Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

DTN-Meteo: Forecasting the Performance of DTN Protocols under Heterogeneous Mobility

Andreea Picu, Thrasyvoulos Spyropoulos

TIK Report No. 348*
November 2012

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 local search heuristics, choosing a 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, these 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 over heterogeneous mobility (contact) models, into a Markov chain traversal over the relevant solution space. (Heterogeneous) meeting probabilities between different pairs of nodes dictate the chain's transition probabilities and determine neighboring solutions. Local optimization algorithms can accept/reject candidate transitions (deterministically or randomly), thus "modulating" the above transition probabilities. We apply our model to two example 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.

1 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, nearly all algorithms proposed for each problem are *local search heuristics*, based on either (i) *deterministic* choices (e.g. greedy) of the next candidate solution or (ii) *randomized* choices (e.g. simulated annealing [5]). Moreover, the performance evaluation of these algorithms 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, distributed caching, buffer management) and the multitude of algorithms for them.

Early analytical models for DTNs were devised [6–9] to complement simulations. 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 [10, 11] 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, 12]. 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 allowing for mobility heterogeneity [13–15] suffer from at least one of the following shortcomings: they introduce a very limited degree of heterogeneity to existing models; they quickly become prohibitively complex; or they deal with simple protocols only. It is thus evident that *a common analytical framework* is needed that can suc-

[*] This is a revised and considerably extended version of TIK Report No. 326. from July 2010.

cessfully 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 current effort in designing such a common analytical framework, which we entitle DTN-Meteo, is based on the following **three common features** of most DTN protocols. Firstly, 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 aims at moving to better states. Secondly, 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*. And thirdly, due to the difficulty (usually impossibility), in this context, of updating nodes' states globally, protocols resort to *local search* heuristics (with a deterministic or randomized search strategy) 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 a combinatorial optimization problem into a Markov chain (MC), where each state is a potential solution (e.g. assignment of content replicas to nodes), 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 time-invariant utility function and accepts better ones either deterministically (*greedy algorithm*) or randomly, attempting to navigate around or escape local maxima[1] (*randomized local search algorithm*). This not only decouples the algorithm's effect and the mobility's effect from each other, but also allows one to derive interesting performance metrics (convergence delay, delivery probability) using transient analysis of this MC.

Summarizing, our main contributions are:

- We formulate a class of important 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 3). This model enables us to quantify the performance – delay, success probability (Sections 4.2 and 4.3) – of fixed-replication algorithms in various mobility scenarios and to evaluate their correctness (Section 4.1).
- We prove that, for state-of-the-art intercontact models (namely, power law with exponential cutoff), compliant with observations across a wide range of real mobility scenarios [16, 17], the points in time at which we embed the above Markovian model are indeed well approximated by an exponential distribution (Section 3.4).
- To demonstrate the value and generality of DTN-Meteo, we apply it to both single- and multi-copy algorithms for two DTN problems: (i) Unicast routing (SimBet [12] and BubbleRap [18]) 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 5).

While DTN-Meteo is more widely applicable, due to space limitations, in this work we mostly focus on algorithms which explicitly limit the number copies of a message at the source [4, 12]. The two other options with respect to replication in DTN algorithms, are either unlimited replication [1] or implicitly limited through utility [18]. We treat the first case (unlimited replication or flooding-based schemes) in [19]. As for the second option, utility-limited replication, we examine in this paper the most important part of such an algorithm, its utility function, by simply forcing it into the first category.

2 Preliminaries

Early analytical studies of Opportunistic Networking algorithms rely on simple, identical node mobility assumptions, where nodes meet each other at independent identically distributed (IID) time intervals, that are exponentially distributed. There is a unique meeting rate λ, describing the contacts of every node pair. As a result, all nodes are equal and can be treated as a group, rather than individually. This is reflected, e.g., in epidemic routing Markov models, where only the number of message copies is modeled, without regard for the specific nodes carrying those copies, as in Fig. 1.

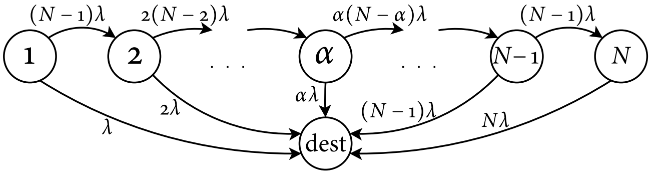


Figure 1: *Markov chain from [6]*

DTN analytical models relying on such mobility assumptions have been successfully applied to: (i) *"oblivious" unicast routing protocols*, such as direct transmission, two-hop routing, epidemic routing and its variants [7, 20] (probabilistic epidemic, time-limited epidemic, epidemic with the VACCINE recovery mechanism etc), (ii) *simple content distribution algorithms* [21], (iii) simple multicast routing protocols [22], and so on. The common denominator of all these protocols is that none of them explicitly differentiates among nodes, based on their characteristics (mobility, resources or other), when making a forwarding/dropping decision.

In contrast, almost all newer DTN algorithms and protocols [4, 18, 23, 24] base their decisions on nodes' utilities towards the scheme's end goal (position in the network, relationship to destination etc). These node utilities are most

[1] For example, by randomly accepting lower utility solutions and always accepting higher-utility ones, as in simulated annealing [5].

$\mathcal{N}$ and N	the network (set of nodes), resp. its size
i, j, k	nodes in $\mathcal{N}$
L	maximum allowed replication (number of copies)
$\mathcal{S}$ and Ω	node, resp. network state space
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	network states (elements of Ω)
$U(\mathbf{x})$, $u(i)$	utility of a network state, resp. of a node
$\delta(\mathbf{x}, \mathbf{y})$	difference between network states **x** and **y**
$A_{\mathbf{xy}}$	acceptance probability for transition **xy**
p^c_{ij}	contact probability between nodes i and j
$\mathbf{P} = \{p_{\mathbf{xy}}\}$	Markov transition probability matrix
$T^{(ij)}_m$, T_n	starting times for (i, j) contacts , resp. any contacts
$J^{(ij)}_m$, J_n	intercontact times for (i, j), resp. any contacts
$K^{(ij)}(t)$	residual intercontact times for (i, j)
α_{ij}	power law exponent of (i, j) intercontact times
β_{ij}	rate parameter of (i, j) intercontact times
F_{ij}, $\overline{F_{ij}}$	CDF, resp. CCDF of (i, j) intercontact times

Table 1: *Important notation in Section 3.*

often based on the observation that human mobility has non-trivial structure, manifesting in meeting patterns which can be extracted from real world experiments [10, 11].

Nevertheless, introducing different contact rates per pair complicates the above picture significantly: one now needs to keep track not just how many, but which nodes exactly have the message, exploding the state space. Adding the notion of utilities quickly makes this approach intractable. As a result, analytical work has not, for the most part, followed up on these developments.

DTN-Meteo is a proposal to enable meaningful analysis, despite this additional complexity, striking a balance between tractability and accuracy. The state space of the MC is also expanded, to have each state now correspond to a different node configuration (of interest), but in a manner that incorporates both contact heterogeneity and utilities such that: (a) their effect is easily separable, and (b) the concept of an absorbing MC is maintained, out of which delivery delays and ratios can be easily computed.

All in all, DTN-Meteo is a considerable step forward from current DTN analyses [6, 15, 20] for two reasons: (i) it is **more generic**, in that it applies to a wide selection of utility functions and to more than one specific problem (e.g., routing) and (ii) it uses a **more realistic mobility model** (with the inevitable complexity tradeoff).

3 DTN-Meteo: 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, in the sense of [25]. 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.

	Replication		**Constraint**
i)	single-copy	[23, 26, 27]	$\sum_{i=1}^{N} x_i = 1$
ii)	fixed budget L	[2, 12]	$\sum_{i=1}^{N} x_i \leqslant L$
iii)	epidemic	[1]	$\sum_{i=1}^{N} x_i \leqslant N$
iv)	utility-based[2]	[3, 18, 28]	$\sum_{i=1}^{N} x_i \leqslant N$

Table 2: *Modeling replication with Eq. (4)*

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.

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

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

- a set of **constraints** $\qquad f_i(x_1, \ldots, x_N) \leqslant \rho_i$ (4)
- a **utility function** $\qquad U : \Omega \mapsto \mathbb{R}$ (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

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

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

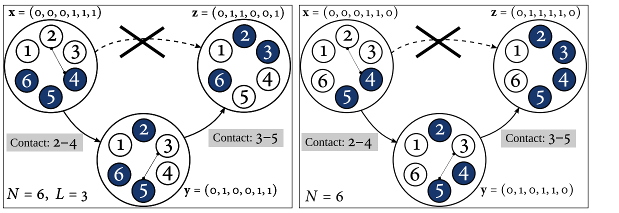
Here, the binary state of a node i is interpreted as the node being ($x_i = 1$) or not being ($x_i = 0$) a provider for the content of interest[4]. There is a single constraint for any allowed solution, namely $\sum_{i=1}^{N} x_i \leqslant L$, that is, in Eq. (4) $\rho_i = \rho = L$.

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 $\mathbf{x}_0 = (0, 0, \ldots, \{x_s = 1\}, \ldots, \{x_d = 0\}, \ldots, 0)$. 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 2. 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 2.

As a final note, replication-based schemes have an initial spreading phase, during which the L copies are distributed to the initial relays. This phase can straightforwardly be included in our model, by considering also all states with 1 to $L-1$ copies in the network. This, however, increases the complexity of the model, while offering little added value. Indeed, the delay of this initial phase is negligible in networks with large N and $L \ll N$, which is the case in scenarios of interest (see e.g. [2]). Hence, we will only consider the effect of this phase on the starting configuration (as shown in Section 4.2, Eq. (22)) of the algorithm and ignore its delay. Thus, our solution space Ω, will only be composed of network states with exactly L copies at L different nodes.

3.2 Exploring the Solution Space



(a) Forwarding (b) Replication

Figure 2: *Example state transitions in two 6-node DTNs.*

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 (purely greedy algorithms). Occasionally, they may also move to lower utility states (using randomization) in order to overcome local maxima, as in randomized local search algorithms (e.g. 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*.

Consider implication (i) first (we treat (ii) when defining our network model, in Section 3.4). 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. 2 provides examples of *potential* state transitions, along with contacts required for the transitions to be possible.

Transition $\mathbf{xz}$ in Fig. 2 requires two contacts to *start* simultaneously, and is thus not possible[5]. This means that only transitions between *adjacent* states are possible, where we define state adjacency as:

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

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.

From Fig. 2 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.*

Summarizing, our solution space exploration goes as follows: When at a state $\mathbf{x}$, the next contact between two nodes i, j presents a new solution $\mathbf{y}$ to the algorithm with a (time-homogeneous) probability p^c_{ij}, the contact probability

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

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

[5] We emphasize that this is not a strict necessity for our networks of interest, but rather a technical assumption of our mobility model, for analytical convenience (see Section 3.4 for more details). However, given the relative sparsity of the networks in question, such events occur with low probability.

of node pair (i, j) (we look deeper into these contact probabilities and the assumptions behind them in Section 3.4). The new solution $\mathbf{y}$ differs from $\mathbf{x}$ in positions i and j only. For example, in Fig. 2(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}.

3.3 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 purely 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***[6]:

$$A_{\mathbf{xy}} = \mathbb{1}\{U(\mathbf{x}) < U(\mathbf{y})\}. \tag{7}$$

More generally, the acceptance probability may be any function of the two utilities: $A_{\mathbf{xy}} \in [0,1]$. This allows DTN-Meteo to model randomized local search algorithms (e.g. simulated annealing) as well.

As mentioned in the introduction, we are considering here time-invariant (or very slowly varying) utilities, such as ones derived from social relationships among humans carrying the devices (nodes) which form our network. These relationships are reflected in the observed node mobility, as revealed in recent experiments [10, 11]. This enables the designing of mobility-based utility functions for e.g., routing [18, 23], content placement [4] etc. Additional types of utilities that are time-invariant (or very slowly varying) that DTN-Meteo can support include functions based on computational power, operating system, long-term traffic patterns etc.

