limit, namely by the constraint in Eq. (6). For this reason, ideally one would like to find a  $W$  such that  $|S| > W$  if and only if  $\mu^S$  violates the constraint in Eq. (6). However, it can be easily shown that budget feasibility cannot be directly expressed with a cardinality constraint. The reason is straightforward. The budget of a policy does not solely depend on the number of transmitting slots, but also on how those slots are distributed among the different classes. Nevertheless, a necessary (not sufficient) cardinality upper bound can be determined via the following theorem.

**Theorem V.4.** *Any feasible threshold integer policy cannot assign full probability of transmission to more than  $W = \min\{\max_c \{r_c(\mu^\emptyset)\}, \frac{K-1}{\epsilon}\}$ , where  $\mu^\emptyset$  is the empty policy.*

*Proof:* Let us assume that  $\hat{c} = \arg \max_c \{r_c(\mu^\emptyset)\}$ . Then, consider a threshold policy  $\mu^S$  where  $|S| > W$ . If  $\mu^S$  is budget-feasible then, by definition, the policy obtained in this way should be feasible too: for every  $(c, k) \in S$  where  $c \neq \hat{c}$  substitute  $(c, k)$  with  $(\hat{c}, h_{\hat{c}} + 1)$ . However, by definition of  $W$  such a policy cannot be budget-feasible. ■

Under the above assumption, the optimal integer policy problem can be associated, up to a relaxation of the feasibility constraint, with the maximization of the set function  $\Phi(S)$ , subject to  $|S| \leq W$ . By relaxing the cardinality constraint we can still derive an approximation bound even though we cannot guarantee its tightness. In the next step we show  $F_D$  submodularity.

**Property V.5.** *The set function  $\Phi$  is submodular w.r.t.  $\Theta$ .*

*Proof:* First, let us consider a setting with a single class. From Property IV.2, we can focus only on threshold policies and rewrite  $\Phi$  as a function of  $h$ , namely the threshold value (this value, in general, can be non-integer). Then it can be easily shown that  $\Phi(h)$  is a concave function since the Hessian matrix has strictly negative eigenvalues. Given a function  $f : \mathbb{N} \rightarrow \mathbb{R}^+$ , then  $f(|S|)$  is submodular on the subsets  $S$  of an arbitrary set  $\Omega$  if and only if  $f$  is concave [30]. We can then conclude that  $\Phi$  is submodular in the case of a single class. Let us now show submodularity for the case with two classes. Let us denote with  $\Delta\Phi(S|e)$  the marginal gain of  $\Phi$  obtained by adding the element  $e$  to the set  $S$ , namely adding a transmitting slot to some class to the policy  $\mu^S$ . For submodularity to hold, we need to show that for every  $S_a$ ,



$S_b$ ,  $e$  such that  $S_a \subseteq S_b \subset \Omega$  and  $e \in \Omega \setminus S_b$  we have that  $\Delta\Phi(S_a|e) \geq \Delta\Phi(S_b|e)$ . By definition  $e$  adds a slot to a single class, let us assume without loss of generality that this class is  $c_1$ . Also, let us denote with  $\Phi_{c_i}$  our function computed as if  $c_i$  were the only present class. Then we have  $\Delta\Phi(S_a|e) = [1 - (1 - (\Phi_{c_1}(S) + \Delta\Phi_{c_1}(S_a|e)))(1 - \Phi_{c_2}(S))] - [1 - (1 - \Phi_{c_1}(S))(1 - \Phi_{c_2}(S))] = (1 - \Phi_{c_2}(S_a))\Delta\Phi_{c_1}(S_a|e)$  and, analogously,  $\Delta\Phi(S_b|e) = (1 - \Phi_{c_2}(S_b))\Delta\Phi_{c_1}(S_b|e)$ . Since  $\Delta\Phi_{c_1}(S_a|e) \geq \Delta\Phi_{c_1}(S_b|e)$  by submodularity of  $\Phi_{c_1}$  and  $\Phi_{c_2}(S_b) \geq \Phi_{c_2}(S_a)$  by  $\Phi_{c_2}$  monotonicity, we have that  $\Phi$  is submodular. The same reasoning can be extended to an arbitrary number of classes. ■

Theorem V.3 can be applied by showing that Algorithm 2 corresponds to the greedy element–selection rule reported in Eq. (7). The rule of Eq. (7), when applied to the integer policy problem, proceeds by locally optimal appends in the same way that Algorithm 2 does. Hence, we have:

**Theorem V.6.** *Let us denote with  $S^*$  the policy returned by Algorithm 1 and with  $S_l^1$  the policy constructed by Algorithm 2 (version 1) after  $l$  iterations. We then have that  $F_D(S_l^1, K) \geq (1 - e^{-l/W})F_D(K, S^*)$ .*

*Proof:* The inequality stated in the theorem follows immediately from the following two properties. First, by applying Theorem V.3 to Algorithm 2 (version 1) we have that  $F_D(S_l^1, K) \geq (1 - e^{-l/W})F_D(S^*, K)$ . Second, since  $\hat{S}^*$  is the optimal solution of a relaxed version of the integer policy problem, it holds that  $F_D(S^*, K) \leq F_D(\hat{S}^*, K)$ . ■

The previous theorem, provides an *online* bound on the solution quality, being it dependent on the number of iterations the algorithm will succeed in performing without violating the actual budget constraint. An *offline* guarantee can be given by computing the minimum number of slot  $s_c$  to be assigned to each class  $c$ . This number can be computed by setting  $\mu_{c'}(i) = 1 \forall i \ 0 \leq i \leq K, c' \neq c$  and computing the maximum number of time slots during which  $c$  can transmit without saturating the budget.

