Predictable Disruption Tolerant Networks and Delivery Guarantees

Jean-Marc François and Guy Leduc
Research Unit in Networking (RUN)
Université de Liège, Belgium
{francois,leduc}@run.montefiore.ulg.ac.be

***Abstract*— This article[1] studies disruption tolerant networks (DTNs) where each node knows the probabilistic distribution of contacts with other nodes. It proposes a framework that allows one to formalize the behaviour of such a network. It generalizes extreme cases that have been studied before where *(a)* either nodes only know their contact frequency with each other or *(b)* they have a perfect knowledge of who meets who and when. This paper then gives an example of how this framework can be used; it shows how one can find a packet forwarding algorithm optimized to meet the 'delay/bandwidth consumption' trade-off: packets are duplicated so as to (statistically) guarantee a given delay or delivery probability, but not too much so as to reduce the bandwidth, energy, and memory consumption.**

I. Introduction

Disruption (or *Delay*) *Tolerant Networks* (DTNs, [1]) have been the subject of much research activity in the last few years, pushing further the concept of Ad Hoc networks. Like Ad Hoc networks, DTNs are infrastructureless, thus the packets are relayed from one node to the next until they reach their destination. Moreover, in DTNs node clusters can be completely disconnected from the rest of the network. In this case, nodes must buffer the packets and wait until node mobility changes the network's topology, allowing the packets to be finally delivered.

A network of Bluetooth-enabled PDAs, a village intermittently connected *via* low Earth orbiting satellites, or even an interplanetary Internet ([2]) are examples of disruption tolerant networks.

The atomic data unit is a group of packets to be delivered together. In DTN parlance, it is called a *message* or a *bundle*; we use the latter in the following.

Routing in such networks is particularly challenging since it requires to take into account the uncertainty of mobiles movements. The first methods that have been proposed in the literature are pretty radical and propose to forward bundles in an "epidemic" way ([3], [4], [5]), *i.e.*, to copy them each time a new node is encountered. This method of course results in optimum delays and delivery probabilities, at the expense of an extremely high consumption of bandwidth (and, thus, energy) and memory. To mitigate those shortcomings, the epidemic routing has been enhanced using heuristics that allow the propagation of bundles to a subset of all the nodes ([6], [7], [8]).

Since node's buffer memory is not unlimited, a cache mechanism has been proposed, where the most interesting bundles are kept (*i.e.* those that are likely to reach their destination soon) and the others are discarded when the cache is full ([9], [10], [11], [12], [13], [14]). Those schemes must thus guess when a bundle will reach its destination, which is most of the time computed thanks to frequency contact estimation (which reflects the probability that two given nodes meet in the future).

Few papers explore how the expected delay could be more precisely estimated (notable exceptions are [15], [16]). It has been proved ([17]) that a perfect knowledge of the future node meetings allows the computation of an optimal bundle routing.

This short overview emphasizes two shortcomings:

- Certain networks might be highly predictable (*e.g.* nodes are satellites and links appear and vanish as they revolve around their planet), others are much more chaotic. Previous work suppose either that nodes contacts are perfectly deterministic and known in advance, or that only the contact frequency is known for each pair of nodes. We propose to generalise these approaches and suppose that each node knows a probability distribution of contacts in the (near) future.
- [5] underlines the tradeoff between bundle delivery guarantees and bandwidth/energy consumption: copying the bundles is costly since, in mobile networks, those resources are both scarce. Current schemes use a cache mechanism that ensures each node only receives the most relevant bundles, which somehow mitigates this problem, but does not provide any rationale, except the need to cope with mobiles limited memory. We propose to route the bundles according to the delivery or delay guarantees required by the user, thus only duplicating packets when it is beneficial.

This paper is organised as follows. Section II presents a way to model the contacts between the nodes of a predictable network. Sections II and IV show how the end-to-end delay of bundles can be predicted. Sections V and VI give a routing algorithm that allows to deliver bundles in a manner that meets a given guarantee. Section VII concludes.

[1]This work has been partially supported by the Belgian Science Policy in the framework of the IAP program (Motion P5/11 project) and by the European E-Next NoE and IST-FET ANA project.

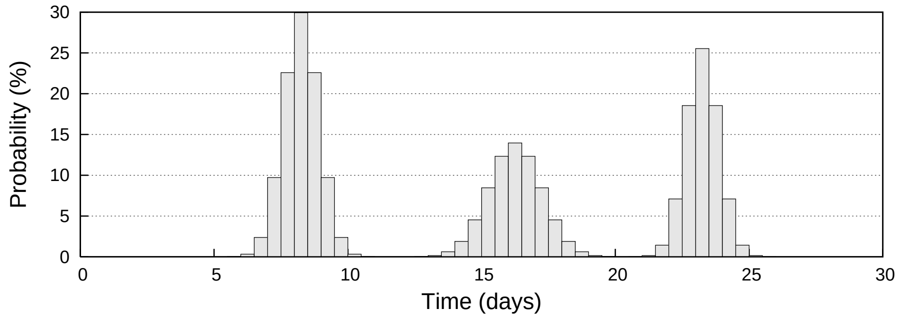


Fig. 1. **Contact profile of a node pair over a month: example.** The height of a bar gives the probability that two nodes meet (*at least* once) during the corresponding 12-hour time period. Here, nodes are supposed to meet at the beginning of each week, but the exact day is unknown.