Moreover, the utility value may not be readily available locally at each node. Therefore, the algorithm must solve the added problem of estimating it – usually through an integrated sampling component [3,28]. As mentioned above, most DTN protocols suppose the existence of a mobility-related node utility $u : \mathcal{N} \mapsto \mathbb{R}$ (e.g. contact frequency). Thus, if the mobility is stationary and the estimation designed correctly, the online and offline utility rankings will coincide. However, estimating node characteristics and utilities is beyond the scope of this work. Therefore, DTN-Meteo assumes any state utility U and/or node utility u are readily available.

Summarizing from sections 3.1, 3.2, and 3.3, the transition probability between *adjacent* network states $\mathbf{x}$ and $\mathbf{y}$ can be expressed in function of the contact probability (to be discussed in Section 3.4) and the acceptance probability as:

$$p_{\mathbf{xy}} = p^c_{ij} \cdot A_{\mathbf{xy}}, \tag{8}$$

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

The algorithm component $A_{\mathbf{xy}}$ is *time-homogeneous* (as per our above discussion of the time-invariance of utility functions) and *memoryless*, i.e. it has the Markov property (as most utility-based DTN algorithm do not include past states in their decisions).

The mobility component p^c_{ij} (contact probability between i, j) is also *time-homogeneous* (as human relationships tend to be relatively stable over time) and memoryless. This will be discussed in depth in the next section, where we show how node mobility can be modeled, based on existing DTN experiments [10, 11] and analyses thereof [16, 29, 30].

Therefore, for any two states $\mathbf{x}$ and $\mathbf{y}$, our algorithm is a discrete-time time-homogeneous Markov chain $(\mathbf{X}_n)_{n\in\mathbb{N}_0}$ over the solution space Ω, described by the transition probability matrix $\mathbf{P} = \{p_{\mathbf{xy}}\}$, with:

$$p_{\mathbf{xy}} = \mathbb{P}\left[\mathbf{X}_{n+1} = \mathbf{y} \,|\, \mathbf{X}_n = \mathbf{x}\right] = \begin{cases} 0, & \delta(\mathbf{x},\mathbf{y}) > 2 \\ p^c_{ij} \cdot A_{\mathbf{xy}}, & 0 < \delta(\mathbf{x},\mathbf{y}) \leqslant 2 \\ 1 - \sum\limits_{\mathbf{z}\neq\mathbf{x}} p_{\mathbf{xz}}, & \mathbf{x} = \mathbf{y}. \end{cases} \tag{9}$$

with i, j as before.

This formulation allows us to transform any problem of our defined class into a simple Markov chain, which can be used for perfomance analysis and prediction.

3.4 Modeling Heterogeneous Node Mobility

Recall our Opportunistic Network $\mathcal{N}$, with $|\mathcal{N}| = N$ nodes. As data is exchanged only upon contacts in $\mathcal{N}$, a mobility model based on contact patterns is sufficient for our analysis. To describe the heterogeneous mobility network $\mathcal{N}$, we use the following model for each and every pair of nodes in the network. (Main implications of the model are briefly discussed immediately following Def. 1.)

Definition 1 (Node pair processes). *Every node pair's contact process is an* independent, *stochastic,* stationary, *and ergodic renewal process* $\left(T^{(ij)}_m\right)_{m\in\mathbb{Z}}$*, where* $T^{(ij)}_m$ *denotes the starting time of a contact between nodes* (i, j)*. The random variables* $J^{(ij)}_m = T^{(ij)}_m - T^{(ij)}_{m-1}$ *are the times between the initiations of two successive contacts between nodes* (i, j)*. The process being renewal implies that the times* $J^{(ij)}_m$ *are IID with a generic distribution.*

[6] In a *distributed* algorithm, the utilities of two states must be comparable locally by the two nodes in contact (e.g. additive function $U(\mathbf{x}) = \sum_{i=1}^{N} x_i u(i)$, where $u : \mathcal{N} \mapsto \mathbb{R}$ can be applied to each node separately). This is not a requirement for our model, but a challenge for the algorithm designer [4].

As far as *stationarity* is concerned, human mobility is known to exhibit piecewise stationary regimes (day/night and weekday/weekend periodicity). When the full stationarity assumption proves unacceptable, these regimes can be treated separately. Def. 1 also implies that every node pair's contacts are mutually *independent* from the contacts of all other node pairs. This is a step forward from the usual assumption that every node moves independently from all other nodes. When all nodes move mutually independently, all contacts happen purely by chance. With our approximation, if nodes i and j come in contact, their two mobilities may be correlated (i.e. they may meet on purpose); however, their contact will happen independently of any other node pairs' mobility, including node pairs involving one of i or j. This is because correlation is not transitive [31].

We must stress here, that we do not claim all these assumptions to hold in real scenarios. They just represent our chosen tradeoff between the tractability and prediction accuracy of our model, the latter being validated against real traces.

Corollary 1 (The network process). *The entire network $\mathcal{N}$ is described by the superposition of all the individual $\left(T_m^{(ij)}\right)_{m\in\mathbb{Z}}$ processes (which are mutually independent), forming a new stationary and ergodic marked point process $(M_n)_{n\in\mathbb{Z}} = \{T_n, \sigma_n\}$, with $\sigma_n = (i, j)$. Therefore, the newly indexed time instants $(T_n)_{n\in\mathbb{Z}}$ are epochs of a* superposition of renewal processes. *The new holding times are $J_n = T_n - T_{n-1}$.*

Making the following two technical provisions, the model is summarized in Fig. 3 for a small toy network of 4 nodes:

(a) $T_n < T_{n+1}$, $\forall\, n \in \mathbb{Z}$ – the probability of two contacts starting at exactly the same time is negligible[7], i.e. $J_n > 0$.

(b) the duration of a contact is negligible compared to the time between two contacts, but sufficient for all data transfers to take place.

As made evident in Fig. 3, the holding times $J_n = T_n - T_{n-1}$ of the superposition process are, in fact, minima of residual times of the pairwise holding times $J_m^{(ij)}$. Since we are planning to define a Markov model on the network's contact process, the probability distribution of the superposition's holding times $J_n = T_n - T_{n-1}$ is crucial.

Based on the above contact model, it is easier to see how our transition probability matrix $\mathbf{P}$, defined in Eq. (9) fits into the picture. We are essentially embedding a discrete time process at points T_n, the epochs of the superposition process. Whether this embedded process is indeed Markovian and time-homogeneous will thus depend on the superposition process $(M_n)_{n\in\mathbb{Z}}$ and its holding times J_n.

If the individual contact processes $\left(T_m^{(ij)}\right)_{m\in\mathbb{Z}}$ are assumed to be Poisson (i.e., pairwise intercontacts $J_m^{(ij)}$ are exponential), an assumption commonly made in most related literature, then it is easy to see that their residuals are exponential as well. This means that the superposition epochs T_n define a Poisson process and the (embedded) Markov chain of Eq. (9) can be used to analyze performance, despite the heterogeneous contact rates.

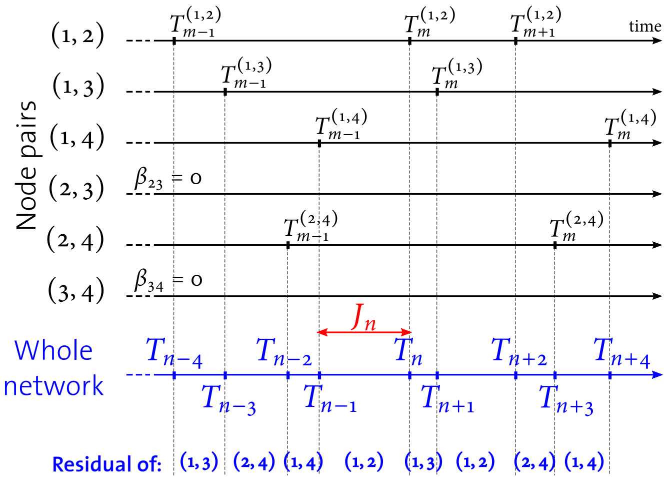


Figure 3: *Contact model for a toy network with* 4 *nodes and* 4 *out of* 6 *node-pairs contacting each other.*

However, the assumption of purely exponential intercontact times has been challenged by real collected contact traces [29]. Recently, a solid case has been made [16, 17] for pairwise intercontact times $J_m^{(ij)}$ being distributed as a power law with an exponential cutoff. Using this as a starting point, we will investigate in the following, whether the new pairwise residual times (which determine holding times J_n) allow us to maintain our Markov model and proceed with the analysis.

The Keystone of DTN Analysis: (non-)Exponentiality of Intercontact Times.

As shown in Fig. 3, the holding times J_n of our superposition process $M_n = \{T_n, \sigma_n\}$ are, in fact, minima of residual times of the original source processes $\left(T_m^{(ij)}\right)_{m\in\mathbb{Z}}$, representing *pairwise* intercontacts. Thus, in order to say something about J_n, we must thoroughly understand the probability distributions of $J_m^{(ij)}$.

Cai et al. showed in [16] that, under relatively generic conditions, the *pairwise* intercontact times have a probability distribution, which is a mixture between a power-law and an exponential (i.e., power-law head and exponential tail). This has also been confirmed in real-world traces by Karagiannis et al. [17], who analyzed the empirical complementary cumulative distribution functions (CCDFs) of the intercontacts. Assuming such a distribution for the pairwise intercontacts $J_m^{(ij)}$, we will show that their residuals converge to exponential distributions. Based on these findings, we can approximate the random variables J_n as exponentials, which makes the contact process a suitable substrate for Markov models.

[7] Even in reality, this is highly improbable in our sparse networks and downright impossible, when working in continuous time.

Definition 2 (**Contact process**). *Let the CCDF of the random variables $J_m^{(ij)}$ be:*

$$\overline{F_{ij}}(x) = \begin{cases} C_{ij} \cdot x^{-\alpha_{ij}} e^{-\beta_{ij} x}, & \text{for } x \geqslant t_0^{(ij)}, \\ 1, & \text{for } 0 < x < t_0^{(ij)}, \end{cases} \tag{10}$$

where $t_0^{(ij)}$ is the minimum intercontact time for the (i, j) node pair and $C_{ij} = \left(t_0^{(ij)}\right)^{\alpha_{ij}} e^{\beta_{ij} t_0^{(ij)}}$ is a positive *normalization constant. The above function is a combination of a Pareto distribution and an exponential distribution.*

For each node pair (i, j), there uniquely corresponds to any fixed time instant t exactly one m, such that $T_{m-1}^{(ij)} < t < T_m^{(ij)}$. Then, the ***residual time*** on (i, j)'s contact process is the time $K^{(ij)}(t)$ (random variable), from t to the next renewal epoch (next (i, j) contact):

$$K^{(ij)}(t) = T_m^{(ij)} - t. \tag{11}$$

Then, applying our earlier observation that aggregate intercontact times are, in fact, minima of residuals of the pairwise intercontact times, the aggregate intercontacts from Fig. 3 can be expressed as:

$$J_{n-3} = T_{n-3} - T_{n-4} = \min_{1 \leqslant i < j \leqslant 4} \left(K^{(ij)} \left(T_{m-1}^{(1,2)} \right) \right), \tag{12}$$

$$J_{n-2} = T_{n-2} - T_{n-2} = \min_{1 \leqslant i < j \leqslant 4} \left(K^{(ij)} \left(T_{m-1}^{(1,3)} \right) \right), \tag{13}$$

and so on for the following ones. Note that, within the aggregate contact process $(M_n)_{m \in \mathbb{Z}}$, the residual times $K^{(ij)}(t)$ are correlated, as a result of the superposition. However, it has been shown [32] that approximating a superposition of independent renewal processes as a renewal process yields accurate results. This amounts to approximating residual times $K^{(ij)}(t)$ as independent, and the aggregate intercontact times above as minima of independent random variables.

