Forecasting DTN Performance under Heterogeneous Mobility: The Case of Limited Replication

Andreea Picu
Communication Systems Group
ETH Zürich, Switzerland
Email: lastname@tik.ee.ethz.ch

Thrasyvoulos Spyropoulos
Mobile Communications
EURECOM, France
Email: firstname.lastname@eurecom.fr

***Abstract*—*Opportunistic* or *Delay Tolerant Networks* (DTNs) may be used to enable communication in case of failure or lack of infrastructure (disaster, censorship, remote areas) and to complement existing wireless technologies (cellular, WiFi). Wireless peers communicate when in contact, forming an impromptu network, whose connectivity graph is highly dynamic and only partly connected. In this harsh environment, communication algorithms are mostly greedy, choosing the best solution among the locally available ones. Furthermore, they are routinely evaluated through simulations only, as they are hard to model analytically. Even when more insight is sought from models, they usually assume *homogeneous* node meeting rates, thereby ignoring the attested heterogeneity and non-trivial structure of (human) mobility.**

We propose DTN-Meteo: a new unified analytical model that maps an important class of DTN optimization problems and the respective (greedy) algorithms into a Markov chain traversal over the relevant solution space. Fully heterogeneous node contact patterns and a range of algorithmic actions jointly (but separably) define transition probabilities. Thus, we provide closed-form solutions for crucial performance metrics under generic settings. While DTN-Meteo has wider applicability, in this paper, we focus on algorithms with explicitly controlled replication. We apply our model to two problems: routing and content placement. We predict the performance of state of the art algorithms (SimBet, BubbleRap) in various real and synthetic mobility scenarios and show that surprising precision can be achieved against simulations, despite the complexity of the problems and diversity of settings. To our best knowledge, this is the first analytical work that can accurately predict performance for utility-based algorithms and heterogeneous node contact rates.

I. Introduction

Opportunistic or *Delay Tolerant Networks* (DTNs) are envisioned to support communication in case of failure or lack of infrastructure (disaster, censorship, rural areas) and to enhance existing wireless networks (e.g. offload cellular data traffic), enabling novel applications. Nodes harness unused bandwidth by exchanging data when they are in proximity (*in contact*), aiming to forward data probabilistically closer to destinations. Using redundancy (e.g. coding, replication) and smart mobility prediction schemes, data can be transported over a sequence of such contacts, despite the lack of end-to-end paths.

Many challenging problems arise in this context: routing [1], [2], resource allocation [3], content placement [4], etc. Given the disconnected and highly dynamic nature of the connectivity graph of Opportunistic Networks, these problems are substantially harder here, than in traditional connected networks. As a result, most solutions proposed for each problem are greedy, local heuristics. Moreover, the performance evaluation of these solutions is largely simulation-based, as it is hard to develop suitable analytical models. While simulations provide quantitative results for realistic settings, they offer little insight into the problems and it is hard to generalize their findings, due to the sheer range of mobility scenarios, optimization problems (e.g. routing) and the multitude of algorithms for them.

This work is funded by ETH Zürich under Research Grant ETH-20 09-1 and by the EU Commission under the SCAMPI (FP7–258414) FIRE Project.

Early analytical models for DTNs were devised [5]–[7], to complement simulations. These are typically based on Markov chains or fluid approximations. For example, in routing, the spread of a message was modeled with a Markov chain, whose states are the number of copies in the network [5]. Alternatively, the spreading process was treated as a fluid flow and the number of copies, approximated as a continuous-valued function of time and the node meeting rate [8].

However, for the sake of tractability, these models mainly rely on simple mobility assumptions (e.g. Random Walk, Random Waypoint), where node mobility is stochastic and *independent identically distributed* (IID). Studies of real scenarios [9], [10] reveal more complex structure, comprising considerable heterogeneity in node mobility, questioning the usefulness of these models' predictions. Protocol design has integrated these findings in new, sophisticated "utility-based" solutions, aiming at exploiting this heterogeneity [4], [11]. However, the complexity of mobility patterns involved, and often of the algorithms themselves, implies that the evaluation of such newer protocols remains purely simulation-based.

Recent techniques that introduce *some* mobility heterogeneity in existing models [12]–[14] quickly become prohibitively complex or deal with simple protocols only. It is thus evident that *a common analytical framework* is needed that can successfully deal with (a) *more realistic mobility assumptions*, and (b) *the range of DTN communication and optimization problems and the abundance of protocols for each*, while still providing insight and, ideally, closed form solutions.

Our *first* step in this direction is to observe that the bulk of proposed algorithms, whether for routing, content dissemination, distributed caching etc, essentially solve a combinatorial optimization problem over the state of every node in the network. Each algorithm defines a preference (*utility*) function over possible network states and moves to better states. The *second* observation is that candidate new states, in the DTN context, are presented according to the *stochastic* mobility of the nodes involved. As such, *the traversal of the solution space of a problem is also stochastic*. The *third* shared element is that, due to the difficulty in steering nodes globally, protocols resort to local deterministic (*greedy*) or randomized heuristics to choose between the current and a new possible state, *involving state changes only in the two nodes in contact*.

Using the above insight, DTN-Meteo maps an optimization problem into a Markov chain (MC), where each state (e.g. assignment of content replicas to nodes) is a potential solution, and transition probabilities are driven by two key factors: (i) *node mobility*, which "offers" new solutions to the algorithm based on the (heterogeneous, but time-invariant) contact probability of each node pair, and (ii) *the algorithm*, which orders the states using a utility function and accepts better ones deterministically (*greedy*) or randomly. This not only decouples the algorithm's effect from mobility, but also allows one to derive interesting performance metrics (convergence delay, delivery probability) using transient analysis of this MC.

While DTN-Meteo is more widely applicable, due to space constraints, in this paper we mostly focus on algorithms, which explicitly limit the copies of a message at the source [4], [15]. In the other category of DTN algorithms, replication is either unlimited [1] or implicitly limited through utility [16]. Here, we also scrutinize the most important part of such an algorithm, its utility function, by simply forcing it into the first category. In [17], we consider unlimited replication (or flooding-based) schemes. Summarizing, our contributions are:

- We formulate an important class of DTN optimization problems using *a Markovian model*, that combines the heterogeneous mobility properties of a scenario and the actions of an algorithm into an appropriate transition matrix over a problem's solution space (Section II). This model enables us to calculate in closed-form the performance (delay, success probability) of a fixed-replication algorithm in various mobility scenarios (Section III).
- To demonstrate the value and relative generality of DTN-Meteo, we apply it to both single- and multi-copy algorithms for two DTN problems: (i) Unicast routing (SimBet [11] and BubbleRap [16]) and (ii) Content placement/Distributed caching [4]. We chose state of the art, utility-based algorithms which cannot be modeled by existing tools. We compare the accuracy of our predictions against simulations for a range of synthetic and real-world mobility traces (Section IV).
- We provide a simple approximation scheme, which improves DTN-Meteo's scalability, without significantly impacting the accuracy of its predictions (Section V).

A key part of DTN models is the *approximation* of pairwise contact patterns with exponential variables. While this idea has spurred some controversy [18], Karagiannis et al. established in [19] that after a short "characteristic time" of about half a day, the inter-contact time does exhibit exponential decay. Thus, *approximating* inter-contact times with exponential variables is not unreasonable, especially when focusing on residual inter-contact times[1], as is the case with analyses of forwarding schemes. The exponential assumption enables an otherwise unfeasible analysis and spares us from other more restrictive assumptions (e.g. path length of at most two hops [20]).