**Corollary V.7.** *For any solution  $S^1$  obtained with Algorithm 2 (version 1) we have that:*

$$F_D(S^1, K) \geq \left(1 - e^{-\sum_{c \in C} s_c / W}\right) F_D(S^*, K).$$

*Second Version, Normalizing  $G_1$  with Budget Costs:* The second version of our algorithm adopts objective function  $G_2$ , obtained by normalizing  $G_1$  with the budget cost that a local modification (the additional time slot) would introduce. As a consequence, given a local modification of  $\mu$  as defined above, here  $\delta_c$  represents a ratio between the marginal gain in the delivery probability obtained if applying such modification and the additional transmission costs that would be paid. For simplicity, we do not consider signaling costs, also because extending our approximate analysis by including them does not seem straightforward. Under the assumption that no signaling costs are present and that we deal with threshold policies, each transmission has an independent cost and the budget spent by a policy  $S$  is given by  $\psi(S) = \sum_{(c,k) \in S} \rho_c N_c e^{-\lambda_c \Delta(k-1)} (1 - e^{-\lambda_c \Delta})$  and, consequently,  $G_2(\mu) = \frac{G_1(\mu)}{\psi(\{(c, h_c + 1)\})}$ .

If we modify the rule in Eq. (7) by normalizing the objective function by the budget cost for each candidate element, we can again show the equivalence between the new rule and Algorithm 2 (version 2). As a consequence, we can again resort to a result presented in [29] and provide a quality bound on the solution obtained with the combination of the two versions of Algorithm 2 when signaling costs are not considered.

**Theorem V.8.** *If no signaling costs are present, then it holds that*

$$\begin{aligned} & \max\{F_D(\hat{S}^1, K), F_D(\hat{S}^2, K)\} \\ & \geq \frac{1}{2} \left(1 - \frac{1}{e}\right) \max_{S \subseteq \Theta: \psi(S) \leq \Psi} \{F_D(S, K)\} \end{aligned}$$

*Proof:* The proof follows immediately by the consideration made above and a straightforward adaptation of results presented in [29]. ■

## VI. EXTENSION TO MULTIHOP ROUTING

We initially describe how the formulation of the problem changes in the case of  $\ell$ –hop routing with  $\ell \geq 3$ . At first, we need some assumptions about the functioning of the routing protocol. More precisely, we assume that: during a time slot, if a mobile node contacts both the source and other mobile nodes carrying the packet, then the mobile node receives the packet directly from the source, and, if a mobile node has received a packet from another mobile node before contacting the source, then the mobile node does not receive the packet also from the source. Furthermore, as assumed in the case of two–hop routing, if a mobile node has dropped a packet, it will never get the packet again in future. With multiple classes, we have two different scenarios: the one in which a node of class  $c$  cannot transmit the packet to a node of class  $c'$  with  $c \neq c'$  and the one in which it can do. Our aim is to find the best (approximate) transmission policy of the source, given the transmission policies of the mobile nodes (characterized by time–out  $t_c$ ).

We focus on the extension of Algorithm 2 (although also the extension of Algorithm 1 is possible, it requires long calculations and it is less significant requiring exponential time). The extension can be simply obtained by providing a procedure to calculate  $X_{c,k}$  necessary to compute objective function  $F$  and a procedure to calculate the cost of a transmission policy. Indeed, the proof of submodularity and the bounds derived in Section V-B hold also with  $\ell$ –hop routing.

We focus on the calculation of  $X_{c,k}$ . We initially describe how, in the basic case of two–hop routing without time–out  $t_c$ , variable  $X_{c,k}$  can be computed. This is useful for presenting the general case. We denote by  $X_{c,k}(i, \mu)$  the probability that, given  $\mu$ , at slot  $k$  there are  $i$  nodes of class  $c$  directly infected by the source. The term  $X_{c,k}(i, \mu)$  can be computed as:

$$\begin{aligned} X_{c,k}(i, \mu) &= \sum_{h=0}^i X_{c,k-1}(h, \mu) (1 - Q_{c,k-1,k}(\mu))^{i-h} \\ & \quad (Q_{c,k-1,k}(\mu))^{n-h} \binom{N_c - h}{i - h} \end{aligned}$$

It can be observed that the computation of  $X_{c,k}(i, \mu)$  requires time and space  $O(|C|N_c K)$ .



We show now how, in the case of two-hop routing with time-out  $t_c$ , variables  $X_{c,k}$  and  $Y_{c,k}$  can be computed. With abuse of notation, we denote by  $X_{c,k}(i, \boldsymbol{\mu}|n)$  the term  $X_{c,k}(i, \boldsymbol{\mu})$  when the number of mobile nodes is  $n$ , potentially different from  $N_c$ . We have:

$$Y_{c,k}(i, \boldsymbol{\mu}) = \begin{cases} X_{c,k}(i, \boldsymbol{\mu}|N_c) & k \leq t_c \\ \sum_{j=0}^{N_c-i} X_{c,k-t_c}(j, \boldsymbol{\mu}|N_c) X_{c,t_c}(i, \boldsymbol{\mu}|N_c - j) & t_c < k \end{cases}$$

It can be observed that the computation of  $Y_{c,k}(i, \boldsymbol{\mu})$  requires the computation of  $X_{c,k}(i, \boldsymbol{\mu}|n)$  for any  $n \in \{1, \dots, N_c\}$ , and therefore it requires time and space  $O(|C|(N_c)^2 K)$ .

Now we focus on the case with  $\ell$ -hop routing. For the sake of clarity, we present the case with only one class and without time-out  $t_c$ , the extension to the general case is discussed below. Initially, denote by  $Z_k(i_1, i_2, \dots, i_{\ell-1}, \boldsymbol{\mu})$  the probability that, given  $\boldsymbol{\mu}$ , at slot  $k$  there are  $i_1$  nodes with a 1-hop infection,  $i_2$  nodes with a two-hop infection, and so on. Variable  $Z_k$  is defined in Eq. (8), shown at the bottom of the page.