With Definition 2 as a starting point, we will show, in the following, that the residual times $K^{(ij)}(t)$ converge to the exponential distribution of rate β_{ij}.

The mean residual lifetime (MRL) of a non-negative random variable X with CDF denoted F is defined as:

$$\mathrm{MRL}(t) = \mathbb{E}[X - t \mid X > t] = \frac{\int_t^\infty \overline{F}(x)\, \mathrm{d}x}{\overline{F}(t)}. \tag{14}$$

Meilijson showed in [33] that if $\mathrm{MRL}(t)$ converges to a constant A as $t \to \infty$, then the conditional distribution of $(X - t)$ given that $X > t$ (which is none other than the residual of X) converges to the exponential distribution of rate A^{-1}.

Theorem 2 (**Residual Convergence to $\mathtt{Exp}(\beta_{ij})$**). *The pairwise* residual *intercontact times $K^{(ij)}(t)$ for a superposition of heterogeneous pairwise contact processes obeying Definition 2, converge to the exponential distribution of rate β_{ij}, as the pairwise contact processes approach equilibrium*[8] *($t \to \infty$).*

Proof. In our case, the MRL is the expectation of the pairwise residual intercontact times $K^{(ij)}(t)$. Using Eq. (10) for $x \geqslant t_0^{(ij)}$, we obtain the following MRL for the pairwise intercontact time variables $J_m^{(ij)}$:

$$\mathrm{MRL}(t) = \mathbb{E}[K^{(ij)}(t)] = \frac{1}{\beta_{ij}} \cdot \frac{\Gamma\left(1 - \alpha_{ij}, \beta_{ij} t\right)}{\left(\beta_{ij} t\right)^{-\alpha_{ij}} e^{-\beta_{ij} t}}. \tag{15}$$

The second fraction in the above equation is known to converge to 1 as $t \to \infty$. Therefore,

$$\lim_{t \to \infty} \mathrm{MRL}(t) = \frac{1}{\beta_{ij}}, \tag{16}$$

which means that the residual time $K^{(ij)}(t)$ converges to the exponential distribution of rate β_{ij}, as the renewal process reaches equilibrium ($t \to \infty$). This implies that, if our pairwise mobility process $\left(M_m^{(ij)}\right)_{m \in \mathbb{Z}}$ has been going on for a very long time ($t \to \infty$) before we start observing it, the residual time of the (i, j) intercontacts will be exponential with rate β_{ij}. ■

Considering Eqs. (12–13) and recalling that the minimum of independent exponentials is itself exponential, we can approximate the distribution of aggregate intercontact times $(J_n)_{n \in \mathbb{Z}}$ by $\mathtt{Exp}(\sum \beta_{ij})$.

3.5 A Markov Chain Model for Distributed Optimization

Therefore, the pairwise contact probabilities p_c^{ij}, that we used in the definition of our Markov chain **P** from Eq. (9) are now defined as:

$$p_{ij}^c = \frac{\beta_{ij}}{\sum_{1 \leqslant i < j \leqslant N} \beta_{ij}}, \tag{17}$$

and **P** defines indeed a discrete-time homogeneous Markov chain, as we have initially claimed.

To analyze this chain in the rest of the paper, we use Eq. (10) and maximum likelihood estimation (MLE) to estimate the pairwise intercontact time parameters α_{ij} and β_{ij} for all (i, j) pairs. The resulting **contact probability matrix** $\mathbf{P}^c = \{p_{ij}^c\}$ entirely describes any given mobility scenario with **heterogeneous node mobility**. Note that, since we are using the embedded Markov chain $(X_n)_{\mathbb{N}_0}$, all the time related quantities that we calculate will be expressed in *event time* (measured in number of contact events or "contact ticks") as opposed to *standard time* (wall-clock time). Because our contact process is stationary and ergodic, it is easy to revert from event time to wall time, using Wald's equation as shown below.

In a somewhat different context focusing on distributed estimation, the authors of [35] assume exponential clock ticks and use the law of large numbers to show that the exact time of the n-th tick is highly concentrated around its average. We argue here that a slightly more generic statement can be made based on renewal theory and *Wald's equation* [36].

[8] An equilibrium renewal process, in the sense of [34] (page 28), is an ordinary renewal process started in the remote past at $t_s \to -\infty$, with respect to $t_o = 0$, when the observation of the process starts.

Applying the DTN-Meteo Framework in Practice

(I) **Mobility.** Extract all pairwise contact rates $\beta_{ij} > 0$ from your mobility scenario. To do this, use the assumption that pairwise intercontacts are distributed as a power law with exponential cuttoff, as in Eq. (10) and a parameter estimation method (e.g., Maximum Likelihood Estimation – MLE).

(II) **Problem.** Identify your problem's state space. Many problems can be described with binary node states, which results in a problem state space Ω formed of binary vectors of length N (size of the network). For example, in epidemic routing, Ω consists of all possible binary vectors of size N.

- Add any constraints your problem may have, to help you reduce the state space size.

(III) **Optimization Algorithm.**

- Based on your state space Ω from Step (II) and the contact probabilities from Step (I), create **M**, an $|\Omega| \times |\Omega|$ probability matrix, describing which contact(s) are needed for which state transition, as in Fig. 2.
- Using the utility function of your protocol, calculate the utility of each state of Ω (your problem's state space defined at Step (II)). Based on these utility values, create **A**, another $|\Omega| \times |\Omega|$ matrix containing acceptance probabilities $A_{\mathbf{xy}}$ (either binary or in $[0,1]$).

(IV) **Markov chain.** Take the element-wise (Hadamard) product of **M** and **A**, to obtain the transition matrix of your Markov chain $\mathbf{P} = \mathbf{M} \circ \mathbf{A}$.

(V) **Analysis.** Using standard Markov chain analysis, obtain performance results for your triple combination of: < mobility – problem – algorithm >.

Figure 4: *Summary of the necessary steps when building a DTN-Meteo model.*

Lemma 3. *Let the expected time between consecutive contact events be $\mathbb{E}[J_n]$. Let further T_d denote the delay or convergence time (in "contact ticks") of a given process over an opportunistic contact graph. Then, the expected delay of this process is equal to:*

$$\mathbb{E}[J_n] \cdot \mathbb{E}[T_d]. \tag{18}$$

Proof. As shown earlier in this section, the times of consecutive contact events can be approximated renewals. If time is counted at renewal times (contact times), it is easy to see that any delay quantity derived in "contact ticks", is a *stopping time*. We can then apply Wald's equation [36] to get Eq. (18). ■

Summarizing from Section 3, Fig. 4 shows the sketched recipe by which anyone can transform a problem of our defined class into a Markov chain.

4 Analyzing DTN Optimization Algorithms

In this section, we show some examples of how to use the DTN-Meteo model to quantify the worst-case and expected performance of an algorithm – Step (V) of the DTN-Meteo recipe from Fig. 4. In particular, we will: (i) show how to easily check the worst-case behavior of a greedy algorithm (greedy content placement); (ii) describe how to calculate the expected performance of greedy algorithms for two DTN problems (routing and content placement), from standard Markov chain analysis; and (iii) present an example of randomized local search algorithm (based on a Markov Chain Monte Carlo (MCMC) method) applied to content placement and derive its expected performance.

4.1 DTN-Meteo for Greedy Algorithms: Worst Case

A crucial point that arises with the use of greedy distributed optimization in DTNs is the (in)ability of such algorithms to efficiently navigate the solution space. Specifically, *what properties of the mobility model or the utility function make simple utility ascent algorithms applicable?* In this section, we illustrate the value of DTN-Meteo in answering this questions, for the example of greedy content placement. To our best knowledge, this is the first *worst-case* analysis for a DTN algorithm in generic mobility scenarios.

Recall from Section 3.1, our working example of greedy con-

	MIT	INFO	ETH	ARMA	SFTAXI	TVCM
Scale and context	92 campus students & staff	41 conference attendees	20 lab students & staff	152 people	536 taxis	24/104 nodes, 2/4 disjoint communities
Period	9 months	3 days	5 days	9 hours	1 month	11 days
Scanning Interval	300s (Bluetooth)	120s (Bluetooth)	0.5s (Ad Hoc WiFi)	30s (GPS)	30s (GPS)	N/A
# Contacts total	81 961	22 459	22 968	12 875	1 339 274	1 000 000

Table 3: *Mobility traces characteristics.*

tent placement. The goal 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 are 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. We consider a simple algorithm, where 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 a desired (time-invariant) utility $U(\mathbf{x})$.

We will prove necessary and sufficient conditions for the correctness of this greedy algorithm for the content placement problem, that is guaranteed discovery of the optimal solution $\mathbf{x}^{\star}$. Assuming an arbitrary additive (or otherwise decomposable) utility function, $U(\mathbf{x}) = \sum_{i=1}^{N} x_i u(i)$ (with $u : \mathcal{N} \mapsto \mathbb{R}$), this solution amounts to "pushing" a replica to each of the L highest utility nodes. We denote this set of nodes as $\mathcal{L}^{\star}$. By definition, the network state $\mathbf{x}^{\star}$ is *absorbing* in the Markov chain of our problem. Depending on the mobility scenario and the chosen utility function, there may be more absorbing states (i.e., local maxima of u). In that case, the algorithm is not guaranteed to converge to the optimal solution from every initial copy assignment, and thus is not correct.

Thm. 4 derives necessary and sufficient conditions on the *contact probability matrix* $\mathbf{P}^c$ and *utility function* u, for the correctness of the greedy content placement algorithm. This is shown to require the existence of an increasing utility path from any node in $\mathcal{N} \setminus \mathcal{L}^{\star}$ (nodes outside the L highest utility ones) to any node in $\mathcal{L}^{\star}$. The proof can be found in the Appendix.

Theorem 4 (**Correctness of greedy content placement**). *For all source nodes and all initial copy allocations and for an arbitrary decomposable utility function $U : \Omega \mapsto \mathbb{R}$,* greedy content placement *is correct if and only if, for all $i \in \mathcal{N} \setminus \mathcal{L}^{\star}$, there exist at least L nodes $j_1, \ldots, j_L \in \mathcal{N}$ with $u(i) < u(j_{(.)})$, such that $p^c_{ij_{(.)}} > 0$ (i.e., $\beta_{ij_{(.)}} > 0$).*

Put differently, given a DTN distributed optimization problem (mapped into a utility function), and a mobility scenario (captured in a contact probability or rate matrix), one can convert (using DTN-Meteo and the example of Thm. 4 above) the often hard task of deciding whether a simple greedy algorithm would suffice, into the often easier task of checking a matrix (the contact matrix) for enough non-zero entries.

Having derived conditions for the correctness of greedy content placement, we investigate below whether and when these conditions actually hold in realistic mobility scenarios. To cover a broad range, we use five real contact traces and two synthetic mobility trace for validation. Their characteristics are summarized in Table 3: (i) the *Reality Mining* trace (MIT) [10], (ii) the Infocom 2005 trace (INFO) [37], (iii) the ETH trace [11], (iv) a Swiss military trace from an outdoor training scenario (ARMA), (v) the San Francisco taxis trace (SFTAXI) and (vi) a synthetic scenario created with a recent mobility model (TVCM) [38].