[1]This is the time until the next contact for a node pair, from an arbitrary point in time. Intuitively, the residual time reflects how much time a device must wait, before being able to forward a message to another given device.

II. A Generic Model for DTN Problems

Let $\mathcal{N}$ be our Opportunistic Network, with $|\mathcal{N}| = N$ nodes. $\mathcal{N}$ is a relatively sparse ad hoc network, where node density is insufficient for establishing and maintaining (end-to-end) multi-hop paths. Instead, data is stored and carried by nodes, and forwarded through intermittent contacts established by node mobility. A *contact* occurs between two nodes who are in range to setup a bi-directional wireless link to each other.

We assume an optimization problem over the N-node network $\mathcal{N}$ (e.g. multicast under resource constraints) and a distributed algorithm for this problem, run by all nodes. Our long term aim is to understand the performance of the algorithm as a function of the nodes' behavior and attributes (mobility, collaboration, resource availability etc). In this section, we first define more precisely the class of problems we consider, as well as the type of algorithms used to solve them. Then, we present our network model and assumptions. Finally, we show how to integrate all of the above into a Markov chain that will allow us to derive useful performance metrics.

A. Solution Space

We consider a class of DTN problems for which a solution can be expressed in terms of nodes' states. In this paper, we restrict ourselves to *binary node states*, to better illustrate the key concepts; however, in a more realistic variant of DTN-Meteo, nodes' states should be chosen from a set of *B-bit integers* (e.g. to allow the modeling of B messages). Then, the space of candidate solutions for such problems is a set of N-element vectors, possibly restricted by a number of constraints. Finally, an algorithm for the problem defines a ranking over these solutions, captured by a utility function $U(\cdot)$. The goal is to maximize this utility (or minimize a cost function). We define our class of problems as follows:

- **node state space** $\quad \mathcal{S} \subset \mathbb{N}$ or $\mathcal{S} = \{0,1\}$ (1)
- **network state space** $\quad \Omega \subseteq \mathcal{S}^N$ (2)

$$\Omega = \{\mathbf{x} \mid \mathbf{x} = (x_1, x_2, \dots, x_N)\}, \quad x_i \in \mathcal{S}, \;\; \forall i \in \mathcal{N} \tag{3}$$

- a set of **constraints** $\quad f_i(x_1, \dots, x_N) \leqslant \rho_i$ (4)
- a **utility function** $\quad U_{\mathbf{x}} = U(x_1, \dots, x_N)$. (5)

This is, in fact, the combinatorial optimization class, which naturally encompasses several DTN problems, as they are dealing with indivisible entities (nodes, messages, channels etc) and have rules that define a finite number of allowable choices (choice of relays, assignment of channels etc). Below are some examples of DTN problems that can be thus modeled.

Content placement. The goal in content placement, is to make popular content (news, software update etc) easily reachable by interested nodes. As flooding the content is unscalable, a small number L of replicas can be pushed from its source to L strategic relays, which will store it for as long as it is relevant, and from whom encountered interested nodes retrieve it[2]. In its simplest form, the source of the content distributes the L replicas to L initial nodes (e.g. randomly, using binary or source spraying [2]). These initial relays then forward their copies only to nodes that improve the desired utility –

[2]L must be derived to achieve a trade-off between desired performance and incurred costs, for example as in [2].

	Replication		Constraint
i)	single-copy	[11], [21], [22]	$\sum_{i=1}^{N} x_i = 1$
ii)	fixed budget L	[2], [15]	$\sum_{i=1}^{N} x_i \leqslant L$
iii)	epidemic	[1]	$\sum_{i=1}^{N} x_i \leqslant N$
iv)	utility-based[4]	[3], [16], [23]	$\sum_{i=1}^{N} x_i \leqslant N$

TABLE I
MODELING REPLICATION WITH EQ. (4)

which can be based on mobility properties, willingness to help, resources, etc. (see [4] for some examples).

Here, the binary state of a node i is interpreted as the node being ($x_i = 1$) or not being ($x_i = 0$) a provider for the content of interest[3]. There is a single constraint for any allowed solution, namely $\sum_{i=1}^{N} x_i \leqslant L$, that is, in Eq. (4) $\rho_i = \rho = L$.

Routing. In routing, be it unicast, multicast, broadcast, or anycast, the binary state of node i is interpreted as carrying or not carrying a message copy. For example, for unicast routing from source node s to destination node d, the initial network state is $\mathbf{x}_0 = (0, 0, \dots, \{x_s = 1\}, \dots, \{x_d = 0\}, \dots, 0)$. The desired network state is any $\mathbf{x}^\star$, with $x_d = 1$ and $x_i \in \{0,1\}$, $\forall i \neq d$. This can easily be extended to group communication, with multiple sources and/or multiple destinations.

Various replication strategies can be expressed using Eq. (4) constraints, as shown in Table I. Different schemes in each category, essentially differ in the utility function used to rank states and the action taken given utility difference. In this paper, we analyze the first two strategies in Table I.

The multi-copy cases of both content placement and fixed budget routing include an initial replication phase, i.e., going from network state $\mathbf{x}_0$ with $\sum_{i=1}^{N} x_i = 1$ to $\mathbf{x}_n$ with $\sum_{i=1}^{N} x_i = L$. Because a very small number of copies L (compared to N) is usually enough for a good cost–gain tradeoff, the initial replication phase has a negligible contribution in the total delay (see e.g. [2]). While we could easily include this phase in our model, the added realism does not justify the increase in complexity. We therefore ignore this delay, to simplify our discussion and deal with a smaller state space.

B. Exploring the Solution Space

In traditional optimization problems, local search methods define a neighborhood around the current solution, evaluate all solutions in the neighborhood and "move" towards the best one therein (greedy algorithms). Occasionally, they may also move to lower utility states (using randomization) in order to overcome local maxima, as in simulated annealing. This aspect is fundamentally different in DTN optimization. The next candidate solution(s) cannot usually be chosen. Instead, the solution space Ω is explored via node mobility (contacts): a new solution is offered and can replace the current one only when two nodes come in contact. This has two major implications: (i) a new solution can differ (from the current) in the state of at most two nodes, the ones involved in the contact; (ii) new solutions are presented randomly; hence, the traversal of a DTN problem's solution space is by nature *stochastic*.

[3]A B-bit integer node state extends this to providing B pieces of content.

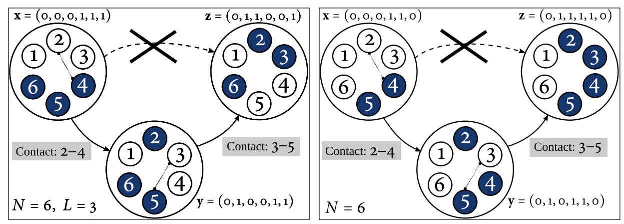
[4]Replicating only to higher utility nodes limits copies implicitly, but in the worst case, there will still be N copies.

Consider implication (i) first (we treat (ii) in Section II-C). Every contact between a relay node/content provider i ($x_i = 1$) and another node j ($x_j = 0$) offers the chance of moving to a new network state $\mathbf{y} \in \Omega$, with $y_i = 0$ and $y_j = 1$ (forwarding) or $y_i = 1$ and $y_j = 1$ (replication). ***If*** the replica is transferred from relay i to j, then the $\mathbf{xy}$ transition happens. Fig. 1 provides examples of *potential* state transitions, along with contacts required for the transitions to be possible.