It can be observed that computing  $Z_k(i_1, i_2, \dots, i_{\ell-1}, \boldsymbol{\mu})$  requires time and space  $O((N_c)^\ell K)$ . The exponential size in  $\ell$  cannot be circumvented, being necessary to keep trace of all the possible configurations of mobile nodes at different hops that are exponential in  $\ell$ . Then,  $X_k(i, \boldsymbol{\mu})$  is:

$$X_k(i, \boldsymbol{\mu}) = \sum_{i_1=0}^i \sum_{i_2=0}^{i-i_1} \cdots \sum_{i_{\ell-2}=0}^{i-\sum_{w=1}^{\ell-3} i_w} Z_k \left( i_1, i_2, \dots, i_{\ell-2}, i - \sum_{w=1}^{\ell-2} i_w, \boldsymbol{\mu} \right)$$

where  $Q_{k-1,k}^{\text{mob}}$  is the probability that two mobile nodes do not have any contact between slots  $k-1$  and  $k$ . In case of  $\ell$ -hop routing with  $|C|$  classes without time-out  $t_c$  we need a structure  $Z_{c,k}(i_1, i_2, \dots, i_{\ell-1}, \boldsymbol{\mu})$  of size  $O(|C|(N_c)^\ell K)$  if a node of class  $c$  cannot transmit to a node of class  $c'$  with  $c \neq c'$ , whereas we need a structure  $Z_{c,k}(i_{1,1}, \dots, i_{|C|,\ell-1}, \boldsymbol{\mu})$  of size  $O(n^{|C|} K)$  if a node of a given class  $c$  can transmit the packet to nodes of different classes than  $c$ . Finally, when mobile nodes have a time-out  $t_c$ , it is necessary an extra multiplicative cost of  $O(\sum_c N_c)$ .

Finally, we now focus on the energy cost of a transmission policy. The energy cost depends only on the number of mobile nodes directly infected by the source. That is:

$$\rho_c \sum_{i_1=0}^i \sum_{i_2=0}^{i-i_1} \cdots \sum_{i_{\ell-2}=0}^{i-\sum_{w=1}^{\ell-3} i_w} Z_k \left( i_1, i_2, \dots, i_{\ell-2}, i - \sum_{w=1}^{\ell-2} i_w, \boldsymbol{\mu} \right).$$

$$Z_k(i_1, i_2, \dots, i_{\ell-1}, \boldsymbol{\mu})$$

$$\begin{aligned} &= \sum_{h_1=0}^{i_1} \sum_{h_2=0}^{i_2} \cdots \sum_{h_{\ell-1}=0}^{i_{\ell-1}} Z_{k-1}(h_1, h_2, \dots, h_{\ell-1}, \boldsymbol{\mu}) \cdot (1 - Q_{k-1,k}(\boldsymbol{\mu}))^{i_1-h_1} (Q_{k-1,k}(\boldsymbol{\mu}))^{n-i_1-\sum_{j=2}^{\ell-1} h_j} \binom{N_c - i_1 - \sum_{j=2}^{\ell-1} h_j}{i_1 - h_1} \\ &\cdot \prod_{w=2}^{\ell} \left( 1 - (Q_{k-1,k}^{\text{mob}})^{h_w-1} \right)^{i_w-h_w} (Q_{k-1,k}^{\text{mob}})^{N_c - \sum_{v=1}^w i_v} \binom{N_c - \sum_{v=1}^{w-1} i_v - \sum_{v=w}^{\ell-1} h_v}{i_w - h_w} \end{aligned} \quad (8)$$

TABLE II  
PARAMETERS USED FOR EXPERIMENTS

<b>Delivery deadline</b> ( $\tau/\Delta$ )	25, 50, 100, 250 time units
<b>Network radius</b> ( $L$ )	350, 500, 750, 1000m
<b>Number of nodes</b> ( $N_c$ )	9, 15, 20
<b>Mobility profiles</b> ( $v_c$ )	pedestrians (1.5m/s), bicycles (6m/s), vehicles (9m/s)
<b>Technology</b>	ZigBee ( $\rho = 0.1989J$ , $\beta = 0.7204 \times 10^{-5}J$ , $R = 15m$ ) Bluetooth 4.0 ( $\rho = 0.1278J$ , $\beta = 0.1136 \times 10^{-5}J$ , $R = 50m$ ) WiFi Direct ( $\rho = 0.0642J$ , $\beta = 0.0392 \times 10^{-5}J$ , $R = 100m$ )

The extension of Algorithm 2 to  $\ell$  hops is then easy and involves only Steps 5 and 6. Informally, these steps are substituted as follows: given a class  $c$ ,  $h_c$  is increased by 1 and both  $F_D$  and the energy cost are computed as described above. If the remaining budget is smaller than the cost of increasing  $h_c$  by 1, then  $h_c$  is reduced to satisfy the budget by employing the Newton algorithm. Finally, values  $\delta_c$  and  $\hat{h}_c$  are returned.

## VII. PERFORMANCE EVALUATION

### A. Evaluation Setting

We generated instances by considering the discretized parameter space of Tab. II. The reference scenario is an urban area populated by mobile devices carried by pedestrians, bicycles, vehicles equipped with heterogeneous transmission technologies (ZigBee, Bluetooth 4.0. and WiFi Direct). We derive the values for  $\rho$  and  $\beta$  by considering the technical specifications of each technology and assuming an application scenario with a 5kB packet and  $\Delta = 10s$ . For simplicity, we assign the same number of users to each class.

Unless differently specified, we consider up to 3 classes and a discretization  $\epsilon \in \{1, 1/3, 1/5\}$ . This represents a good trade-off between accuracy and computational effort to evaluate our algorithms as shown, see Fig. 1, by the theoretical lower bound of the delivery probability (Theorem V.2) for different resolutions and numbers of classes. As it can be seen, a maximum resolution of  $\epsilon = 1/5$  is a reasonable choice to guarantee about 95% of the optimal solution quality without the burden of a prohibitive number of time slots. On the other side, by adopting a maximum number of 3 classes we obtain a case which is fairly close to the worst case (derived for an infinite number of classes) and that is computable by means of our grid algorithm (recall that our grid search requires a computing time exponential in the number of classes).

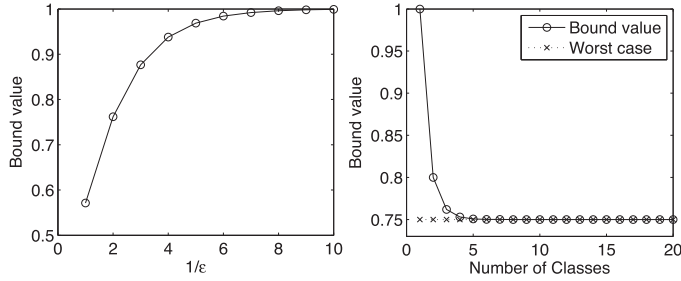


Fig. 1. Theoretical lower bound over solution's quality (Theorem V.2).