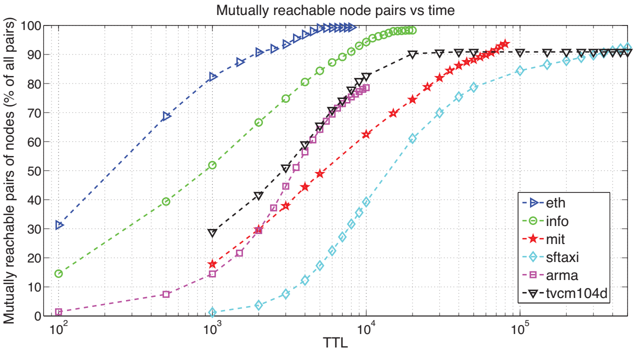
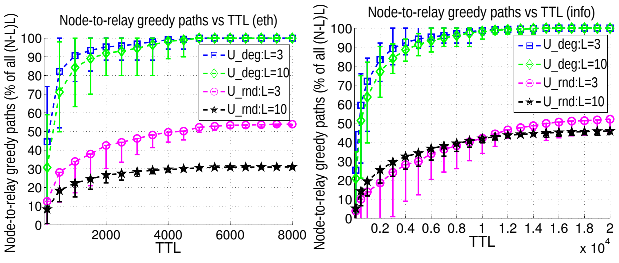


Figure 5: *Multihop greedy paths in traces*



(a) Utility Ascent paths (ETH) (b) Utility Ascent paths (Infocom)

Figure 6: *Comparison of several utilities*

Figure 5 shows the percentage of node pairs that are mutually reachable by a utility ascent path as a function of TTL. While for the whole trace duration, over 90% of the node pairs are greedily reachable (note that the whole trace duration exceeds one week worth of contacts for some of the traces), for smaller, more realistic TTL values, many node pairs do not have paths for the greedy algorithm to use. This is due either the *nodes being unreachable* or to existence of *absorbing local maxima*. This can be determined by comparison to the number of mutually reachable pairs when allowing non-greedy paths. In ETH, the number of paths does not increase significantly, making this scenario suitable for greedy solutions. In contrast, in the other scenarios, the improvement ranges from double to fivefold, making greedy solutions waste potential.

In addition to the contact pattern of a mobility scenario, the conditions of Thm. 4 (and thus algorithm correctness) may be affected by the following factors: (a) the value of L, (b) the choice of the node utility function u. In Figure 6, we inspect more closely the effect of L and the relation between a mobility-dependent node utility (node degree) and a random utility, generated independently for each node, from the uniform distribution on $(0, 1)$.

Two observations ensue from Figure 6. First and foremost, there do not always exist utility ascent paths leading to optimal solutions. This means local maxima are present even for simple utility functions. Larger L may exacerbate this problem, as in Figure 6(a). Second, the correlation or lack thereof between a node's utility value and that node's mobility characteristics considerably affects the navigability properties of the contact graph. This stresses the need to choose utility functions carefully.

Summarizing, we have used the derived correctness conditions, to analyze the worst case behavior of an example *greedy content placement* algorithm in a variety of real and synthetic mobility scenarios. We conclude that this algorithm is relatively fragile in the face of parameters like TTL and the choice of utility function. As mentioned earlier, our model can adapt to study other algorithms and optimization problems in a similar way. When an algorithm's worst case behavior is important, this is an essential tool for its analysis.

4.2 DTN-Meteo for Greedy Algorithms

In this section, we demonstrate how to calculate the expected performance of greedy algorithms for two example DTN problems: routing and content placement. We use the theory of absorbing Markov chains [39], 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 and *probability and delay* of optimally placing content. We make no further assumption on utility, except the existing ones (namely, utility is time-invariant and can be locally calculated by any node).

Maximum utility (optimum) network states are, by definition, *absorbing* states in the DTN-Meteo Markov chain from Eq. (9). 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 of nodes. These correspond to local maxima and are also absorbing states in $\mathbf{P}$. We use the following notation for the two sets of states:

$\mathcal{N}$, N	the network (set of nodes), resp. its size
L	maximum allowed replication (number of copies)
Ω	network state space (set of binary vectors $\mathbf{x}$)
$\mathbf{x}^\star$	optimum network state(s) (element(s) of Ω)
$\mathbf{P}$	Markov chain under analysis (transition matrix)
$p_{\mathbf{x}}^{(0)}$	initial probability for state $\mathbf{x}$
p_d, T_d	convergence probability, resp. delay
$p_d(\mathbf{x}^\star)$	prob. of convergence to one of several opt. states $\mathbf{x}^\star$
$T_d(\mathbf{x}^\star)$	delay of convergence to one of several opt. states $\mathbf{x}^\star$
$\mathbf{Q}$	transient part of $\mathbf{P}$ (greedy algo's)
$\mathbf{R}_1$, $\mathbf{R}_2$	local/global opt. absorbing parts of $\mathbf{P}$ (greedy algo's)
$\mathbf{N} = \{n_{\mathbf{xy}}\}$	fundamental matrix of $\mathbf{P}$ (greedy algo's)
$b_{\mathbf{xx}^\star}$, $\tau_{\mathbf{xx}^\star}$	prob., resp. delay of convergence from $\mathbf{x}$ to $\mathbf{x}^\star$
$\mathbf{Z} = \{z_{\mathbf{xy}}\}$	fundamental matrix of $\mathbf{P}$ (randomized algo's)
π	stationary distribution of $\mathbf{P}$ (randomized algo's)
$m_{\mathbf{xy}}$	mean first passage time from $\mathbf{x}$ to $\mathbf{y}$ (rand. algo's)

Table 4: *Important notation in Section 4.*

$$\mathcal{GM} = \{\mathbf{x}^\star \in \Omega \mid U(\mathbf{x}^\star) \geqslant U(\mathbf{y}),\ \forall \mathbf{y} \in \Omega\} \qquad \textbf{(global maxima)},$$
$$\mathcal{LM} = \{\mathbf{x} \in \Omega \setminus \mathcal{GM} \mid p_{\mathbf{xy}} = 0,\ \forall \mathbf{y} \in \Omega\} \qquad \textbf{(local maxima)}.$$

For example, in Spray and Wait routing, global maxima 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. Local maxima states $\mathcal{LM}$ arise when the probabilities $p_{\mathbf{xy}}$ for transitions leaving the $\mathcal{LM}$ state $\mathbf{x}$ is zero, either because:

- $\delta(\mathbf{x}, \mathbf{y}) > 2$ – the states are not adjacent, or because
- $p_{ij}^c = 0$ – the required nodes never meet, or because
- $U(\mathbf{y}) < U(\mathbf{x})$ – neighboring states $\mathbf{y}$ have lower utilities.

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

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 $\mathcal{GM}$:

$$\mathbf{P} = \begin{array}{c} \\ \\ \end{array} \begin{array}{ccc|l} \text{TR} & \mathcal{LM} & \mathcal{GM} & \\ \mathbf{Q} & \mathbf{R}_1 & \mathbf{R}_2 & \text{TR} \\ \mathbf{0} & \mathbf{I}_1 & \mathbf{0} & \mathcal{LM} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathcal{GM} \end{array} \tag{19}$$

Let $|\mathcal{GM}| = r_2$, $|\mathcal{LM}| = r_1$ and $|\text{TR}| = t$. 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:

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \cdots \tag{20}$$

The last equality is easy to derive (see [39], 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**, $p_{\mathbf{x}}^{(0)}$, of the DTN-Meteo chain $(\mathbf{X}_n)_{n \in \mathbb{N}_0}$. For illustrative purposes, let us assume that all N nodes are equally likely content/message sources. However, we cannot directly use this as the initial probability distribution of our chain, since the model does not include the algorithms' replication phase (as explained in Section 3.1). Therefore, we derive the initial probability distribution for all states $\mathbf{x} \in \Omega$ (with exactly L copies at L different nodes) as:

$$p_{\mathbf{x}}^{(0)} = \frac{1}{N} \sum_{i \mid x_i = 1} \mathbb{P}[\mathbf{X}_0 = \mathbf{x} \mid \text{source is } i]. \tag{21}$$

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:

$$p_{\mathbf{x}}^{(0)} = \frac{1}{N} \sum_{i \mid x_i = 1} \frac{\prod_{j=1}^{N} x_j p_{ij}^c}{\sum\limits_{\mathbf{y} \mid y_i = 1} \left(\prod_{j=1}^{N} y_j p_{ij}^c \right)}. \tag{22}$$

From Absorption Analysis to Practical Metrics. Based on the fundamental matrix and the initial probability distribution, we can now easily 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 5 (**Success Probability**). *The end-to-end* delivery probability *for a greedy routing algorithm modeled by chain* $\mathbf{P}$ *starting from any initial source(s) with equal probability, is*

$$p_d = \sum_{\mathbf{x}^\star \in \mathcal{GM}} p_d(\mathbf{x}^\star) = \sum_{\mathbf{x}^\star \in \mathcal{GM}} \left(\sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{xx}^\star} \right), \tag{23}$$

where $b_{\mathbf{xx}^\star}$ 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 $\mathbf{x}$. The success probability *of greedy content placement finding the best set of L relays, starting from any initial source(s) with equal probability obeys the same relation.*

Proof. Starting from transient state $\mathbf{x}$, the process may be captured in the optimal state $\mathbf{x}^\star$, in one or more steps. The probability of capture on a single step is $p_{\mathbf{xx}^\star}$. If this does not happen, the process may move either to an absorbing state in $\mathcal{LM}$ (in which case it is impossible to reach $\mathbf{x}^\star$), or to a transient state $\mathbf{y}$. In the latter case, there is probability $b_{\mathbf{yx}^\star}$ of being captured in the optimal state. Hence we have:

$$b_{\mathbf{xx}^\star} = p_{\mathbf{xx}^\star} + \sum_{\mathbf{y} \in TR} p_{\mathbf{xy}} \cdot b_{\mathbf{yx}^\star}, \tag{24}$$

which can be written in matrix form, for all $\mathbf{x}^\star$, as $\mathbf{B}^\star = \mathbf{R}_2 + \mathbf{QB}^\star$. Thus $\mathbf{B}^\star = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R}_2 = \mathbf{NR}_2$. $\mathbf{B}^\star$ is the matrix of delivery ratios starting from each of the t transient states towards each of the r_2 optimal states. The delivery ratio starting from any source uniformly toward a given state $\mathbf{x}^\star$ is:

$$p_d(\mathbf{x}^\star) = p_{\mathbf{x}_1}^{(0)} \cdot b_{\mathbf{x}_1\mathbf{x}^\star} + \cdots + p_{\mathbf{x}_t}^{(0)} \cdot b_{\mathbf{x}_t\mathbf{x}^\star} = \sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{xx}^\star}. \tag{25}$$

Finally, since absorption events at the different optimal states are mutually exclusive, the delivery ratio starting from any initial state uniformly, toward any optimal state is the sum of Eq. (25) over all $\mathbf{x}^\star$. ∎

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 $p_d(\mathbf{x}^\star) = \sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} b_{\mathbf{xx}^\star}$ defined in Eq. (23) above.