Transition $\mathbf{xz}$ in Fig. 1 require two contacts to *start* simultaneously, and is thus not allowed[5]. This means that only transitions between *adjacent* states are possible, where we define state adjacency as:

$$\delta(\mathbf{x}, \mathbf{y}) = \sum_{1 \leqslant i \leqslant N} \mathbb{1}\{x_i \neq y_i\} \leqslant 2, \tag{6}$$

i.e., the two network states may only differ in one or two nodes. An encounter of those two nodes is a necessary (but not sufficient) condition for transition $\mathbf{xy}$ to happen.



(a) Forwarding (b) Replication

Fig. 1. Example state transitions in two 6-node DTNs.

From Fig. 1 it is clear that *the sequence of solutions presented to a distributed optimization algorithm in this context is dictated by node mobility and is thus a random sequence of contact events*. We next examine this contact process.

C. Modeling Heterogeneous Node Mobility

As pointed out in the beginning of this section, network $\mathcal{N}$ is a sparse ad hoc network of mobile nodes, in which data is exchanged only upon contacts. Thus, a mobility model based on contact patterns is sufficient for our analysis. We describe the network $\mathcal{N}$ using a marked point process $(M_n)_{n \in \mathbb{Z}} = \{T_n, \sigma_n\}$, where T_n denotes the starting time of a contact and $\sigma_n = (i, j, \Delta_n)$ denotes the two nodes in contact $i, j \in \mathcal{N}$ and the duration of the contact Δ_n [24], [25]. The random variables $J_n = T_n - T_{n-1}$ are the times between the initiations of two successive contacts. We assume the following:

1) $(T_n)_{n \in \mathbb{Z}}$ are epochs of a stationary and ergodic renewal process – the times J_n are IID with intensity[6] λ, dependent on network sparsity.
2) $T_n < T_{n+1}$, $\forall n \in \mathbb{Z}$ – no two contacts start at the same time, i.e., $J_n > 0$.
3) $\Delta_n << J_n$ – the duration of a contact is negligible compared to the time between two contacts, but sufficient for all data transfers to take place. Therefore, $\sigma_n = (i, j)$.
4) $(\sigma_n)_{n \in \mathbb{Z}}$ is a discrete IID process, with common distribution $\mathbb{P}[\sigma_n = (i, j)] = p_{ij}^c$. This is the pairwise

[5]We emphasize that this is not a strict necessity for our networks of interest, but rather an assumption of our mobility model, for analytical convenience (see Section II-C for more details). However, given the relative sparsity of the networks in questions, such events occur with low probability.

[6]Expected number of points or contact "arrivals" per time unit.

contact probability, that the next contact in the sequence is between nodes i and j.

We distinguish between *standard time* (wall-clock time) and *event time* (measured in number of contact events or "contact ticks"). Because our contact process is stationary and ergodic, it is easy to relate event time to wall time, using Wald's equation [26]. For simplicity, we use event time throughout the paper. Consequently, within the IID $(\sigma_n)_{n\in\mathbb{Z}}$ process, at any moment, the remaining *discrete* inter-contact delay between nodes i and j has a *geometric distribution* with parameter p^c_{ij}.

Under the above assumptions, a given mobility scenario with *heterogeneous node mobility* can be described through its pairwise contact probabilities p^c_{ij} forming the ***contact probability matrix***:

$$\mathbf{P}^c = \{p^c_{ij}\}. \tag{7}$$

Probabilities p^c_{ij} can be either measured directly from a given real or synthetic mobility trace (e.g. maximum likelihood estimation of p^c_{ij} for every pair); or they can be calculated using the pair's contact statistics (e.g. frequency), as in [27]. We apply the former approach to all traces we use (Table II).

Going back to our solution space exploration, our contact process model implies the following: When at a state $\mathbf{x}$, the next contact "tick" presents a new solution $\mathbf{y}$ to the algorithm with probability p^c_{ij}. The new solution $\mathbf{y}$ differs from $\mathbf{x}$ in positions i and j only. For example, in Fig. 1(a): when at state $\mathbf{x} = (0,0,0,1,1,1)$ the algorithm *could* move to a new solution $\mathbf{y} = (0,1,0,0,1,1)$ in the next contact, with probability p^c_{24}.

D. Modeling a Local Optimization Algorithm

Node contacts merely *propose* new candidate solutions. Whether the relay i does in fact hand over its message or content replica to j (or, more generally, whether a new allocation of objects between i and j is chosen), is decided by the *algorithm*. In greedy (utility-ascent) schemes, a *possible* state transition occurs only if it improves the utility function U, specific to each problem. Then, for our DTN problems, a possible transition $\mathbf{xy}$ occurs with ***acceptance probability***[7]:

$$A_{\mathbf{xy}} = \mathbb{1}\{U_{\mathbf{x}} < U_{\mathbf{y}}\}. \tag{8}$$

More generally, the acceptance probability may be any function of the two utilities: $A_{\mathbf{xy}} \in [0,1]$. This allows us to model stochastic utility-ascent algorithms (e.g. simulated annealing), as well, in DTN-Meteo. Due to space limitations, in the remainder of this paper we will focus on greedy (deterministic) algorithms; we stress though that a similar, equivalent analysis is applicable to stochastic (local) optimization schemes.

E. A Markov Chain Model for Distributed Optimization

Summarizing, the transition probability between *adjacent* network states $\mathbf{x}$ and $\mathbf{y}$ can be expressed in function of the contact probability and the acceptance probability as:

$$p_{\mathbf{xy}} = p^c_{ij} \cdot A_{\mathbf{xy}}, \tag{9}$$

where nodes i and j are the two nodes whose encounter provokes the state transition.

[7]In a distributed algorithm, the utilities of two states must be comparable locally by the two nodes in contact (e.g., additive function). This is not a requirement for our model, but a challenge for the algorithm designer [4].

Given our mobility model (stationary contact process), the above transition probability is *time-homogeneous*. Most DTN protocols suppose the existence of a mobility-related node utility (e.g. contact frequency). DTN-Meteo assumes this utility is readily available. In practice, protocols do not know the utility a priori; thus, they must solve the added problem of estimating it – usually through an integrated sampling component [3], [23]. However, if the mobility is stationary (and the estimation designed correctly), the online and offline utility rankings will coincide. We discuss non-stationarity in Section VI.

The final observation to put our pieces of DTN-Meteo together is that, for most utility-based DTN algorithms, the transition from any state $\mathbf{x}$ to any other state $\mathbf{y}$ only depends on these two states and not on past states. This means that our system has the Markov property, therefore we model it with a **time-homogeneous discrete-time Markov chain** $(\mathbf{X}_n)_{n\in\mathbb{N}_0}$ over the solution space Ω. From above, the transition probability matrix of the Markov chain is $\mathbf{P} = \{p_{\mathbf{xy}}\}$, with:

$$p_{\mathbf{xy}} = \mathbb{P}[\mathbf{X}_{n+1} = \mathbf{y} \,|\, \mathbf{X}_n = \mathbf{x}] = \begin{cases} 0, & \delta(\mathbf{x},\mathbf{y}) > 2 \\ p^c_{ij} \cdot A_{\mathbf{xy}}, & 0 < \delta(\mathbf{x},\mathbf{y}) \leqslant 2 \\ 1 - \sum\limits_{\mathbf{z}\neq\mathbf{x}} p_{\mathbf{xz}}, & \mathbf{x} = \mathbf{y}. \end{cases} \tag{10}$$

where, i and j are the two nodes whose encounter provokes the state transition. p^c_{ij} is the ***mobility component*** of the transition probability and $A_{\mathbf{xy}}$ is the ***algorithm component***.