We use the following benchmarks for evaluating our algorithms in two-hop routing scenarios.

1) *Greedy on Arrival Rate*: it sorts the classes in descending order of  $\lambda_c$ , then it allocates all the possible budget to the classes from the first one to the last one. The rationale is that we expect that the larger the arrival rate the larger the delivery probability. The complexity of this algorithm is low: the policy can be found by solving at most  $|C|$  equations.

2) *Class-Independent Policies*: it searches for the optimal solution of an over-constrained problem in which the policies related to all the classes are the same, formally  $\mu(k) = \mu_c(k)$  for all  $c$ , and, when the policy is probabilistic, then either the source transmits to all the classes or it does not transmit at all. This leads to a new formulation of the budget constraint:

$$\sum_{c \in C} \rho_c N_c \cdot (1 - Q_{c,0,K}(\mu)) + \sum_{\omega \in \Omega} \sum_{k=0}^{K-1} \beta_\omega \left(1 - (1 - \mu(k))^{|\mathcal{C}_\omega|}\right) \leq \Psi$$

By Property IV.1, the optimal policy is such that the budget  $\Psi$  is completely consumed and therefore the above inequality holds with equality. Therefore, the optimization problem reduces to the problem of finding the policy that completely consumes the budget. Formally, interpreting the (class-independent) threshold  $h$  as a continuous variable, we can write:

$$g(h) = \sum_{c \in C} \rho_c N_c \cdot (1 - e^{-\lambda_c \Delta h}) + \sum_{\omega \in \Omega} \beta_\omega [h] + \sum_{\omega \in \Omega} \beta_\omega \left(1 - (1 - h + [h])^{|\mathcal{C}_\omega|}\right) - \Psi = 0$$

Function  $g$  is a single-variable function strictly monotonically decreasing in  $h$  and infinitely differentiable. Such a function admits only one zero, and therefore the above equation admits only one solution. Such a solution can be found (approximately) by using the Newton method that, due to the function property holding in this case, has a quadratic convergence speed (the number of correct digits roughly doubles in every iteration). Thus, we obtain an approximate solution of high quality within very short time.

3) *Upper Bound Over the Optimal Value*: an upper bound over the value of the optimal solution can be found by using a variation of the algorithm described in Section V-A. More precisely, we use Algorithm 1 to enumerate all the policies consuming entirely the budget and we change each policy rounding

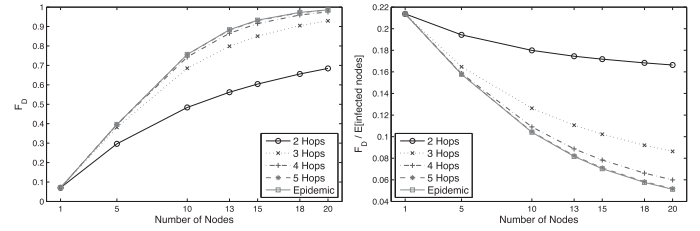


Fig. 2. Performance at different hops of Greedy construction (1). At left: delivery probability. At right: delivery probability divided by expected number of transmissions in the network.

each  $h_c$  to the smallest integer and then adding 1 for every  $c$ . Notice that these new policies violate the budget constraint. Among all these policies we find the one maximizing the delivery probability. Its value is an upper bound over the value of the optimal policy. In the following, we denote this value as  $UB$ . The proof sketch follows. Call  $\mu^*$  the optimal policy profile with (potentially fractional) thresholds  $h_c^*$ . Call  $\hat{\mu}$  a generic policy profile obtained as described above. It can be easily observed (it follows from the fact that, fixed the policies of all the classes but one, the policy of the remaining class that consumes entirely the budget is always one) that there always exists a policy profile  $\hat{\mu}$  such that  $\hat{h}_c \geq h_c^*$  for all  $c$ . Therefore, given that the objective function is strictly monotone in  $h_c$ , the objective value of  $\hat{\mu}$  is strictly better than the value  $\mu^*$ .

4) *Fluid Approximation*: We use the approach described in [2] based on fluid approximation to derive approximate routing policies.

## B. Comparing $l$ -hops Routing Policies

We apply our greedy algorithms when the number of hops is in  $\{1, 2, 3, 4, 5, \text{epidemic}\}$  to the simulation setting described above restricting the number of classes to be one. In this case, our algorithms return optimal solutions. We evaluate how the delivery probability and the ratio between delivery probability and the number of expected transmissions in the network vary as the number of hops varies. The first index provides a measure of the improvement of the objective function, while the second index provides a measure of efficiency between objective function and consumed energy. In all our simulations, we observed that the delivery probability increases in the number of hops and the increment decreases exponentially in the number of hops, achieving asymptotically the epidemic routing, while the ratio between delivery probability and expected number of transmissions decreases in the number of hops. We report the data of the most significant simulation in Fig. 2, in which we use WiFi Direct, bicycles,  $L = 500m$ ,  $K = 25$  and a variable number of nodes. This shows that two-hop routing provides the best trade-off between delivery probability and energy consumption, with a ratio of almost 5 w.r.t. the epidemic routing and a ratio of about 2 w.r.t. three-hop routing. From here on, we focus our performance evaluation on two-hop routing protocol.

## C. Algorithms Performance Analysis With Two Hops

Fig. 3 reports how  $F_D/UB$  varies as the values of the parameters  $\tau$ ,  $L$ ,  $N_c$  vary as summarized in Table II,  $|C| \in \{1, 2, 3\}$ ,

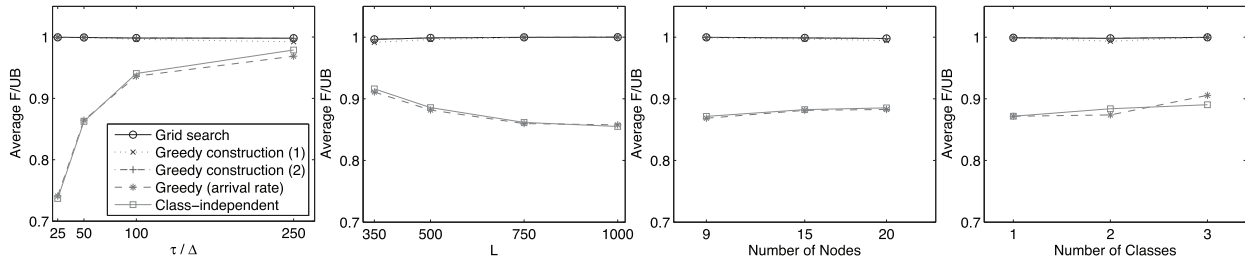


Fig. 3. Average  $F_D/UB$  w.r.t. different parameters at  $\frac{1}{\epsilon} = 5$ .