Theorem 6 (**Expected Delay**). *The expected end-to-end* delivery delay *for a greedy routing algorithm modeled by chain* $\mathbf{P}$*, starting from any source with equal probability,* given that it does not get absorbed in any local maximum *is:*

$$\mathbb{E}[T_d] = \sum_{\mathbf{x}^\star \in \mathcal{GM}} \frac{p_d(\mathbf{x}^\star)}{p_d} \mathbb{E}[T_d(\mathbf{x}^\star)] = \sum_{\mathbf{x}^\star \in \mathcal{GM}} \frac{p_d(\mathbf{x}^\star)}{p_d} \left(\sum_{\mathbf{x} \in TR} p_{\mathbf{x}}^{(0)} \tau_{\mathbf{xx}^\star} \right), \tag{26}$$

where $\tau_{\mathbf{xx}^\star}$ 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 $\mathbf{x}$. The expected convergence delay *for greedy 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. We compute these as follows. Let $\mathbf{a}$ be the statement "$\mathbf{X}_n$ is absorbed in state $\mathbf{x}^\star$". Then, if $\mathbf{x}$ is a transient state, the transition probabilities $\hat{p}_{\mathbf{xy}}$ for $\mathbf{Y}_n$ are:

$$\hat{p}_{\mathbf{xy}} = \mathbb{P}[\mathbf{X}_{n+1} = \mathbf{y} \mid \mathbf{a} \wedge \mathbf{X}_n = \mathbf{x}] = \frac{\mathbb{P}[\mathbf{X}_{n+1} = \mathbf{y} \wedge \mathbf{a} \mid \mathbf{X}_n = \mathbf{x}]}{\mathbb{P}[\mathbf{a} \mid \mathbf{X}_n = \mathbf{x}]}$$
$$= \frac{\mathbb{P}[\mathbf{a} \mid \mathbf{X}_{n+1} = \mathbf{y}] \cdot \mathbb{P}[\mathbf{X}_{n+1} = \mathbf{y} \mid \mathbf{X}_n = \mathbf{x}]}{\mathbb{P}[\mathbf{a} \mid \mathbf{X}_n = \mathbf{x}]} = \frac{b_{\mathbf{yx}^\star} p_{\mathbf{xy}}}{b_{\mathbf{xx}^\star}}.$$

The canonical form for $\hat{\mathbf{P}}$, the transition matrix of $\mathbf{Y}_n$, is obtained as follows. The matrix $\hat{\mathbf{R}}$ (corresponding to the single absorbing state in the new chain) is a column vector with $\hat{\mathbf{R}} = \{\frac{p_{\mathbf{xx}^\star}}{b_{\mathbf{xx}^\star}}\}$. Let $\mathbf{D}$ be a diagonal matrix with diagonal entries $b_{\mathbf{xx}^\star}$, for $\mathbf{x}$ transient. Then $\hat{\mathbf{Q}} = \mathbf{D}^{-1}\mathbf{QD}$ and thus, $\hat{\mathbf{N}} = \mathbf{D}^{-1}\mathbf{ND}$. $\hat{\mathbf{B}}^\star$ is now a t-sized column vector of ones. Recall that a fundamental matrix' entry $\hat{n}_{\mathbf{xy}}$ is the expected number of steps, the chain is in state $\mathbf{y}$, starting from $\mathbf{x}$, before getting absorbed. Since $\mathbf{x}^\star$ is the only absorbing state in the new chain, $\boldsymbol{\tau}_{\mathbf{xx}^\star}$ is simply a row sum of the new fundamental matrix $\hat{\mathbf{N}}$.

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. (26) is obtained. ■

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 Section 5, we will compare our predictions obtained from Thms. 5 and 6 to results obtained from simulations, for both problems.

4.3 DTN-Meteo for Randomized Local Search

So far, we have only considered greedy algorithms, for which the acceptance probability $A_{\mathbf{xy}}$ from Section 3.3 is always 0 or 1: a better solution is always chosen and a worse one is never chosen. This feature makes the Markov chain (Eq. (9)) for the problem absorbing and may also create local maxima, where greedy algorithms may get blocked.

To this end, randomized local search algorithms have been proposed. While a randomized local search still deterministically accepts a better (higher utility) solution, it may also move to a lower utility solution with a probability $A_{\mathbf{xy}} > 0$. These probabilities (of moving to lower utility states) are calibrated so as to provably converge to the/an optimal solution. One example of a class of such randomized local search algorithms are Markov Chain Monte Carlo (MCMC) methods [5]. While MCMC methods are often used to simulate and sample complex (and non-invertible) functions, they also provide a powerful tool for stochastic optimization. The most commonly used MCMC optimization algorithm and also our choice for this example is Metropolis-Hastings.

As transitions between two adjacent states are now possible in both directions, the DTN-Meteo Markov chain (Eq. (9)) for an MCMC algorithm is ***not*** *absorbing*. Moreover, since we have established in Section 3.4 that mobility is stationary and ergodic, the DTN-Meteo chain is, as well stationary and ergodic. It has a unique stationary distribution, which depends on both the contact probabilities (mobility) and on the acceptance probabilities (algorithm). The latter can be chosen by the algorithm designer.

Since we are interested in finding a/the global maximum solution, we naturally should try to maximize the stationary probabilities of good (high utility) solutions. One example of such a distribution is the Gibbs distribution [5]:

$$\pi(\mathbf{x}) = \frac{\exp\left(U(\mathbf{x}) \cdot T^{-1}\right)}{\sum_{\mathbf{y}\in\Omega} \exp\left(U(\mathbf{y}) \cdot T^{-1}\right)}, \tag{27}$$

where T is an algorithm parameter, the temperature. When T is small, the distribution is concentrated around the large values of $U(\mathbf{x})$ and thus, the algorithm converges to a "good" solution with high probability.

Using the Metropolis-Hastings acceptance probability formula [5] and the above-defined stationary distribution, we obtain the following:

$$A_{\mathbf{xy}} = \min\left(1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}\right) = \min\left(1, \exp\left(\frac{U(\mathbf{y}) - U(\mathbf{x})}{T}\right)\right). \tag{28}$$

A Metropolis-Hastings algorithm using this acceptance probability has been proved to eventually converge to an/the optimum solution [5].

To sum up, assuming that node mobility creates a connected contact graph (i.e. for all nodes i, there exists at least one node j such that their meeting probability $p_{ij}^c > 0$) and recalling that our mobility process is ergodic, the Markov chain of our randomized local search algorithm will be irreducible and aperiodic, with $\pi(\mathbf{x})$ as its unique stationary distribution. Since we have chosen π (see Eq. (27)) to be highly concentrated around the states $\mathbf{x}^\star \in \mathcal{GM}$ with the largest utility, the Markov chain, and hence, the algorithm, will eventually converge to those states with high probability ($\mathbb{P}[\text{convergence}] \to 1$).

Remark: Sometimes, the temperature parameter T is varied *during* the algorithm's operation, in order to speed up convergence. This is known as *simulated annealing*, when T starts relatively high and gradually "cools down". However, it is beyond the scope of our paper, as we do not focus on "tweaking" the algorithm itself.

Convergence Analysis for Randomized Local Search. Since the Markov chain is now irreducible, hitting an optimal state does not guarantee that the algorithm stays there forever. Convergence to one of the optimum states $\mathbf{x}^\star \in \mathcal{GM}$ is now asymptotic (as opposed to exact, for the absorbing Markov chain in the previous section), therefore, we will calculate the convergence time as the *first hitting/passage time(s)* to the Markov chain state(s) corresponding to (one of) the optimal state(s) $\mathbf{x}^\star \in \mathcal{GM}$, starting from any other state $\mathbf{y} \in \Omega \setminus \mathcal{GM}$.

The first passage time may still correspond exactly to a performance metric of interest, for example end-to-end delay in routing (the algorithm terminates as soon as the destination node has been reached). However, in other cases, the first passage time will be a lower bound, for example, for the delay of reaching the optimum content storers in content placement. If the algorithm is designed correctly (according to the remark above), this bound can become tight.

Several methods are available for the analysis of stationary and ergodic Markov chains, from first step analysis to the electric network analogy, and including spectral properties.

In the following, we will use an approach similar to the one in Section 4.2, based on an equivalent of the *fundamental matrix for ergodic chains*, derived from first step analysis:

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{\Pi})^{-1}, \tag{29}$$

where $\mathbf{\Pi}$ is a matrix with each of its rows being the *stationary probability vector* π (Eq. (27)) for transition matrix $\mathbf{P}$.

We will derive a lower bound for the expected convergence delay of randomized greedy content placement and the exact expected end-to-end delay of randomized greedy routing, in an equivalent way to Theorem 6. We use the new fundamental matrix $\mathbf{Z}$ and the initial probability vector derived in Eq. (22).

Recall from Section 4.2 that, for each DTN problem's state space, there may be more than one optimal (maximum utility) state $\mathbf{x}^\star$. For example, in fixed-replication routing, all of the $\binom{N-1}{L-1}$ states in which one of the L copies is at the destination node d are equally good and will form the set $\mathcal{GM}$. This is less likely in our content placement example, where all of the L relays defining a network state $\mathbf{x}$ count towards that state's final utility $U(\mathbf{x})$.

When several optimal states are present, each of their first passage times must be combined, in order to obtain the final performance metric of the randomized greedy algorithm (e.g., end-to-end delivery delay in routing or convergence delay in content placement). Below, we will first show how to derive the first passage time for a unique optimal state (Theorem 7) and then how to combine the results to obtain the final performance metrics (Corollary 8).

Theorem 7 (**First Passage Time**). *Assuming a* ***unique optimal state*** $\mathbf{x}^\star$*, a lower bound for the expected* convergence delay *of a randomized algorithm modeled by our Markov chain* $\mathbf{P}$*, starting from any source with equal probability is:*

$$\mathbb{E}[T_d(\mathbf{x}^\star)] = \sum_{\mathbf{y}\neq\mathbf{x}^\star} p^{(0)}_{\mathbf{y}} \frac{z_{\mathbf{x}^\star\mathbf{x}^\star} - z_{\mathbf{y}\mathbf{x}^\star}}{\pi(\mathbf{x}^\star)}, \tag{30}$$

where $\mathbf{Z} = \{z_{\mathbf{xy}}\}$ *is the fundamental matrix for ergodic chains and* $|\Omega|$ *is the size of the solution space.*

Proof. In [39] (p. 78), the authors prove that the matrix $\mathbf{M}$, containing the mean first passage times from any state $\mathbf{x} \in \Omega$ to any other state $\mathbf{y} \in \Omega$ is given by: $\mathbf{M} = (\mathbf{I} - \mathbf{Z} + \mathbf{E}\mathbf{Z}_{\mathbf{dg}})\mathbf{D}$, where $\mathbf{E}$ is a square matrix with all entries 1, $\mathbf{Z}_{\mathbf{dg}}$ agrees with $\mathbf{Z}$ on the main diagonal and is 0 elsewhere, and $\mathbf{D}$ is a diagonal matrix with diagonal elements $d_{\mathbf{yy}} = (\pi(\mathbf{y}))^{-1}$.

From a given state $\mathbf{y}$ to the unique optimal configuration $\mathbf{x}^\star$, the mean first passage is element $m_{\mathbf{yx}^\star}$ of $\mathbf{M}$, which from above can be written as:

$$m_{\mathbf{yx}^\star} = \frac{z_{\mathbf{x}^\star\mathbf{x}^\star} - z_{\mathbf{y}\mathbf{x}^\star}}{\pi(\mathbf{x}^\star)}. \tag{31}$$

Using the initial probability distribution in Eq. (22), we calculate the weighted average of Eq. (31) over all non-optimal states $\mathbf{y}$ to obtain the expected first passage time starting from any source node with equal probability to the unique optimal state $\mathbf{x}^\star$ as shown in Eq. (30). ■

To address the case when the several optimal states of equal utility values are present, we must also obtain the probabilities for each of those optimum states to be reached first (i.e., before any other optimum state), in addition to the first passage times. Then, using the first passage times calculated as above, we can derive, e.g., the expected end-to-end delay of randomized greedy routing.

The probability for an optimum state $\mathbf{x}^\star$ to be reached before any other optimum state can be easily calculated by setting all states $\mathbf{x}^\star \in \mathcal{GM}$ as absorbing in our Markov chain of the randomized greedy algorithm and using Theorem 5 from the previous section to calculate each $p_d(\mathbf{x}^\star)$. (Note that, in this case, it will not make sense to add up the probabilities, like in Theorem 5. They will simply sum to 1, as there will be no other absorbing states in the chain.)