III. Convergence Analysis

In the previous section, we have transformed any (difficult) problem of our defined class into a Markov chain. The chain $(\mathbf{X}_n)_{n\in\mathbb{N}_0}$ defined in Eq. (10), modeling greedy utility-based algorithms, is *absorbing*. Thus, we use the theory of absorbing Markov chains [28], to obtain absorption probabilities and delays for all pairs: (transient state → absorbing state). From these we derive crucial performance metrics for our algorithms, such as *delivery ratio* and *delivery delay* in routing or the *probability and delay of optimally placing content*.

A. Absorption Analysis of Discrete-Time Markov Chains

Maximum utility (optimum) network states are, by definition, absorbing states in the Markov chain. In addition to maximum utility states, there may also be network states of smaller utility, but from which it is impossible to greedily advance to higher utility, due to the mobility pattern. These correspond to local maxima and are also absorbing states in $\mathbf{P}$. We use the following notation for the two sets of states:

$$\begin{aligned} \mathcal{GM} &\subset \Omega \quad \text{(global maxima)}, \\ \mathcal{LM} &\subset \Omega \quad \text{(local maxima)}. \end{aligned}$$

$\mathcal{GM}$ contains all solutions $\mathbf{x}^\star \in \Omega$, such that $U_{\mathbf{x}^\star} \geqslant U_{\mathbf{y}}$, for all states $\mathbf{y} \in \Omega$. For example, in Spray and Wait routing, states $\mathbf{x}^\star$ are the $\binom{N-1}{L-1}$ states in which one of the L copies is at the destination node d. $\mathcal{LM}$ contains all solutions $\mathbf{x} \in \Omega \setminus \mathcal{GM}$, such that for all states $\mathbf{y} \in \Omega$, the transition probability $p_{\mathbf{xy}}$ is zero, either because:

- $\delta(\mathbf{x},\mathbf{y}) > 2$ – the states are not adjacent, or because
- $p^c_{ij} = 0$ – the required nodes never meet, or because

- $U_{\mathbf{y}} < U_{\mathbf{x}}$ – all "neighboring" states $\mathbf{y}$ have lower utilities.

Every other solution in $\Omega \setminus \mathcal{GM}$ is a transient state. Denote by $\mathrm{TR} \subset \Omega$, the set of transient states. Then, $\Omega = \mathcal{GM} \cup \mathcal{LM} \cup \mathrm{TR}$.

In order to derive absorption related quantities, we write the matrix $\mathbf{P}$ in *canonical form*, where states are re-arranged such that transient states (TR) come first, followed by absorbing states corresponding to local maxima ($\mathcal{LM}$), followed by maximum utility states $\mathcal{GM}$:

$$\mathbf{P} = \begin{array}{c} \begin{array}{ccc} \mathrm{TR} & \mathcal{LM} & \mathcal{GM} \end{array} \\ \left(\begin{array}{c:c:c} \mathbf{Q} & \mathbf{R_1} & \mathbf{R_2} \\ \hdashline \mathbf{0} & \mathbf{I_1} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{I_2} \end{array}\right) \end{array} \begin{array}{l} \\ \mathrm{TR} \\ \mathcal{LM} \\ \mathcal{GM} \end{array} \tag{11}$$

Let $|\mathcal{GM}| = r_2$, $|\mathcal{LM}| = r_1$ and $|\mathrm{TR}| = t$. That is, there are r_2 optimum states, r_1 local maxima and t transient states. Then, $\mathbf{I_1}$ and $\mathbf{I_2}$ are, respectively, the $r_1 \times r_1$ and the $r_2 \times r_2$ identity matrices, $\mathbf{Q}$ is a $t \times t$ matrix, and $\mathbf{R_1}$ and $\mathbf{R_2}$ are, respectively, non-zero $t \times r_1$ and $t \times r_2$ matrices.

We can now define the *fundamental matrix* $\mathbf{N}$ for the absorbing Markov chain as follows:

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \cdots \tag{12}$$

The last equality is easy to derive (see [28], page 45). $\mathbf{N}$ is a $t \times t$ matrix whose entry $n_{\mathbf{xy}}$ is the expected number of times the chain is in state $\mathbf{y}$, starting from state $\mathbf{x}$, before getting absorbed. Thus, the sum of a line of the fundamental matrix of an absorbing Markov chain is the expected number of steps until absorption, when starting from the respective state.

Finally, for the derivation of absorption quantities, we also need the initial probability distribution of the Markov chain $(\mathbf{X}_n)_{n \in \mathbb{N}_0}$. Denote by $p_{\mathbf{x}}^{(0)}$ the probability that the chain starts in state $\mathbf{x} \in \Omega$. Assuming source nodes are equally likely,

$$p_{\mathbf{x}}^{(0)} = \frac{1}{N} \sum_{i \mid x_i = 1} \mathbb{P}[\mathbf{X}_0 = \mathbf{x} \mid \text{source is } i]. \tag{13}$$

The conditional probability above may be hard to calculate depending on the initial replication strategy. For the sake of tractability, we assume that simple source spraying [2] is used and that the spreading completes before the forwarding algorithm starts. We defer the treatment of more sophisticated initial spreading conditions to future work. Then:

$$p_{\mathbf{x}}^{(0)} = \frac{1}{N} \sum_{i \mid x_i = 1} \frac{\prod_{j=1}^{N} x_j p_{ij}^c}{\sum_{\mathbf{y} \mid y_i = 1} \left(\prod_{j=1}^{N} y_j p_{ij}^c \right)}. \tag{14}$$

B. From Absorption Analysis to Practical Metrics

Based on the fundamental matrix and the initial probability distribution, we can now derive the metrics of interest for any algorithm of our class. In the following theorems, we show how to do this for our example problems: routing and content placement. However, the theorems apply to any other problem unchanged, as long as the state space and utility are defined.

Theorem 1 (Success Probability): The end-to-end *delivery probability* for a routing algorithm modeled by Markov chain $\mathbf{P}$ starting from any initial source(s) with equal probability, is

$$p_d = \sum_{\mathbf{x}^\star \in \mathcal{GM}} p_d(\mathbf{x}^\star) = \sum_{\mathbf{x}^\star \in \mathcal{GM}} \left(\sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{x}\mathbf{x}^\star} \right), \tag{15}$$

where $b_{\mathbf{x}\mathbf{x}^\star}$ is the probability of being absorbed at state $\mathbf{x}^\star$, given we start at $\mathbf{x}$ and $p_{\mathbf{x}}^{(0)}$ is the probability of starting at $\mathbf{x}$. In matrix form, $\mathbf{B}^\star = \{b_{\mathbf{x}\mathbf{x}^\star}\}$ is a $t \times r_2$ matrix, with $\mathbf{B}^\star = \mathbf{N}\mathbf{R_2}$.

The *success probability* of content placement finding the best set of L relays, starting from any initial source(s) with equal probability obeys the same relation.