and  $\frac{1}{\epsilon} = 5$ . For each parameter, we average  $F_D/UB$  over the other instances sharing the same value for that parameter. It can be observed that grid search and greedy constructions obtain a remarkably better performance in each case when compared with the benchmarking greedy algorithms based on the arrival rate and the class-independent one. Not exploiting the knowledge about the different classes and solely considering the arrival rate turned out to achieve very similar performances. By increasing the value of  $\tau$ , it can be seen how this gap with the benchmarks shrinks, suggesting the intuition that when the deadline for packet delivery is large even simplistic policies are able to obtain good delivery probabilities. Another aspect that can be observed is that greedy constructions revealed to be quite effective for the tested cases, since they were able to obtain high performances comparable to the grid search. By increasing the value of  $L$ , it can be seen how this gap with the benchmarks increases, instead the gap keeps to be approximately constant as  $N_c$  and  $|C|$  vary. Interestingly, the approximation ratio of our algorithms is almost constant (i.e.,  $> 99\%$ ) w.r.t. all the parameters values.

A more detailed overview on how the performance (measured again as  $F_D/UB$ ) varies with  $\epsilon$  at different values of  $\tau$  is shown by the boxplots of Fig. 4. These graphs show the similarity in performance between the grid search and the greedy constructions algorithms. These last ones obtained worse performances for a limited number of outlier instances. Also it is evident how having finer resolutions remarkably improves the solution quality. As shown by the boxplots, the levels of statistical significance corroborate our claims.

The above results suggest that greedy constructions seem to be quite effective approaches to approximate the optimal policy requiring, at the same time, much lower computational effort than the grid search. In Fig. 5a, we show a comparison between computational times obtained with the grid search and the greedy construction algorithms respectively. In particular, we evaluated the algorithms' scalability when the number of classes grows. To obtain these results we fixed the values of some parameters ( $\epsilon = 1/3$ ,  $\tau = 100$ ,  $N_c = 10$ ,  $L = 500$ ) and we generated random mobility profiles and transmission technologies by uniformly sampling from the following intervals:  $R_c \in [15, 50]$ ,  $v_c \in [1, 15]$ ,  $\rho_i \in [0.05, 0.25]$ ,  $\beta_c \in [3 \times 10^{-7}, 8 \times 10^{-7}]$ . It is easy to see how grid search shows an exponential growth in time, while greedy construction proved to be much more efficient even for larger number of classes. Considering a deadline of 1 hour, grid search was not able to compute a solution for more than 4 classes, while greedy

construction managed to compute solution up to 800 classes. Notice that the time needed on average to find the best policy (i.e., 10s with  $|C| \leq 10$  and 100s with  $|C| \leq 100$ ) may be, in some scenarios, excessively long. Thus, to have a more accurate estimate of the overhead in real-world system, we implemented our approximation algorithms in C language, obtaining a compression of about 100 times (as observed in the literature for MATLAB vs. C for a number of applications). Such an overhead is compliant with the most application of DTNs.

Finally, we evaluate the accuracy of the fluid approximation approach. For each simulation setting described above, we run our approximation algorithms and the fluid approximation algorithm and we compare the solutions in terms of average  $F_D$ . In our results, we excluded the basic settings with  $|C| = 1$ , in which the fluid approximation is optimal as our approximation algorithms. As in other applications, fluid approximation provides an accurate approximation when the number of users/mobile nodes is large. More precisely, on average over the number of classes, the error (in percentage) of the fluid approximation w.r.t. our greedy algorithms behaves as  $6.44 + 31.01 \exp[-0.135 \sum_c N_c]$  with a confidence bound of 95%, providing thus an average error from about 35% for few mobile nodes to about 5% for 100 mobile nodes. This suggests that with 100 mobile nodes or less our algorithms, besides introducing suitability for any number of hops, can provide an important improvement w.r.t. the state of the art.

#### D. Two-Hop Routing Analysis

We now focus on how two-hop routing policies computed with our most viable method (greedy constructions) behave w.r.t. absolute and relative temporal deadlines, that is  $\tau$  and the time to live  $t_c$  for a particular class  $c$ . Indeed, together with budget requirements, temporal deadlines turned out to be the most sensible parameters, namely dimensions along which performance exposed remarkable variations.

Fig. 5b depicts how the delivery probability varies as  $\tau$  is set to increasing values for a number of budget settings. For such experiments we considered again three different classes (ZigBee, Bluetooth 4.0, and WiFi Direct associated with mobility profile in increasing speed as per Table II) with the same number of nodes (20), no packet discarding and the same relative budget scale we used in the previous section. The trends confirm the intuition by suggesting that strict delivery deadlines are a critical factor in worsening the expected performance of the computed policy. On the other side, the influence exercised