Corollary 8 (**Expected Delay**). *The exact expected end-to-end* delivery delay *for a randomized routing algorithm modeled by chain* $\mathbf{P}$*, starting from any source with equal probability, is:*

$$\mathbb{E}[T_d] = \sum_{\mathbf{x}^\star\in\mathcal{GM}} p_d(\mathbf{x}^\star)\mathbb{E}[T_d(\mathbf{x}^\star)] \tag{32}$$

where $\mathbb{E}[T_d(\mathbf{x}^\star)]$ *is given in Eq. (30). A lower bound for the expected* convergence delay *for randomized content placement to find the best set of L relays, starting from any initial source with equal probability obeys the same relation.*

Summarizing Section 4, we have shown detailed examples of how to use the DTN-Meteo model and Markovian analysis, to quantify the worst-case and expected performance of two types of algorithms (purely greedy and randomized greedy), for two important DTN problems: routing and content placement. We have used generic utility functions, only requiring that the functions be time-invariant and that they can be locally calculated at each node.

In the next section, we validate the accuracy of the results provided by DTN-Meteo against simulation results from both real world traces and synthetic mobility models. We use state-of-the-art routing and content placement algorithms, whereof the utility functions obey our requirements.

5 Applications to Communication Algorithms

In this section, we apply DTN-Meteo to the state-of-the-art routing algorithms – SimBet [12] and BubbleRap [18], and to content placement, all of them using the first two replication strategies in Table 2: single-copy and fixed budget L. None of these utility-based algorithms can be modeled by existing tools.

5.1 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 [40].

5.1.1 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 highest number of (unique) nodes per unit time [41]. Using the graph $\mathbf{W}$, this number amounts to 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:

$$U(\mathbf{x}) = \sum_{i=1}^{N} x_i d_i. \tag{33}$$

5.1.2 SimBet

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[9]:

$$u_d(i) = \alpha \cdot Sim_d(i) + \beta \cdot Bet(i) \tag{34}$$

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

SimBet was first published as a single-copy utility-based protocol. It was later enhanced [12] with the option of using a fixed number of copies L.

5.1.3 BubbleRap

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 [42] on the same graph to detect communities. Thus, in BubbleRap the utility of node i, for a destination node d is:

$$u_d(i) = \mathbb{1}\{i \in C_d\} \cdot LBet(i) + \overline{\mathbb{1}\{i \in C_d\}} \cdot GBet(i), \tag{35}$$

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: $U_d(\mathbf{x}) = \sum_{i=1}^{N} x_i u_d(i)$.

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. (9) is now entirely defined[10] and we can apply the convergence analysis from Section 4 to both the purely greedy and to the randomized greedy versions of 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 Section 4.1.

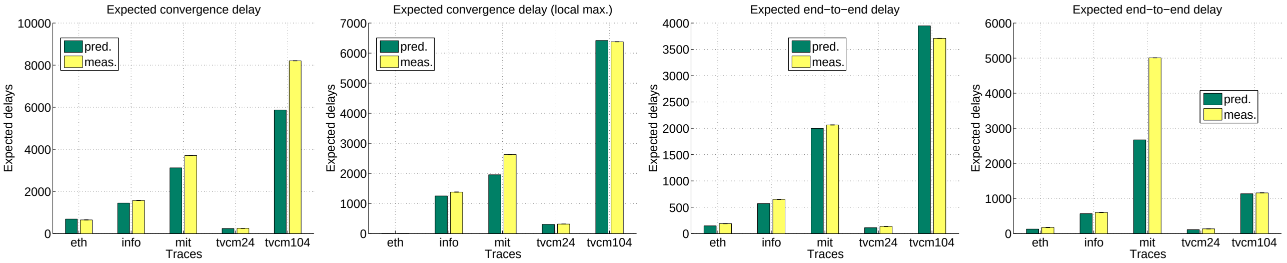
5.2 Measuring the Accuracy of DTN-Meteo

To evaluate the prediction accuracy of DTN-Meteo, we use most of the traces summarized in Table 3.

In simulations (trace replays), we measure absorption quantities as follows. For every node i in the network, we set i as a source of content/messages and do one simulation run. For the two routing algorithms, we do one run per source–destination node pair (using only a subset of randomly chosen destinations, in larger scenarios). In each run, the source

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

[10] 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.



(a) Absorption by optimum (b) Absorption by local max. (c) SimBet (d) BubbleRap

Figure 7: *Predictions and measurements for **purely greedy** content placement and routing*

node i generates pieces of content/messages using a Poisson process. This ensures, via the PASTA property [43], that we do not introduce any sampling bias. The content/message generation process produces a sample of at least 1 000 observations per source node/run for shorter traces and up to 15 000 observations per source node/run for longer ones. For all measured delays, we compute the 95th percentile using the normal distribution.

To obtain the source (and destination) independent metrics, we then average over all source (and destination) nodes.

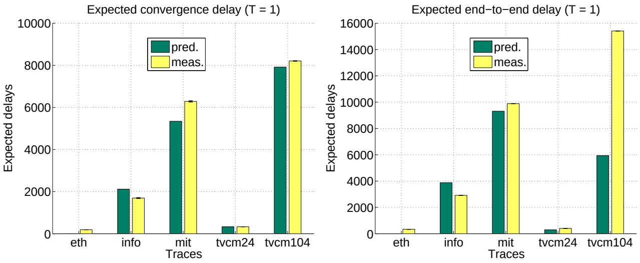
	L	Optimum		Best local max.	
		pred.	meas.	pred.	meas.
ETH	5	1.00	1.00	N/A	N/A
Infocom	3	0.34	0.36	0.18	0.20
MIT	2	0.39	0.38	0.42	0.42
TVCM24	4	0.10	0.10	0.38	0.38
TVCM104	2	0.60	0.48	0.36	0.44

Table 5: *Absorption probabilities*

Tab. 5 shows the measured vs. predicted (Thm. 5) success probabilities of greedy 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 greedy routing algorithms show similar accuracy; we omit them due to space limitations.

The first two plots in Fig. 7 compare the measured and predicted (Thm. 6) values of the convergence delay of greedy content placement, averaged over all initial states (L values as in Tab. 5). The theoretical results coincide once again surprisingly well with the measured delays, both for absorption by the optimum state – Fig. 7(a), and for absorption by a local maximum – Fig. 7(b). In Fig. 7(b), the ETH trace does not have any local maxima with our utility.

Figs. 7(c) and 7(d) compare the measured and predicted (Thm. 6) values of the end-to-end delivery delay of greedy SimBet and BubbleRap routing, averaged over all initial states (L values as in Tab. 5). 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.



(a) Rand. Content placement ($L = 1$) (b) Randomized SimBet ($L = 2$)

Figure 8: *Predictions and meas. for **randomized** content placement and routing*

Finally, in Fig. 8, we explore the randomized versions of content placement and SimBet routing. As explained in Section 4.3, randomization helps navigate around local maxima, at the cost of convergence time. Figs. 8(a) and 8(b) show the predicted (Thm. 8) versus measured values for randomized content placement and respectively, randomized SimBet routing. The predictions generally show similar accuracy as their greedy counterparts, with the exception of randomized SimBet routing on the ETH trace and the TVCM104 scenario. These suffer from a similar deficiency as in Fig. 7(d) for BubbleRap routing on the MIT trace.

Summarizing, we have used DTN-Meteo to model greedy and randomized versions of three DTN communication algorithms and obtained 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.*

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 6 for SimBet on ETH.

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

Table 6: *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.

6 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. (20), 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 5, 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 → 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.

6.1 The Independence of Copies

In multi-copy DTN algorithms, copies may be dependent in two ways: First, as implied in Section 3.3, a state's utility $U(\mathbf{x})$, may not be decomposable in node utilities $u(i)$. 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 7. 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 7: *Copy dependency vs. L to N ratio*

6.2 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 9 (**Routing Approx.**). *The expected end-to-end delay for L-copy routing from starting nodes $s_1, \dots, s_L$ to node d is:*

$$\tau_{rt} = \mathbb{E}[T_{rt}] = \mathbb{E}\left[\min(T_{s_1 d}, \dots, T_{s_L d})\right]. \tag{36}$$

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

Proof. We are performing L-copy routing from source node s_1 to destination node d and spraying brings the L copies to nodes $s_1, s_2, \dots, s_L$. The copies evolve indepedently and are absorbed at d after, respectively, $T_{s_1 d}, \dots, T_{s_L d}$. These are absorption delays on the single-copy Markov chain, whose expectations $\tau_{s_1 d}, \dots, \tau_{s_L d}$ we can calculate from Thm. 6. 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 these L random variables as in Eq. (36). ■

Lemma 10 (**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:*

$$\tau_{cp} = \mathbb{E}[T_{cp}] = \mathbb{E}\left[\sum_{i=1}^{L} T_i^{(L-i+1)}\right]. \tag{37}$$

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.

Proof. We are performing content placement with L replicas starting from a random initial node with equal probability. The copies must arrive at the L highest utility nodes, denoted $n_1, \ldots, n_L$ with $u(n_i) > u(n_{i+1})$. In the exact $\binom{N}{L}$-state Markov chain, the delay of the algorithm corresponds to the time until all these nodes are occupied by a copy. Consider our approximation with the N-state Markov chain and assume that the copies move in parallel over the same chain (corresponding to the single-copy problem), without any interaction *except in the final states* (i.e., no more than one copy at any $n_i, 1 \leqslant i \leqslant L$).

- $T_1^{(L)}$ is the expected delay until *any* of the L copies is absorbed at n_1. Then, $T_1^{(L)}$ is the minimum of L independent identical variables with mean τ_1 (expected time for one walk to be absorbed at n_1 starting from a random node).
- $T_2^{(L-1)}$ is the expected *residual* delay (after $T_1^{(L)}$) until *any* of the $L-1$ remaining copies is absorbed at n_2. Then, $T_2^{(L-1)}$ is the minimum of $L-1$ independent identical variables with mean τ_2 (expected *residual* time for one walk to be absorbed at n_2 starting from a random node). Continuing in the same spirit, we arrive at:
- $T_L^{(1)}$, the *residual* delay (after $T_{L-1}^{(2)}$) until the last copy is absorbed at n_L. $T_L^{(1)}$ has mean τ_L (expected *residual* time for one walk to be absorbed at n_L starting from a random node).

Then, the expected convergence delay for L-copy content placement is a sum of these variables as in Eq. (37). $\tau_1, \ldots, \tau_L$ are (residual) absorption delays on the single-copy chain. ■

6.3 Absorption Time Distribution

To find the minimum in Eq. (36) and the minima and sum of random variables in Eq. (37), 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 [44].

Phase-type distributions have two very useful properties, both proven in [44]: (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.

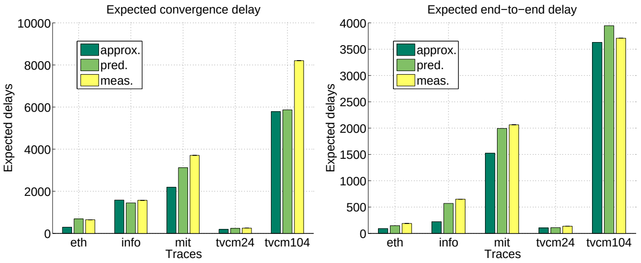
Using this approximation and copy independence, the calculations in Eqs. (36) and (37) are straightforward. The minimum of L independent geometric random variables with parameters $p_k = \tau_k^{-1}$ and with $q_k = 1 - p_k$, is also geometric, with expectation $\tau_{\min} = \left(1 - \prod_{k=1}^{L} q_k\right)^{-1}$. Then, Eq. 36 becomes:

$$\tau_{rt} = \left(1 - \prod_{k=1}^{L}\left(1 - \tau_{s_k d}^{-1}\right)\right)^{-1}. \tag{38}$$

In Eq. (37), 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. 6. 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. (38). 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. (37) is a sum of the independent geometric variables and, by the linearity of expectation,

$$\tau_{cp} = \sum_{i=1}^{L} \tau_i^{(L-i+1)} = \sum_{i=1}^{L}\left(1 - \left(1 - \tau_i^{-1}\right)^{L-i+1}\right)^{-1}. \tag{39}$$

Using Eqs. (38) and (39), we can now predict delays for much larger scenarios, where L is virtually unhampered.