Proof: The delivery probability p_d is obtained from the absorption probabilities $b_{\mathbf{x}\mathbf{x}^\star}$ of Markov chain $(\mathbf{X}_n)_{n \in \mathbb{N}_0}$. The absorption probabilities can be derived using first-step analysis as in [28], page 52. We refer the interested reader to [26], for a comprehensive proof of this theorem. ■

In addition to knowing what chances a greedy algorithm has of finding an optimal solution, we are also interested in how long it will take. In the following theorem, we derive the expected end-to-end delivery delay of routing and the convergence delay of content placement using the fundamental matrix $\mathbf{N}$ and the individual delivery ratios/success probabilities $p_d(\mathbf{x}^\star) = \sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{x}\mathbf{x}^\star}$ defined in Eq. (15) above.

Theorem 2 (Expected Delay): The expected end-to-end *delivery delay* for a routing algorithm modeled by Markov chain $\mathbf{P}$, starting from any source with equal probability, *given that it does not get absorbed in any local maximum* is:

$$\mathbb{E}[T_d] = \sum_{\mathbf{x}^\star \in \mathcal{GM}} \frac{p_d(\mathbf{x}^\star)}{p_d} \mathbb{E}[T_d(\mathbf{x}^\star)] = \sum_{\mathbf{x}^\star \in \mathcal{GM}} \frac{p_d(\mathbf{x}^\star)}{p_d} \left(\sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} \tau_{\mathbf{x}\mathbf{x}^\star} \right), \tag{16}$$

where $\tau_{\mathbf{x}\mathbf{x}^\star}$ is the delay of being absorbed at state $\mathbf{x}^\star$, given we start at $\mathbf{x}$, and $p_{\mathbf{x}}^{(0)}$ is the probability of starting at $\mathbf{x}$. In matrix form, $\boldsymbol{\tau}_{\mathbf{x}^\star} = \{\tau_{\mathbf{x}\mathbf{x}^\star}\}$ is a t-element column vector with $\boldsymbol{\tau}_{\mathbf{x}^\star} = \mathbf{D}^{-1}\mathbf{N}\mathbf{D}\mathbf{c}$. $\mathbf{c}$ is a t-element column vector with ones, and $\mathbf{D}$ is a $t \times t$ diagonal matrix with entries $b_{\mathbf{x}\mathbf{x}^\star}$ for $\mathbf{x} \in TR$ and $\mathbf{x}^\star \in \mathcal{GM}$. The expected *convergence delay* for content placement to find the best set of L relays, starting from any initial source with equal probability obeys the same relation.

Proof: Assume we start in a transient state $\mathbf{x}$ of our chain $\mathbf{X}_n$ and compute all *conditional transition probabilities*, given that the process ends up in optimal state $\mathbf{x}^\star$. Then, we obtain a new absorbing chain $\mathbf{Y}_n$ with a single absorbing state $\mathbf{x}^\star$. The transient states are unchanged, except we have new transition probabilities, which we can calculate as in [28], page 64. Then, $\boldsymbol{\tau}_{\mathbf{x}^\star}$ is obtained from the fundamental matrix of the new chain, as the matrix' row sums. This process must be repeated for all $\mathbf{x}^\star \in \mathcal{GM}$. Then, using the initial probabilities $p_{\mathbf{x}}^{(0)}$ and the law of total expectation, Eq. (16) is obtained. We refer the reader to [26], for a comprehensive proof of this theorem. ■

Summarizing, we have just shown how to calculate convergence probabilities and delays from any transient state to any absorbing state. We have also shown how these can be mapped to metrics of great practical interest for two DTN problems: routing and content placement. In the next section, we compare our predictions obtained from Thms. 1 and 2 to results obtained from simulations, for both problems.

IV. Applications to Communication Algorithms

In this section, we apply our analysis to the state-of-the-art routing algorithms – SimBet [11] and BubbleRap [16], and to greedy content placement, all of them using the first two replication strategies in Table I: single-copy and fixed budget L. These are all utility-based algorithms which cannot be modeled by existing tools.

A. Utilities for Routing and Content Placement

First, we briefly describe the utility used by each algorithm. These utilities were chosen by the designer of the proposed algorithms for the respective problem, and are not necessarily optimal (different utilities would define different algorithms).

In many recent protocols (including our case studies), a node's utility is assessed using the strength of its mobility ties to other nodes, e.g., based on contact frequency and/or duration etc. These tie strengths are sometimes used as such. However, predominantly, they are aggregated in a single static ("social") graph, on which node utility can be mapped to metrics from social network analysis, such as centrality and community membership or similarity. This may be a weighted ($\mathbf{W}$) or a binary ($\mathbf{W}_{\text{bin}}$) graph. For our case studies, we use normalized pairwise meeting frequencies as weights w_{ij}. When necessary, we obtain $\mathbf{W}_{\text{bin}}$ from $\mathbf{W}$ by keeping only the highest weights up to the optimal link density [29].

1) ***Content Placement****:* Recall the goal of content placement: to make popular content (news, software update etc) easily reachable by interested nodes, by pushing L copies of it from its source to L "strategic" relays. The accessibility that a relay offers to the rest of the network is related to the expected meeting delay between the relay and any other node. This delay is minimized (and accessibility maximized) by relays who meet the most (unique) nodes per unit time [30]. Using the graph $\mathbf{W}$, this number amounts to the relay's (or a node's) degree: $d_i = \sum_{j=1}^{N} w_{ij}$, with $w_{ii} = 0$ by convention. Thus, we define the utility of a network state $\mathbf{x}$ as:

$$U_{\mathbf{x}} = \sum_{i=1}^{N} x_i d_i. \tag{17}$$

2) ***SimBet****:* is a DTN routing algorithm based on social network analysis. It assesses similarity (number of neighbors in common) to detect nodes that are part of the same community, and (ego) betweenness centrality to identify bridging nodes, that could carry a message from one community to another. We calculate these metrics on the binary graph $\mathbf{W}_{\text{bin}}$. Thus, in SimBet the utility of node i, for a destination node d is[8]:

$$U_i(d) = \alpha \cdot Sim_i(d) + \beta \cdot Bet_i \tag{18}$$

and the utility of a network state is, as above, the sum of individual relay utilities: $U_{\mathbf{x}}(d) = \sum_{i=1}^{N} x_i U_i(d)$.

SimBet was first published as a single-copy utility-based protocol. It was later enhanced [15] with the option of using a fixed number of copies L.

3) ***BubbleRap****:* uses an approach to routing similar to SimBet. Again, betweenness centrality is used to find bridging nodes until the content reaches the destination community. Communities are explicitly identified by a community detection algorithm, instead of implicitly by using similarity. Once in the right community, content is only forwarded to other nodes of that community: a local centrality metric is used to find increasingly better relays within the community. We use $\mathbf{W}_{\text{bin}}$ to obtain betweenness and apply the Louvain method [31] on the same graph to detect communities. Thus, in BubbleRap the utility of node i, for a destination node d is:

$$U_i(d) = \mathbb{1}\{i \in C_d\} \cdot LBet_i + \overline{\mathbb{1}\{i \in C_d\}} \cdot GBet_i, \tag{19}$$

where $\mathbb{1}\{i \in C_d\}$ is an indicator variable for node i belonging to the destination's community, and $LBet_i$ and $GBet_i$ are the local and global centralities, respectively. The utility of a network state is: $U_{\mathbf{x}}(d) = \sum_{i=1}^{N} x_i U_i(d)$.

Bubble Rap does not originally limit the number of copies, but this is easily accomplished with only insignificant modification of the algorithm.