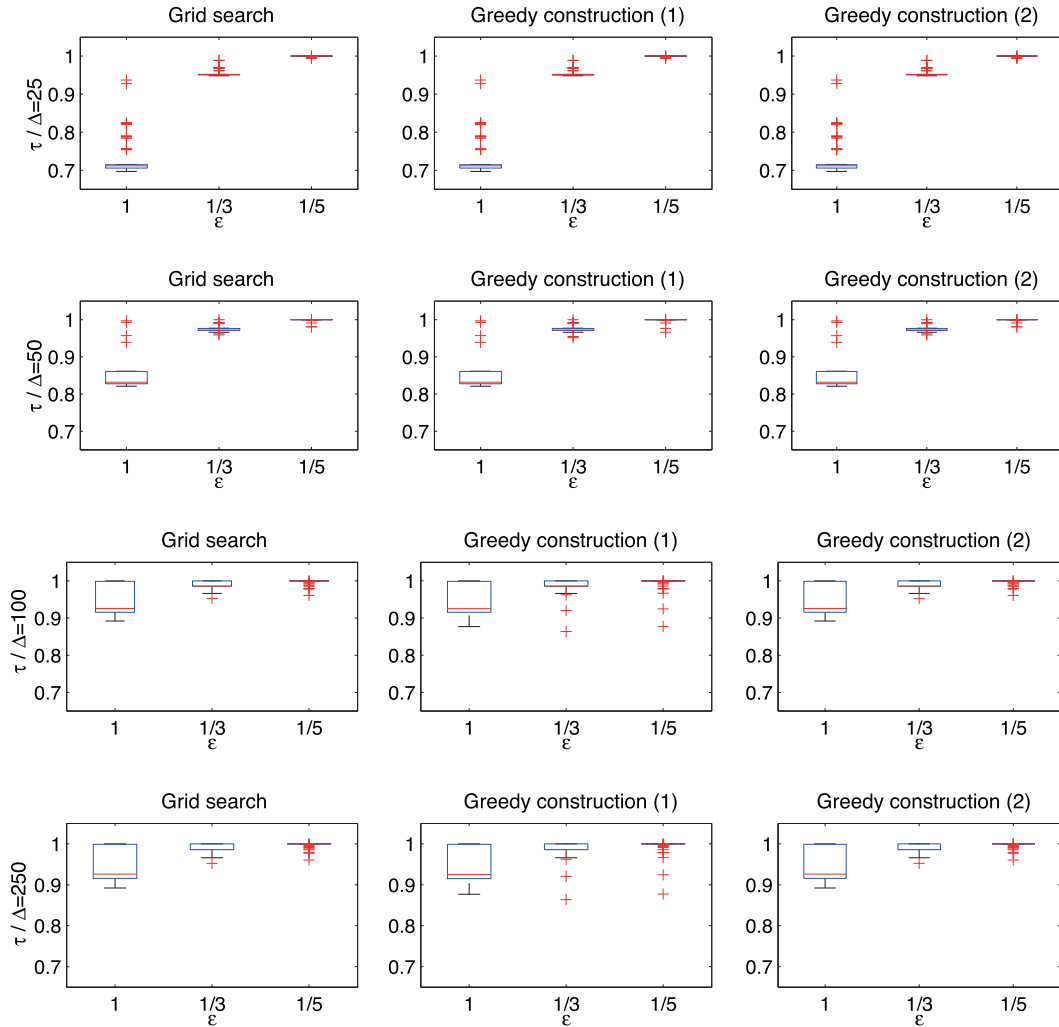


Fig. 4. Boxplots showing  $F_D/UB$  w.r.t.  $\tau$  for different algorithms.

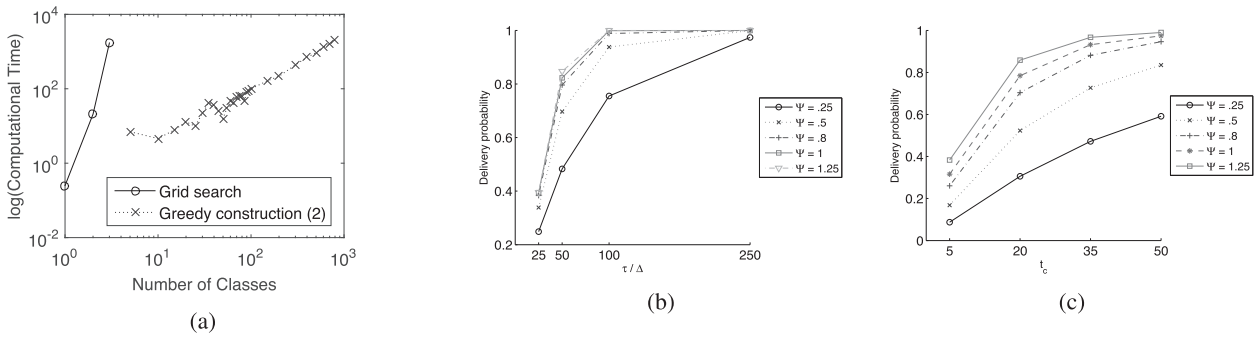


Fig. 5. Time (in seconds) scalability with the number of classes (a).  $F_D$  w.r.t. deadline (b) and packet’s time to live (c) under different budget constraints.

by the budget seems to muffle as its value becomes higher and higher.

In Fig. 5c we assess the impact of the packet’s time to live on the delivery probability. In this experiment we used the same three classes as before but we enabled the packet discarding behavior setting a given  $t_c$  equal for all the classes. We set  $\tau/\Delta = 50$  in order to disable packet discarding when  $t_c = 50$ . What can be observed from the figure is a trend similar to the one observed for the delivery deadline with the following

interesting difference. Here variations on the budget constraint seem to have a slightly stronger impact than before, suggesting that when nodes start to drop packets having extra units of budget can introduce non-negligible improvements in the performance.

Fig. 6a and 6b depict a qualitative evaluation of the policies returned by our algorithms. The same three classes as before are considered before and, for clarity, we selected an instance where  $\tau/\Delta = 10$  (similar trends could be observed in