(a) Content Placement (b) SimBet Routing

Figure 9: *Approximation accuracy*

Fig. 9 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.

7 Related Work

Few studies prior to our work incorporate the heterogeneities of node mobility into analyses of DTN algorithms, and no work (to our knowledge) attempts to provide an analytical framework which can be applied to more than one problem.

One of the first approaches was to model a small number of *mobility classes*. For example, in [14], the authors focus on a network with k mobility classes. Each node permanently belongs to only one of the k classes, and mobility patterns are different across classes, but identical inside each class, resulting in different intra- and inter-class meeting rates. To analyze simple DTN routing protocols (epidemic, spray and wait) in such a network, Spyropoulos et al. use the fluid model formulation for epidemic routing. The routing process is treated as a fluid flow and the number of message copies is approximated as a continuous-valued function of time and the various node meeting rates. This results in a system of ordinary differential equations (ODE), which can be solved to obtain performance metrics, such as delivery delay and delivery ratio.

In a more recent study [15], Chaintreau et al. also consider a network formed of a limited number of mobility classes. However, unlike [14], where classes represent stable groups of identical nodes, in [15], the mobility classes are disjoint regions (cells) of the space in which the network operates. Each cell has its own mobility model and nodes adopt this model upon entering the cell. Using this network model, the authors analyze the age of an epidemically spread and constantly updated piece of information. They employ mean field theory to derive asymptotics for the distribution of the age of the piece of information across the network.

Contemporary and closest to our work is the analysis in [13], where the authors also consider a mobility model in which each node pair meets according to an individual rate (potentially different from all other meeting rates). Using this, Boldrini et al. define a very simple Markov chain for *greedy single-copy "social" routing* and use first step analysis to derive the expected delivery delay.

As mentioned in the introduction, these analyses suffer from various shortcomings which DTN-Meteo attempts to improve. [14] and [15] only support a limited degree of heterogeneity and become exceedingly complex with increased heterogeneity. All analyses only deal with a specific problem and a specific (usually very simple) algorithm, such as routing [13,14] or updating content [15]. Moreover, the empirical validation in [13, 14] consists only of small synthetic scenarios, specifically designed to respect initial assumptions.

8 Conclusion and Future Work

In conclusion, 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 a* ***larger class*** *of utility-based algorithms (deterministic and randomized) 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 [24] can be modeled 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 [19]). 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.

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*. ACM WDTN 2005.

[3] Balasubramanian A, Levine B et al. *DTN routing as a resource allocation problem*. ACM SIGCOMM, 2007.

[4] Picu A and Spyropoulos T. *Distributed stochastic optimization in Opportunistic Nets: The case of optimal relay selection*. ACM CHANTS 2010.

[5] Brémaud P. Markov chains : Gibbs fields, Monte Carlo simulation, and queues. Springer, Berlin, Germany, 2001. ISBN 0-38798-509-3.

[6] Groenevelt R, Nain P et al. *The message delay in mobile ad hoc networks*. Perform Eval, 62:210–228, 2005.

[7] Haas ZJ and Small T. *A new networking model for biological applications of ad hoc sensor networks*. IEEE/ACM Trans Netw, 2006.

[8] Spyropoulos T, Psounis K et al. *Performance analysis of mobility-assisted routing*. ACM MobiHoc 2006.

[9] Hanbali AA, Kherani A et al. *Simple models for the performance evaluation of a class of two-hop relay protocols*. IFIP Networking 2007.

[10] Eagle N and Pentland A. *Reality mining: sensing complex social systems*. Pers Ubiq Comput, 10(4):255–268, 2006. ISSN 1617-4909.

[11] Lenders V, Wagner J et al. *Measurements from an 802.11b mobile ad hoc network*. IEEE WoWMoM 2006.

[12] Daly E and Haahr M. *Social network analysis for information flow in disconnected delay-tolerant MANETs*. vol. 8. 2009.

[13] Boldrini C, Conti M et al. *Modelling social-aware forwarding in opportunistic networks*. IFIP PERFORM 2010.

[14] Spyropoulos T, Turletti T et al. *Routing in delay-tolerant networks comprising heterogeneous node populations*. IEEE Trans Mob Comp, 8, 2009.

[15] Chaintreau A, Le Boudec JY et al. *The age of gossip: spatial mean field regime*. ACM SIGMETRICS 2009, 109–120.

[16] Cai H and Eun DY. *Crossing Over the Bounded Domain: From Exponential to Power-Law Intermeeting Time in Mobile Ad Hoc Networks*. IEEE/ACM Trans Netw, 17(5):1578–1591, 2009.

[17] Karagiannis T, Le Boudec JY et al. *Power Law and Exponential Decay of Intercontact Times between Mobile Devices*. IEEE Trans Mob Comp, 9(10):1377–1390, 2010.

[18] Hui P, Crowcroft J et al. *Bubble rap: social-based forwarding in delay tolerant networks*. ACM MobiHoc 2008.

[19] Picu A, Spyropoulos T et al. *An analysis of the information spreading delay in heterogeneous mobility dtns*. IEEE WoWMoM 2012.

[20] Zhang X, Neglia G et al. *Performance modeling of epidemic routing*. Comput Netw, 51:2867–2891, 2007.

[21] Helgason OR, Legendre F et al. *Performance of opportunistic content distribution under different levels of cooperation*. IEEE EW 2010, 903–910.

[22] Lee U, Oh SY et al. *Relaycast: Scalable multicast routing in delay tolerant networks*. IEEE ICNP 2008, 218–227.

[23] Daly E and Haahr M. *Social network analysis for routing in disconnected delay-tolerant MANETs*. ACM MobiHoc 2007.

[24] Gao W, Li Q et al. *Multicasting in delay tolerant networks: a social network perspective*. ACM MobiHoc 2009.

[25] Fall K. *A delay-tolerant network architecture for challenged internets*. ACM SIGCOMM 2003.

[26] Grossglauser M and Tse DNC. *Mobility increases the capacity of ad hoc wireless networks*. IEEE/ACM Trans Netw, 10, 2002.

[27] Spyropoulos T, Psounis K et al. *Efficient routing in intermittently connected mobile networks: the single-copy*. IEEE/ACM Trans Netw, 16(1), 2008.

[28] Lindgren A, Doria A et al. *Probabilistic routing in intermittently connected networks*. ACM SIGMOBILE MC2R, 2003.

[29] Chaintreau A, Hui P et al. *Impact of human mobility on the design of opportunistic forwarding algorithms*. IEEE INFOCOM 2006.

[30] Karagiannis T, Le Boudec JY et al. *Power law and exponential decay of inter contact times between mobile devices*. ACM MobiCom 2007, 183–194.

[31] Castro Sotos AE, Vanhoof S et al. *The transitivity misconception of Pearson's correlation coefficient*. Statistics Education Research Journal, 8(2):33–55, 2008.

[32] Albin SL. *Approximating a Point Process by a Renewal Process, II: Superposition Arrival Processes to Queues*. Operations Research, 32(5):1133–1162, 1984.

[33] Meilijson I. *Limiting properties of the mean residual lifetime function*. Ann Math Statist, 43(1):354–357, 1972.

[34] Cox DR. Renewal Theory. Methuen's monographs on applied probability and statistics. Methuen, London, UK, 1962. ISBN 0-38798-509-3.

[35] Boyd S, Ghosh A et al. *Randomized gossip algorithms*. IEEE/ACM Trans Netw, 14(SI):2508–2530, 2006. ISSN 1063-6692.

[36] Ross S. Stochastic Processes. 1996.

[37] Hui P, Chaintreau A et al. *Pocket switched networks and human mobility in conference environments*. ACM WDTN 2005.

[38] Hsu WJ, Spyropoulos T et al. *Modeling spatial and temporal dependencies of user mobility in wireless mobile networks*. IEEE/ACM Trans Netw, 2009.

[39] Kemeny J and Snell L. Finite Markov Chains. 1960.

[40] Hossmann T, Spyropoulos T et al. *Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing*. IEEE INFOCOM 2010.

[41] Picu A and Spyropoulos T. *Minimum Expected *-cast Time in DTNs*. ICST BIONETICS 2009.

[42] Blondel V, Guillaume JL et al. *Fast unfolding of communities in large networks*. Journal of Stat Mech: Theory and Exp, 2008(10), 2008.

[43] Wolff R. *Poisson Arrivals See Time Averages*. Operations Research, 30(2), 1982.

[44] Asmussen S and Albrecher H. Ruin Probabilities. 2nd ed., 2010.

A Proof for Theorem 4

Theorem 4 (Correctness of greedy content placement). *For all source nodes and all initial copy allocations and for an arbitrary decomposable utility function $U : \Omega \mapsto \mathbb{R}$,* greedy content placement *is correct if and only if, for all $i \in \mathcal{N} \setminus \mathcal{L}^\star$, there exist at least L nodes $j_1, \ldots, j_L \in \mathcal{N}$ with $u(i) < u(j_{(.)})$, such that $p^c_{ij_{(.)}} > 0$ (i.e., $\beta_{ij_{(.)}} > 0$).*

First we state this definition, which we will use in the proof.

Definition 3 (Visit probability). *The random variable $D_{\mathbf{xy}} := \min\{n \geqslant 1 | \mathbf{X}_n = \mathbf{y},\ \mathbf{X}_0 = \mathbf{x}\}$ counts the number of steps needed by the DTN-Meteo Markov chain to get from network state* **x** *to network state* **y**. *$D_{\mathbf{xy}}$ is called* hitting time *from state* **x** *to state* **y**. *If* **y** *is never reached, we set $D_{\mathbf{xy}} = \infty$. The probability to get from state* **x** *to state* **y** *after arbitrarily many steps is called* visit probability *$v_{\mathbf{xy}}$:*

$$v_{\mathbf{xy}} := \mathbb{P}[D_{\mathbf{xy}} < \infty]. \tag{40}$$

Proof. *(L = 1 **case**)* Let us first prove Theorem 4 for the simple case of $L = 1$, when the solution space Ω corresponds to the set of nodes $\mathcal{N}$, and the Markov Chain **P** is one-dimensional. In this case, a potential solution $\mathbf{x} \in \Omega$ of the optimization problem is still an N-element vector, but this vector has a single non-zero element x_i, where i is the node holding the one existing copy of content. Given this fact, we can use the more intuitive notation $\mathbf{x} = i$.

Denote by l the single node in $\mathcal{L}^\star$. l is an absorbing state in the chain, and we are interested in v_{il} for all $i \neq l$.