For all three problems, the respective Markov chain $\mathbf{P}$ from Eq. (10) is now entirely defined[9] and we can apply the convergence analysis from Section III to each of them. While content placement fundamentally differs from routing (in one problem, node characteristics are sought for, in the other the nodes themselves), our three example problems are suddenly similar: same state space, just different utilities. This is, to a great extent, the merit of our *unified* framework DTN-Meteo, whose declared goal is to exploit the similarities of DTN problems and algorithms. The seemingly small difference in utilities is, in fact very consequential. It has the following effects: (i) content placement, has a single utility function per network scenario – in routing, we must evaluate a collection of utilities (one per destination d) for each network; (ii) content placement usually has a single optimal state – in routing, for each utility function or destination d, there are $\binom{N-1}{L-1}$ optimal states (any subset of L nodes containing d); (iii) local maxima (and thus an algorithm's behavior) radically change with the utility function, as shown in [26].

B. Measuring the Accuracy of DTN-Meteo

To cover a broad range, we use three real-world connectivity traces and two synthetic mobility traces for validation: (i) the *Reality Mining* trace (MIT) [9], (ii) the Infocom 2005 trace (INFO) [32], (iii) the ETH trace [10], and (iv) two synthetic scenarios created with a recent mobility model (TVCM) [33]. Their characteristics are summarized in Table II.

In simulations, we measure absorption quantities, starting from every node separately. The starting node generates messages using a Poisson process. This ensures, via the PASTA property [34], that we do not introduce any sampling bias. The message generation process produces a sample of at least 1 000 observations per source node for shorter traces and up to 15 000 observations per source node for longer ones. For all measured delays, we compute the 95$^{\text{th}}$ percentile using the normal distribution.

	L	Optimum		Best local max.	
		pred.	meas.	pred.	meas.
ETH	5	1.00	1.00	N/A	N/A
Infocom	3	0.34	0.36	0.18	0.20
MIT	2	0.39	0.38	0.42	0.42
TVCM24	4	0.10	0.10	0.38	0.38
TVCM104	2	0.60	0.48	0.36	0.44

TABLE III
ABSORPTION PROBABILITIES

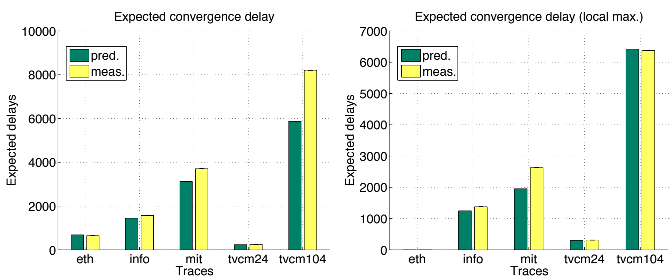
[8]For parameters α and β, we use the original paper values: $\alpha = \beta = 0.5$.

[9]The state space Ω is formed by all L-node subsets of $\mathcal{N}$, i.e., $|\Omega| = \binom{N}{L}$, as we ignore the delay of the initial replication phase.

	MIT	INFO	ETH	TVCM
Scale and context	92 campus students & staff	41 conference attendees	20 lab students & staff	24/104 nodes, 2/4 disjoint communities
Period	9 months	3 days	5 days	11 days
Scanning Interval	300s (Bluetooth)	120s (Bluetooth)	0.5s (Ad Hoc WiFi)	N/A
# Contacts total	81 961	22 459	22 968	1 000 000

TABLE II
MOBILITY TRACES CHARACTERISTICS.

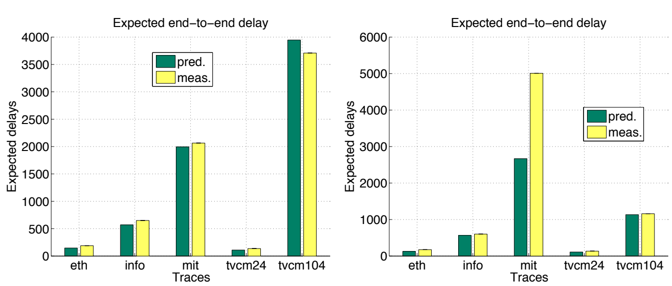
Tab. III shows the measured vs. predicted (Thm. 1) success probabilities of content placement. The first two columns give the probability of absorption by the global optimum, the second two – the probability of absorption by a local maximum. In the majority of cases, the prediction is reliably accurate, both with a single absorbing state, the global maximum (in ETH) and when local maxima are present, resulting in multiple absorbing states (in Infocom, MIT, TVCM24, TVCM104). The delivery ratios of the routing algorithms show similar accuracy; we omit them due to space limitations.



(a) Absorption by optimum (b) Absorption by local max.

Fig. 2. Prediction accuracy for content placement

Fig. 2 compares the measured and predicted (Thm. 2) values of the convergence delay of content placement, averaged over all initial states for the greedy algorithm (L values as in Tab. III). The theoretical results coincide once again surprisingly well with the measured delays, both for absorption by the optimum state – Fig. 2(a), and for absorption by a local maximum – Fig. 2(b). In Fig. 2(b), the ETH trace does not have any local maxima with our utility.



(a) SimBet (b) BubbleRap

Fig. 3. Prediction accuracy for routing

Fig. 3 compares the measured and predicted (Thm. 2) values of the end-to-end delivery delay of SimBet and BubbleRap routing, averaged over all initial states (L values as in Tab. III). Again, the theoretical results of both algorithms coincide well with the measured delays, with the exception of BubbleRap routing on the MIT trace. This points to a potential sensitivity of our model to the utility function, which we plan to further investigate in the future.

Summarizing, we have used DTN-Meteo to model three DTN communication algorithms and obtained closed form results for crucial performance metrics for these algorithms (delivery/convergence probabilities and delays). Despite simplifying assumptions, our models' forecasts are reliably accurate, under a large variety of real and realistic mobility scenarios. *To our best knowledge, this is the first analytical work that can accurately predict performance metrics for utility-based algorithms and general, heterogeneous mobility.*

The framework we have described can evaluate almost any single-copy DTN communication algorithm for relatively large network sizes (order 10^3), as well as multi-copy algorithms with low replication. However, the computational complexity may become prohibitvely large with increasing replication L. In the next section, we briefly analyze this complexity and propose further simplifications to the framework trading accuracy for greater applicability.

V. A NOTE ON STATE SPACE SIZE

The size of our state space Ω varies from linear in network size (N or $poly(N)$) for low replication, to exponential ($\leqslant 2^N$) in the worst case (epidemic replication). This affects Eq. (12), where finding the fundamental matrix requires the inversion of a $t \times t$ matrix. The number of transient states, t, may have the order of magnitude of Ω or it may be significantly smaller (depending on the contact pattern $\mathbf{P}^c$ and on the utility function). The fastest known matrix inversion algorithm takes $O(n^{2.376})$ operations (with n the matrix size). This allows the treatment of fairly large networks for single-copy or low replication algorithms, but becomes practically challenging with high or uncontrolled replication. To alleviate this, we propose a delay approximation technique to deal with the complexity caused by increased replication L. We also examine the quality of this approximation, by comparison with the exact predictions from Section IV, as well as with simulation results.

As we have seen in the previous sections, modeling the evolution in the network, of a single copy of an object over time is computationally feasible for fairly large networks. Our analysis yields absorption probability and delay results for every pair: (starting node $\to$ finishing node). From these quantities, a large variety of aggregates can be derived, that have high practical value. Below we examine under which conditions we can use the pairwise absorption delays for $L = 1$, to approximate absorption delays for multi-copy DTN communication algorithms. Due to space limitations, we provide intuitive arguments for our approximation and defer a more formal treatment (including error bounds) for future work.