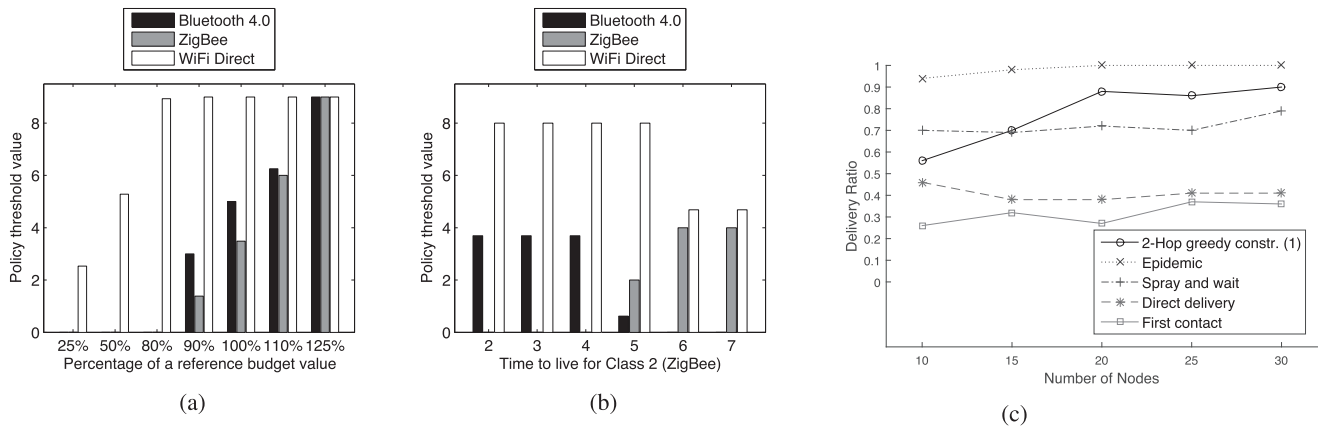


Fig. 6. Policy thresholds with different upper bounds on budget (a) and for different time to live values for one class (b). Comparisons with other routing schemes (c).

any other setting). In Fig. 6a, we consider a reference value for the budget upper bound  $\Psi$  and we show how the thresholds of the optimal policy (obtained with grid search) are distributed across the three different technologies. It can be observed how, by increasing the budget, the optimal policy tends to schedule transmissions with all the three technologies. When the budget gets smaller and smaller, then the policy tries to rely more on those technologies that have a longer communication range.

In Fig. 6b we show how policies change as different time to live values are adopted. We set  $t_{c_2} = 4$  for the class adopting Bluetooth 4.0 technology and  $t_{c_3} = 2$  for that using WiFi Direct. The time to live for the class using ZigBee varies as shown in the picture. As it can be seen, such class is not even exploited by the policy when its time to live is small, the other two classes are preferred. Interestingly, WiFi is assigned a higher threshold even though it has the lowest time to live (higher transmission range and node velocity provide some kind of overcompensation). As  $t_c$  increases, ZigBee starts to be included in the classes used by the policy until it is completely preferred over Bluetooth 4.0. Such trend demonstrates how there are settings where less profitable classes (in terms of speed and transmission range) can be preferred over better ones due to a particular configuration of the time to live values.

### E. Comparison With State-of-the-Art Techniques

We complement the previous qualitative observations with a comparative analysis of our method against a number of routing schemes proposed in literature that operate under the same assumption of *no a priori* mobility knowledge we took in this work. We selected ONE (Opportunistic Network Environment simulator) [4] to run our simulations as it offers embedded support to simulate realistic wireless technologies, as well as built-in modules running state-of-the-art routing schemes. Since we are interested in comparing the proposed approach against state-of-the-art alternatives, we focus here on results obtained with a specific setting of the reference scenario parameters; even if the simulation results might obviously change in the absolute value if changing the simulation environment (i.e., considering different mobility models), the proposed simulation

campaign is indeed insightful to showcase the relative performances trends of the considered delay tolerant routing techniques. In detail, we consider three different classes of nodes. For simplicity we assume that each class has the same number of nodes and the same mobility profile with average speed of  $6m/s$ . Transmission technologies are ZigBee, Bluetooth 4.0, and WiFi Direct for each class respectively (see Table II for the associated parameters). The source node energy budget is set by taking as a reference the battery of a smartphone (approx.  $5.45Wh$ ) and by considering an application layer consuming no more than the 30% of the total energy with a daily maximum load of  $20Mb$  of data. Assuming that a single application packet has a size of  $5kb$  we obtain a budget of  $1.4715J$  for each packet delivery. Each packet has to be delivered within 15 minutes from its creation at the source and each mobile node (excluding source and destination) has a local timeout of 5 minutes. Nodes move randomly in a free environment of radius of  $700m$ . We consider different number of nodes  $N \in \{10, 15, 20, 25, 30\}$  (recall that each class has the same number of nodes so each experiment has  $3N + 2$  total nodes populating the environment). For each  $N$  we generate 100 different random joint mobility patterns in which we assess the delivery of a single packet at the destination node with different routing schemes. We consider:

- 2-Hop greedy construction (1): the method studied in this work where Alg. 2 with  $G_1$  is adopted to compute the multi-class transmission policy for the source node;
- *spray and wait*: a bounded-copy, multi-hop routing scheme where copies are forwarded at each encounter (see [31]); the number of copies present in the network is bounded by a parameter  $k$  that we set at the source as the maximum number of packet transmissions given the total energy budget;
- *first contact*: a single copy, multi-hop, routing scheme where nodes forward the packet to the first encountered node;
- *direct delivery*: a single copy, single-hop, routing scheme where delivery can be performed only directly from the source;
- *epidemic*: multi-copy, multi-hop routing scheme where the packet is forwarded at every encounter.

Results of our simulations are reported in Fig. 6c. The epidemic protocol achieves, as expected, the best performance at the cost of a very large number of transmission in the network. Our method was able to outperform others for increasing number of nodes confirming the core motivation of this work, i.e., that trying to optimize the energy budget with a threshold policy can be profitable both in terms of delivery ratio and of spent energy. In particular, our approach was able to differentiate transmissions to different classes of nodes in a more optimized way, trying to concentrate more transmissions towards the classes most likely to be in the position of deliver the packet in the current remaining time. Direct delivery and first contact can be considered as baselines for this evaluation. The apparently counterintuitive better performance of direct delivery over first contact is due to the presence of the local timeout in mobile nodes (recall that, the local timeout at the source node is, by definition, the packet's time to live).

### VIII. CONCLUSIONS

We studied two-hop routing for Delay Tolerant Networks with heterogeneous technologies considering the signaling cost of the routing process. We formulated an optimization problem to derive the optimal two-hop policies and we designed and experimentally efficient approximation algorithms with theoretical bounds. Finally, we thoroughly evaluated our algorithms in realistic network settings in terms of approximation ratio and compute time as the parameters change. We experimentally showed that for all the generated instances our algorithms have an approximation ratio larger than 99% and that they scale linearly as the values of the parameters increase and therefore they can be applied with realistically large instances.

### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for the fruitful comments which helped improve the manuscript.