"Sufficient": If $\forall\, i \neq l,\ \exists\, j \in \mathcal{N}$ with $u(i) < u(j)$, such that $p^c_{ij} > 0$, then we must prove that, for all $i \neq l$:

$$v_{il} = p^c_{il} \cdot A_{il} + v_{il} \cdot p_{ii} + \sum_{\substack{1 \leqslant k \leqslant N \\ k \neq l,i}} v_{kl} \cdot p^c_{ik} \cdot A_{ik} = 1. \tag{41}$$

Let us further develop Eq. (41), using Eq. (9) and Eq. (7):

$$\begin{aligned} v_{il} &= p^c_{il} + v_{il} \cdot \left(1 - \sum_{\substack{1 \leqslant k \leqslant N \\ k \neq i}} p^c_{ik} \cdot A_{ik} \right) + \sum_{\substack{k \neq l,i \\ u(k) > u(i)}} v_{kl} \cdot p^c_{ik} \\ &= p^c_{il} + v_{il} - v_{il} \cdot \sum_{\substack{k \neq i \\ u(k) > u(i)}} p^c_{ik} + \sum_{\substack{k \neq l,i \\ u(k) > u(i)}} v_{kl} \cdot p^c_{ik} \\ 0 &= p^c_{il} - v_{il} \cdot \sum_{\substack{k \neq i \\ u(k) > u(i)}} p^c_{ik} + \sum_{\substack{k \neq l,i \\ u(k) > u(i)}} v_{kl} \cdot p^c_{ik} \\ v_{il} &= \frac{p^c_{il} + \sum\limits_{\substack{k \neq l,i \\ u(k) > u(i)}} v_{kl} \cdot p^c_{ik}}{\sum\limits_{\substack{k \neq i \\ u(k) > u(i)}} p^c_{ik}} = \frac{\sum\limits_{\substack{k \neq i \\ u(k) > u(i)}} v_{kl} \cdot p^c_{ik}}{\sum\limits_{\substack{k \neq i \\ u(k) > u(i)}} p^c_{ik}}. \end{aligned} \tag{42}$$

From Eq. (42), we see that it would suffice that $v_{kl} = 1$ $(k \neq i)$, for $v_{il} = 1$ $(i \neq l)$ to be true. This is perfectly sensible since v_{kl} and v_{il} are almost the same. We already know $v_{ll} = 1$, since $A_{li} = 0$ for all $i \in \mathcal{N}$, hence we have to prove it for $k \neq i, l$. We will use *complete induction*.

Let us take k_1 such that $u(l) > u(k_1) > u(j)$ for all $j \neq l, k_1$; in other words, k_1 is the node with the second highest utility in the network. Hence, for k_1, using our hypothesis that $p^c_{k_1 l} > 0$, Eq. (42) becomes:

$$v_{k_1 l} = \frac{v_{ll} \cdot p^c_{k_1 l}}{p^c_{k_1 l}} = 1. \tag{43}$$

Further, let us take $k_{\alpha-1}$ such that $u(l) > u(k_1) > \cdots > u(k_{\alpha-1}) > u(j)$ for all $j \neq l, k_1, \ldots, k_{\alpha-1}$; this means $k_{\alpha-1}$ is the node with the α-th highest utility in the network. Moreover, assume $v_{k_1 l} = \ldots = v_{k_{\alpha-1} l} = 1$. Then, we will prove that for k_α, the node with the $(\alpha + 1)$-th highest utility, $v_{k_\alpha l} = 1$. From the hypothesis, for this node we have either $p^c_{k_\alpha l} > 0$ or there exists $1 \leqslant \beta < \alpha$ such that $p^c_{k_\alpha k_\beta} > 0$ or both. Then:

$$v_{k_\alpha l} = \frac{v_{ll} \cdot p^c_{k_\alpha l} + \sum\limits_{1 \leqslant \beta < \alpha} v_{k_\beta l} \cdot p^c_{k_\alpha k_\beta}}{p^c_{k_\alpha l} + \sum\limits_{1 \leqslant \beta < \alpha} p^c_{k_\alpha k_\beta}} = \frac{p^c_{k_\alpha l} + \sum\limits_{1 \leqslant \beta < \alpha} p^c_{k_\alpha k_\beta}}{p^c_{k_\alpha l} + \sum\limits_{1 \leqslant \beta < \alpha} p^c_{k_\alpha k_\beta}} = 1. \tag{44}$$

Therefore, we have proven that $v_{il} = 1$ for all $i \neq l$ and thus *greedy content placement* is correct.

"Necessary": If *greedy content placement* is correct, then $\forall\, i \neq l,\ \exists\, j \in \mathcal{N}$ with $u(i) < u(j)$, such that $p^c_{ij} > 0$. The assumption that *greedy content placement* is correct means that $v_{il} = 1$. Using Eq. (41) and our hypothesis, $v_{il} = 1$, we obtain that $p^c_{il} + p_{ii} + \sum\limits_{u(i) < u(j)} p^c_{ij} = 1$. Assuming $p^c_{il} = 0$ and $p^c_{ij} = 0$, for all $j \in \mathcal{N}$ with $u(i) < u(j)$ would mean $p_{ii} = 1$, i.e., i is an absorbing state which implies that $v_{il} = 0$, a contradiction. Therefore, there must exist at least one $j \in \mathcal{N}$ with $u(i) < u(j)$ such that $p^c_{ij} > 0$. This ends the proof for $L = 1$.

*(L > 1 **case**)* The proof for the case of $L > 1$ is more involved but similar in substance. For the sake of presentation we will argue for the case of $L = 2$. The proof is similar in substance for $L > 2$. Denote by l_1 and l_2 the two nodes in $\mathcal{L}^\star$. The state space of the Markov chain is L-dimensional, therefore it explodes with the increase of L. For the current case, $L = 2$, the state/solution space is $\Omega = \mathcal{N} \times \mathcal{N}$. Transitions still happen only at contacts, hence transitions are only along one dimension at a time. In the 2-dimensional case, these transitions are governed, as previously, by factors of the form $A_{ij(k)} = \mathbb{1}\{j \neq k, u(i) < u(j)\}$, where i, j represent the transition dimension and k is the value of the other dimension.

"Sufficient": If $\forall\, i \neq l_1, l_2,\ \exists\, j_1, j_2 \in \mathcal{N}$ with $u(i) < u(j_1)$ and $u(i) < u(j_2)$, such that $p^c_{ij_1} > 0$ and $p^c_{ij_2} > 0$, then *greedy content placement* is correct. We must prove here that, for all initial $i_1, i_2 \neq l_1, l_2$:

$$\begin{aligned} v_{(i_1 i_2)(l_1 l_2)} &= p_{i_1 l_1} \cdot p_{i_2 l_2} + p_{i_1 l_2} \cdot p_{i_2 l_1} + v_{(i_1 i_2)(l_1 l_2)} \cdot p_{i_1 i_1} \cdot p_{i_2 i_2} \\ &\quad + \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq l_1, l_2, i_1 \\ k_2 \neq l_1, l_2, i_2}} v_{(k_1 k_2)(l_1 l_2)} \cdot p^c_{i_1 k_1} \cdot p^c_{i_2 k_2} \cdot A_{i_1 k_1 (i_2)} \cdot A_{i_2 k_2 (i_1)} \\ &= 1. \end{aligned} \tag{45}$$

An expansion of Eq. (45) results in Eq. (46). Using this and a complete induction argument as above, it is easy to prove that *greedy content placement* is correct.

$$
\begin{aligned}
v_{(i_1 i_2)(l_1 l_2)} &= p^c_{i_1 l_1} \cdot p^c_{i_2 l_2} + p^c_{i_1 l_2} \cdot p^c_{i_2 l_1} + v_{(i_1 i_2)(l_1 l_2)} \cdot \left(1 - \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq i_1, k_2 \neq i_2}} p^c_{i_1 k_1} \cdot A_{i_1 k_1 (i_2)} \cdot p^c_{i_2 k_2} \cdot A_{i_2 k_2 (i_1)} \right) + \\
&+ \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq l_1, l_2, i_1 \\ k_2 \neq l_1, l_2, i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} v_{(k_1 k_2)(l_1 l_2)} \cdot p^c_{i_1 k_1} \cdot p^c_{i_2 k_2} \\
&= p^c_{i_1 l_1} \cdot p^c_{i_2 l_2} + p^c_{i_1 l_2} \cdot p^c_{i_2 l_1} + v_{(i_1 i_2)(l_1 l_2)} - v_{(i_1 i_2)(l_1 l_2)} \cdot \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq i_1, k_2 \neq i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} p^c_{i_1 k_1} \cdot p^c_{i_2 k_2} + \\
&+ \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq l_1, l_2, i_1 \\ k_2 \neq l_1, l_2, i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} v_{(k_1 k_2)(l_1 l_2)} \cdot p^c_{i_1 k_1} \cdot p^c_{i_2 k_2} \\
0 &= p^c_{i_1 l_1} \cdot p^c_{i_2 l_2} + p^c_{i_1 l_2} \cdot p^c_{i_2 l_1} - v_{(i_1 i_2)(l_1 l_2)} \cdot \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq i_1, k_2 \neq i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} p^c_{i_1 k_1} \cdot p^c_{i_2 k_2} + \sum_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq l_1, l_2, i_1 \\ k_2 \neq l_1, l_2, i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} v_{(k_1 k_2)(l_1 l_2)} \cdot p^c_{i_1 k_1} \cdot p^c_{i_2 k_2} \\
v_{(i_1 i_2)(l_1 l_2)} &= \frac{p^c_{i_1 l_1} \cdot p^c_{i_2 l_2} + p^c_{i_1 l_2} \cdot p^c_{i_2 l_1} + \sum\limits_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq l_1, l_2, i_1 \\ k_2 \neq l_1, l_2, i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} v_{(k_1 k_2)(l_1 l_2)} \cdot p^c_{i_1 k_1} \cdot p^c_{i_2 k_2}}{\sum\limits_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq i_1, k_2 \neq i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} p^c_{i_1 k_1} \cdot p^c_{i_2 k_2}} = \frac{\sum\limits_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq i_1 \\ k_2 \neq i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} v_{(k_1 k_2)(l_1 l_2)} \cdot p^c_{i_1 k_1} \cdot p^c_{i_2 k_2}}{\sum\limits_{\substack{1 \leqslant k_1, k_2 \leqslant N \\ k_1 \neq i_1, k_2 \neq i_2 \\ u(i_1) < u(k_1), u(i_2) < u(k_2)}} p^c_{i_1 k_1} \cdot p^c_{i_2 k_2}} .
\end{aligned}
\tag{46}
$$

"Necessary": If *greedy content placement* is correct, then $\forall\, i \neq l_1, l_2$, $\exists\, j_1, j_2 \in \mathcal{N}$ with $u(i) < u(j_1)$ and $u(i) < u(j_2)$, such that $p^c_{ij_1} > 0$ and $p^c_{ij_2} > 0$. The assumption that *greedy content placement* is correct means that $v_{(i_1 i_2)(l_1 l_2)} = 1$. Using Eq. (45) and our hypothesis, $v_{(i_1 i_2)(l_1 l_2)} = 1$, we obtain that $p^c_{i_1 l_1} \cdot p^c_{i_2 l_2} + p^c_{i_1 l_2} \cdot p^c_{i_2 l_1} + p_{i_1 i_1} \cdot p_{i_2 i_2} + \sum_{u(i_1) < u(j_1), u(i_2) < u(j_2)} p^c_{i_1 j_1} \cdot p^c_{i_2 j_2} = 1$.

Clearly, assuming $p^c_{il_1} = 0$, $p^c_{il_2} = 0$ and $p^c_{ij} = 0$, for all $j \in \mathcal{N}$ with $u(i) < u(j)$ would mean $p_{i_{(.)} i_{(.)}} = 1$, i.e., $i_{(.)}$ are absorbing states which implies that $v_{(i_1 i_2)(l_1 l_2)} = 0$, a contradiction. Therefore, there must exist at least one $j \in \mathcal{N}$ with $u(i) < u(j)$ such that $p^c_{ij} > 0$.

Further, let us assume that for at least two nodes $i \neq l_1, l_2$ there exists exactly one $j \in \mathcal{N}$ with $u(i) < u(j)$, such that $p^c_{ij} > 0$. Consider the case when for both nodes $j = l_1$, the highest utility node in the network. In Eq. (45), this would once again mean that $p_{i_{(.)} i_{(.)}} = 1$, a contradiction. In conclusion, to ensure that *greedy content placement* is correct, i.e., $v_{(i_1 i_2)(l_1 l_2)} = 1$, we must *always* have at least one of the products $p^c_{i_1 j_1} \cdot p^c_{i_2 j_2} > 0$. This can be ensured only if $\forall\, i \neq l_1, l_2$, $\exists\, j_1, j_2 \in \mathcal{N}$ with $u(i) < u(j_1)$ and $u(i) < u(j_2)$, such that $p^c_{ij_1} > 0$ and $p^c_{ij_2} > 0$. ■