1) ***The Independence of Copies:*** In multi-copy DTN algorithms, copies may be dependent in two ways: First, as implied in Section II-D, a state's utility $U_{\mathbf{x}}$, may not be decomposable in node utilities $U_i x_i$. In this case, copies cannot be followed separately (i.e., as L identical independent N-state Markov chains running in parallel). However, due to the difficulty of locally estimating global quantities in

DTNs, almost all utilities are decomposable – we adopt this assumption henceforth. Second, in most algorithms, copies interact with each other: when two nodes meet and both carry copies of the same message or content, no exchange/transition happens, although one's utility may be higher than the other's.

Let D, the dependency among copies, be measured as the "number of node encounters per time unit, where forwarding should have occured, but did not, due to dependency". Consider the evolution of this metric in function of the L to N ratio, shown in Table IV. The dependency D decreases significantly with the L to N ratio. In other words, the L copies – and thus, the L walks on the N-state Markov chain – become almost independent, for small L to N ratios. Clearly, the independence of copies is a valid assumption, which we will use for the rest of the analysis.

L/N	0.25	0.20	0.15	0.10
D ($\times 10^{-5}$)	220	130	63.6	3.45

TABLE IV
COPY DEPENDENCY VS. L TO N RATIO

*2) **One Chain per Copy**:* With the above assumption, we can simply run L identical N-state Markov chains (with different starting nodes), to find the individual expected convergence delays of each copy. Combining these, we can obtain the overall expected convergence delay from initial state $\mathbf{x}$ to an absorbing state $\mathbf{y}$. Below we explain how to do this for both content placement and routing.

Lemma 3 (Routing Approx.): The expected end-to-end delay for L-copy routing from starting nodes $s_1, \ldots, s_L$ to node d is:

$$\tau_{rt} = \mathbb{E}[T_{rt}] = \mathbb{E}\left[\min(T_{s_1 d}, \ldots, T_{s_L d})\right]. \tag{20}$$

where $T_{s_1 d}, \ldots, T_{s_L d}$ are the end-to-end delays from each starting node s_i to the destination d.

Intuitively, we can explain this lemma as follows: The L copies start at different nodes $s_1, s_2, \ldots, s_L$ (as a consequence of the negligible initial distribution phase). We are only interested in *one* of the copies reaching the destination, regardless of the positions of the other $L-1$ copies. Then, the overall convergence delay for L-copy routing is the minimum of the L independent delivery delays as in Eq. (20).

Lemma 4 (Content Placement Approx.): The expected convergence delay for L-copy content placement from any starting nodes to the optimal relays $n_1, \ldots, n_L$ (with $U(n_1) \geqslant U(n_2) \ldots \geqslant U(n_L)$) is:

$$\tau_{cp} = \mathbb{E}[T_{cp}] = \mathbb{E}\left[\sum_{i=1}^{L} T_i^{(L-i+1)}\right]. \tag{21}$$

where $T_i^{(L-i+1)}$ is the residual delay (after $i-1$ copies have been absorbed at $n_1, \ldots, n_{i-1}$) of any of the $L-i+1$ remaining copies being absorbed at n_i.

The key to Eq. (21) is to consider the occupation of the optimal relays n_i in the order of their utility. The total content placement delay is the sum of: (i) the time until the best relay n_1 received any one of L copies; plus (ii) the *remaining* time until the second best relay n_2 received any one of $L-1$ copies; and so on until the last relay receives the last copy. Because we do not distinguish between copies, each delay component of the sum is the minimum of $L-i+1$ individual delays, as in Eq. (20). In addition, here, we also do not distinguish between starting nodes. Thus the $L-i+1$ individual delays are identical: the *residual* absorption time starting from any node with equal probility to relay n_i. We denote its expectation τ_i.

*3) **Absorption Time Distribution**:* To find the minimum in Eq. (20) and the minima and sum of random variables in Eq. (21), we need their distributions. The variables are absorption times of a finite Markov chain; as such, they follow a *phase-type distribution*. In fact, phase-type distributions are defined by an absorbing Markov chain ***with a single absorbing state*** and by its initial distribution vector [35].

Phase-type distributions have two very useful properties, both proven in [35]: (i) they have exponential tails, and (ii) if absorption rates/probabilities (to the single absorbing state) are small, not only the tail, but the entire distribution is asymptotically exponential. Although the condition for the second property may not always be fulfilled in our scenarios, we believe that in our context, the first property alone justifies an approximation of the individual node absorption times by exponential/geometric distributions.

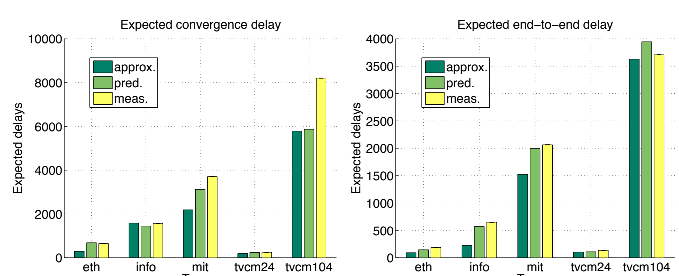
Using this approximation and copy independence, the calculations in Eqs. (20) and (21) are straightforward. The minimum of L independent geometric random variables with parameters $p_k = \tau_k^{-1}$ and with $q_k = 1 - p_k$, is also geometric, with expectation $\tau_{\min} = \left(1 - \prod_{k=1}^{L} q_k\right)^{-1}$. Then, Eq. 20 becomes:

$$\tau_{rt} = \left(1 - \prod_{k=1}^{L}\left(1 - \tau_{s_k d}^{-1}\right)\right)^{-1}. \tag{22}$$

In Eq. (21), each term $T_i^{(L-i+1)}$ of the sum is also a minimum of $L-i+1$ identical individual *residual* absorption times. By the independence of copies, these *residual* absorption times are mutually indepedent. In addition, because absorption times are geometric – a memoryless distribution – *residual* absorption times are also geometric with the same original parameter τ_i^{-1}. Parameters are obtained from the single-copy Markov chain, using Thm. 2. Then, random variables $T_i^{(L-i+1)}$ are also geometric with expectation $\tau_i^{(L-i+1)} = \left(1 - (1 - \tau_i^{-1})^{L-i+1}\right)^{-1}$, calculated as in Eq. (22). By the memoryless property of the geometric distribution, variables $T_i^{(L-i+1)}$ are also indepedent, although they are residual of one another. Then, the sum in Eq. (21) is a sum of the independent geometric variables and, by the linearity of expectation,

$$\tau_{cp} = \sum_{i=1}^{L} \tau_i^{(L-i+1)} = \sum_{i=1}^{L}\left(1 - \left(1 - \tau_i^{-1}\right)^{L-i+1}\right)^{-1}. \tag{23}$$

Using Eqs. (22) and (23), we can now predict delays for much larger scenarios, where L is virtually unhampered.



(a) Content Placement (b) SimBet Routing

Fig. 4. Approximation accuracy

Fig. 4 shows our evaluation of the precision of the above approximation scheme. We compare approximate predictions with simulation results and with exact predictions, for content placement and SimBet routing (BubbleRap results are similar). We reuse the same L-values as before, to bound simulation time, as well as to enable comparison with exact predictions. While the predictions have traded accuracy for increased applicability, the results are still sufficiently precise.