### REFERENCES

- [1] E. Altman, A. P. Azad, T. Basar, and F. D. Pellegrini, "Combined optimal control of activation and transmission in delay-tolerant networks," *IEEE ACM Trans. Netw.*, vol. 21, no. 2, pp. 482–494, Apr. 2013.
- [2] E. Altman, T. Basar, and F. D. Pellegrini, "Optimal control in two-hop relay routing," *IEEE Trans. Automat. Control*, vol. 56, no. 3, pp. 670–675, Mar. 2011.
- [3] F. D. Pellegrini, E. Altman, and T. Basar, "Optimal monotone forwarding policies in delay tolerant mobile ad hoc networks with multiple classes of nodes," in *Proc. 8th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2010, pp. 497–504.
- [4] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for DTN protocol evaluation," in *Proc. SIMUTOOLS*, 2009, p. 55.
- [5] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. ACM SIGCOMM*, 2004, pp. 145–158.
- [6] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "SGBR: A routing protocol for delay tolerant networks using social grouping," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2472–2481, Dec. 2013.
- [7] C. Liu and J. Wu, "On multicopy opportunistic forwarding protocols in nondeterministic delay tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 6, pp. 1121–1128, Jun. 2012.
- [8] H. Nishiyama, A. Takahashi, N. Kato, K. Nakahira, and T. Sugiyama, "Dynamic replication and forwarding control based on node surroundings in cooperative delay-tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 1–1, Oct. 2014.
- [9] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.
- [10] E. Bulut and B. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2254–2265, Dec. 2012.
- [11] K. Wei, D. Zeng, S. Guo, and K. Xu, "On social delay-tolerant networking: Aggregation, tie detection, and routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1563–1573, Jun. 2014.
- [12] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless routing in pocket switched networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 252–261, Jan. 2015.
- [13] E. Altman, G. Neglia, F. D. Pellegrini, and D. Miorandi, "Decentralized stochastic control of delay tolerant networks," in *Proc. INFOCOM*, 2009, pp. 1134–1142.
- [14] A. Guerrieri, I. Carreras, F. D. Pellegrini, D. Miorandi, and A. Montresor, "Distributed estimation of global parameters in delay-tolerant networks," *Comput. Commun.*, vol. 33, no. 13, pp. 1472–1482, 2010.
- [15] A. Bassamboo and R. S. Randhawa, "On the accuracy of fluid models for capacity sizing in queueing systems with impatient customers," *Oper. Res.*, vol. 58, no. 5, pp. 1398–1413, 2010.
- [16] W. Chahin, R. E. Azouzi, F. D. Pellegrini, and A. P. Azad, "Blind online optimal forwarding in heterogeneous delay tolerant networks," in *Proc. Wireless Days*, 2011, pp. 1–6.
- [17] E. Altman, L. Sattatelli, and F. D. Pellegrini, "Dynamic control of coding for progressive packet arrivals in DTNs," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 725–735, Feb. 2013.
- [18] E. Altman and F. D. Pellegrini, "Forward correction and fountain codes in delay-tolerant networks," *IEEE ACM Trans. Netw.*, vol. 19, no. 1, pp. 1–13, Feb. 2011.
- [19] A. Elwhishi, P.-H. Ho, K. Naik, and B. Shihada, "A novel message scheduling framework for delay tolerant networks routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 5, pp. 871–880, May 2013.
- [20] G. Resta and P. Santi, "A framework for routing performance analysis in delay tolerant networks with application to noncooperative networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1, pp. 2–10, Jan. 2012.
- [21] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with Wi-Fi direct: Overview and experimentation," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 96–104, Jun. 2013.
- [22] J. Liu, C. Chen, and Y. Ma, "Modeling neighbor discovery in Bluetooth low energy networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1439–1441, Sep. 2012.
- [23] N. Banerjee, M. D. Corner, and B. N. Levine, "Design and field experimentation of an energy-efficient architecture for DTN throwboxes," *IEEE ACM Trans. Netw.*, vol. 18, no. 2, pp. 554–567, Apr. 2010.
- [24] G. Resta and X. Zhang, "Optimal delay-power tradeoff in sparse delay tolerant networks: A preliminary study," in *Proc. ACM SIGCOMM CHANTS*, 2006, pp. 237–244.
- [25] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proc. ACM SIGCOMM Workshop Delay Tolerant Netw. (WDIN)*, 2005, pp. 260–267.
- [26] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay Tolerant Netw. (WDTN)*, 2005, pp. 252–259.
- [27] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, no. 10, pp. 2867–2891, Jul. 2007.
- [28] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Perform. Eval.*, vol. 62, nos. 1–4, pp. 210–228, Oct. 2005.
- [29] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*. Cambridge, U.K.: Cambridge Univ. Press, 2012, pp. 71–104.
- [30] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*. New York, NY, USA: Springer, 1983, pp. 235–257.
- [31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay Tolerant Netw. (WDIN)*, 2005, pp. 252–259.





**Nicola Basilico** received the M.S. degree in computer science and the Ph.D. degree in information technology from Politecnico di Milano, Milano, Italy, in 2007 and 2011, respectively. Since 2014, he has been an Assistant Professor of computer science with the University of Milan, Milan, Italy. His research interests include artificial intelligence with a focus on multiagent systems, algorithmic game theory, and autonomous robotics.



**Nicola Gatti** received the M.S. degree in biomedical engineering and the Ph.D. degree in information engineering from Politecnico di Milano, Milano, Italy, in 2001 and 2005, respectively. Since 2015, he has been an Associate Professor of Computer Science with Politecnico di Milano. His research interests include algorithmic game theory, computational microeconomics, artificial intelligence, and operations research. In 2011, he was the recipient of the AIXIA Award as Best Young Italian Researcher in artificial intelligence.



**Matteo Cesana** (S'01–M'04) received the M.S. degree in telecommunications engineering and the Ph.D. degree in information engineering from Politecnico di Milano, Milano, Italy, in 2000 and 2004, respectively. Currently, he is an Associate Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy. From September 2002 to March 2003, he was a Visiting Researcher with the Computer Science Department, University of California, Los Angeles (UCLA), Los Angeles, CA, USA. His

research interests include design, optimization, and performance evaluation of wireless networks with a specific focus on wireless sensor networks and cognitive radio networks.