VI. Discussion and Applications

All in all, despite: (i) the complexity of heterogeneous mobility, (ii) the complexity and diversity of the problems and algorithms considered, and (iii) simplifying assumptions ensuring tractability of the above, DTN-Meteo predicts relevant performance metrics for Routing and Content Placement surprisingly accurately under a wide variety of real and realistic mobility scenarios. *To our best knowledge, this is the first analytical work that can predict performance for utility-based algorithms with general, heterogeneous mobility.* Moreover, while our current results have focused on these two problems, the main components of DTN-Meteo are generic and should enable accurate performance predictions for other problems. For example, DTN multicast [36] can be modelled similarly to content placement, by simply redefining the utility. In future work, we intend to conduct a similar performance analysis for more problems (buffer management, anycast, multicast, etc), to further validate the merit of DTN-Meteo.

In addition, we plan to further develop the analysis, so as to investigate more sophisticated initial replication strategies in fixed budget algorithms, as well as unlimited and implicitly limited replication (preliminary results in [17]). A second planned improvement is the support for time-inhomogeneous utilities. This is well achievable for contact processes with multiple alternating stationary regimes (e.g., day–night), where utilities can be estimated and used separately for each. Finally, we will derive bounds for the errors of all our approximations.

Beyond the theoretical aspects of our analysis, the presented model is useful to protocol or system designers. Firstly, DTN-Meteo offers valuable insight into a protocol's inner-workings: e.g., a small delivery ratio can be directly linked to the presence of local maxima or to an insufficient TTL, compared to the predicted average delay. Consequently, the TTL or the replication constraints can be tuned to achieve target delivery parameters, as shown in Table V for SimBet on ETH.

Desired Expected Delay <	475	285	190	135
Minimum Required L	1	2	3	4

TABLE V
Minimum L to achieve expected delay in ETH with SimBet

Moreover, DTN-Meteo can help tune parameters of a utility function (e.g. α and β in SimBet) or compare two functions, to have as little local maxima as possible and good delays.

VII. Conclusion

In this paper, we presented DTN-Meteo a generic model and analytical framework for DTN algorithms. Unlike earlier analytical research work, our model captures the full heterogeneity of node mobility, which has been observed in real scenarios [9], [10], even for sophisticated utility-based algorithms. Moreover, our framework allows the examination of a larger class of algorithms, instances of which are very frequently proposed as solutions to DTN problems: utility-ascent/descent algorithms, be they probabilistic of deterministic.

References

[1] Vahdat A and Becker D. *Epidemic routing for partially connected ad hoc networks*. Tech. rep., 2000.
[2] Spyropoulos T, Psounis K et al. *Spray and Wait: an efficient routing scheme for intermittently connected mobile networks*. WDTN 2005.
[3] Balasubramanian A, Levine B et al. *DTN routing as a resource allocation problem*. SIGCOMM, 2007.
[4] Picu A and Spyropoulos T. *Distributed stoch. optimization in Opportunistic Nets: The case of optimal relay selection*. CHANTS 2010.
[5] Hanbali AA, Kherani A et al. *Simple models for the performance evaluation of a class of two-hop relay protocols*. IFIP Networking 2007.
[6] Groenevelt R, Nain P et al. *The message delay in mobile ad hoc networks*. Perform Eval, 62, 2005.
[7] Spyropoulos T, Psounis K et al. *Performance analysis of mobility-assisted routing*. MobiHoc 2006.
[8] Zhang X, Neglia G et al. *Performance modeling of epidemic routing*. IFIP Networking 2006.
[9] Eagle N and Pentland A. *Reality mining: sensing complex social systems*. Pers Ubiq Comput, 10(4):255–268, 2006. ISSN 1617-4909.
[10] Lenders V, Wagner J et al. *Measurements from an 802.11b mobile ad hoc network*. WOWMOM 2006.
[11] Daly E and Haahr M. *Social network analysis for routing in disconnected delay-tolerant MANETs*. MobiHoc 2007.
[12] Boldrini C, Conti M et al. *Modelling social-aware forwarding in opportunistic networks*. PERFORM 2010.
[13] Spyropoulos T, Turletti T et al. *Routing in delay-tolerant networks comprising heterogeneous node populations*. IEEE TMC, 8, 2009.
[14] Chaintreau A, Le Boudec JY et al. *The age of gossip: spatial mean field regime*. SIGMETRICS 2009, 109–120.
[15] Daly E and Haahr M. *Social network analysis for information flow in disconnected delay-tolerant MANETs*. vol. 8. 2009.
[16] Hui P, Crowcroft J et al. *Bubble rap: social-based forwarding in delay tolerant networks*. MobiHoc 2008.
[17] Picu A, Spyropoulos T et al. *An analysis of the information spreading delay in heterogeneous mobility DTNs*. Tech. Rep. 341, ETHZ, 2011.
[18] Chaintreau A, Hui P et al. *Impact of human mobility on the design of opportunistic forwarding algorithms*. INFOCOM 2006.
[19] Karagiannis T, Le Boudec JY et al. *Power law and exponential decay of inter contact times between mobile devices*. MobiCom 2007, 183–194.
[20] Gunawardena D, Karagiannis T et al. *Scoop: decentralized and opportunistic multicasting of information streams*. MobiCom 2011, 169–180.
[21] Grossglauser M and Tse DNC. *Mobility increases the capacity of ad hoc wireless networks*. IEEE ToN, 10, 2002.
[22] Spyropoulos T, Psounis K et al. *Efficient routing in intermittently connected mobile networks: the single-copy*. IEEE ToN, 16(1), 2008.
[23] Lindgren A, Doria A et al. *Probabilistic routing in intermittently connected networks*. SIGMOBILE MC2R, 2003.
[24] Baccelli F and Brémaud P. Elements of Queueing Theory. 2003.
[25] Carreras I, Miorandi D et al. *A simple model of contact patterns in delay-tolerant networks*. Wireless Networks, 16, 2010.
[26] Picu A and Spyropoulos T. *Distributed optimization in DTNs: Towards understanding greedy and stoch. algos*. Tech. Rep. 326, ETHZ, 2010.
[27] Hossmann T, Spyropoulos T et al. *A complex network analysis of human mobility*. NetSciCom 2011.
[28] Kemeny J and Snell L. Finite Markov Chains. 1960.
[29] Hossmann T, Spyropoulos T et al. *Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing*. INFOCOM 2010.
[30] Picu A and Spyropoulos T. *Minimum Expected *-cast Time in DTNs*. BIONETICS 2009.
[31] Blondel V, Guillaume JL et al. *Fast unfolding of communities in large networks*. Journal of Stat Mech: Theory and Exp, 2008(10), 2008.
[32] Hui P, Chaintreau A et al. *Pocket switched networks and human mobility in conference environments*. WDTN 2005.
[33] Hsu WJ, Spyropoulos T et al. *Modeling spatial and temporal dependencies of user mobility in wireless mobile networks*. IEEE ToN, 2009.
[34] Wolff R. *Poisson Arrivals See Time Averages*. Operations Research, 30(2), 1982.
[35] Asmussen S and Albrecher H. Ruin Probabilities. 2nd ed., 2010.
[36] Gao W, Li Q et al. *Multicasting in delay tolerant networks: a social network perspective*. MobiHoc 2009.