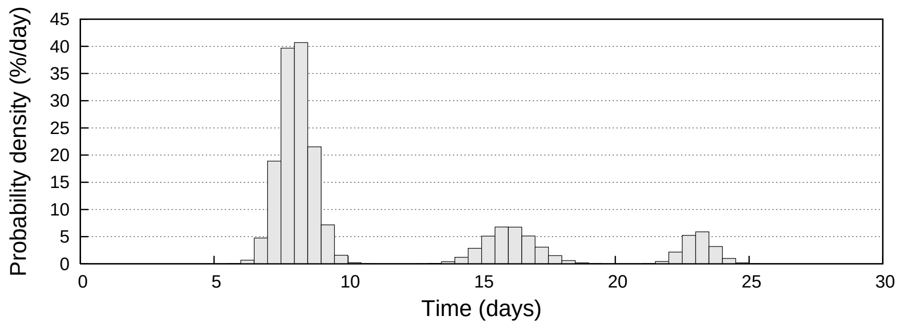


Fig. 2. **First contact probability distribution** corresponding to the contact profile figure 1. (Each bar corresponds to a 12-hour period.)

II. Predictable future contacts

The network is composed of a finite set of wireless nodes $\mathcal{N}$ that can move and thus, from time to time, come into contact.

In the sequel, a *contact* between two nodes happen when those nodes have setup a bi-directional wireless link between them. A contact is always considered long enough to allow all the required data exchanges to take place[2].

A. Contact profiles

We expect the mobiles motion to be, to a certain extent, predictable, yet obviously the degree of predictability varies from one network to another. Sometimes nodes motion is known in advance because they must stick to a given schedule (*e.g.* a network of buses) or because their trajectory can easily be modelled (*e.g.* nodes embedded in a satellite). Other networks are less predictable, yet not totally random: colleagues could be pretty sure to meet every day during working hours, without any other time guarantee. Mobile nodes behaviour could also be learnt automatically so as to extract cyclical contact patterns.

We therefore suppose that each node pair $\{a, b\} \subset \mathcal{N}$ can estimate its contact probability for each time step in the near future. We call it a *contact profile* and denote it $C_{ab} : \mathbb{N} \to [0, 1]$. The time step duration should be chosen small compared to the expected network's end-to-end delay. Figure 1 gives an hypothetical contact profile. In the following, we suppose the profile known for each node pair.

Contact profiles can easily represent situations usually depicted in the literature:

- A constant profile $C_{ab}(t) = k$ describes a node pair that only knows its contact frequency. For example, the profile $C_{ab}(t) = 1/30$ (probability of contact per day) corresponds to two nodes a and b meeting once a month on average.
- Perfect knowledge of nodes meeting times results in a profile made of peaks: $\forall t \in \mathbb{N} : C_{ab}(t) \in \{0, 1\}$.

In practice, unknown contact profiles could be replaced by a null function to get a defensive approximation of their behaviour.

The following sections aim at studying how bundles propagate from one node to another in a network whose nodes' contact profiles are known.

B. First contact distribution

It is easy to deduce the probability distribution of a (first) contact at time t between nodes a and $b \in \mathcal{N}$ given their profile C_{ab}; we denote this distribution d_{ab}. Since the probability of a first contact at time t is the probability of meeting at time step t times the probability not to meet at time steps $0, 1, \ldots, t-1$, we have:

$$d_{ab}(t) = C_{ab}(t) \prod_{i=0}^{t-1} \big(1 - C_{ab}(i)\big) \qquad \forall a, b \in \mathcal{N},\ \forall t \in \mathbb{N} \quad (1)$$

[2]This is a major difference with [17] which does not neglect bundle transmission times.

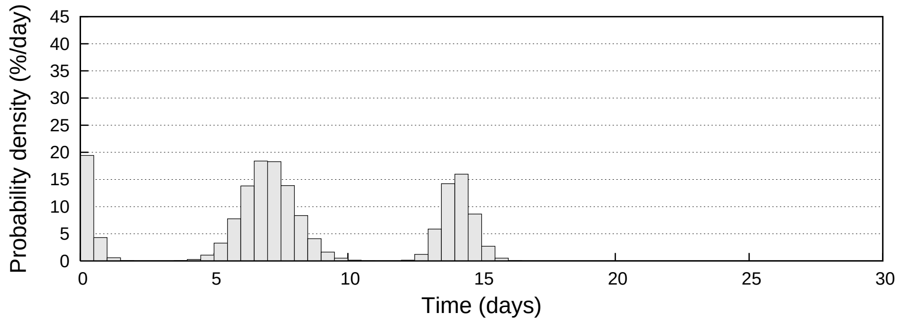


Fig. 3. **The contact probability density $D_{ab}(9,\cdot)$** matching the contact profile given in figure 1.

The distributions domain is $\mathbb{N}$ since contact profiles have been defined using discrete time steps. We extend the distributions to $\mathbb{R}$ to get rid of this artifact. Notice that d_{ab} is not a well-defined probability distribution since its integral over its domain is not equal to 1: two nodes might never meet. Those considerations directly lead to the definition of the first contact distribution set.

Definition 1: The first contact distribution set, $\mathcal{C}$, is the set of functions[3] $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that $\int_0^\infty f(x)\,dx \leq 1$.

Contact profiles have a shortcoming: they do not allow us to express contact interdependencies; for example, they cannot model that two nodes are certain to meet during the weekend without knowing exactly which day. First contact distributions have no such limitations. Therefore, when it is possible, one could find preferable to generate them directly without relying on contact profiles.

Figure 2 gives the d_{ab} distribution corresponding to the contact profile C_{ab} depicted in figure 1.

Notice that if a bundle is delivered directly from a to b, knowing the first contact distribution allows an easy verification of a large spectrum of guarantees, such as the average delay or the probability of delivery before a certain date.

III. Delivery distributions

A. Definition

First contact distributions can be generalized to take into account the knowledge that no contact were made before a certain date.

Let $D_{ab}(T,t)$ be the probability distribution that a and b require a delay of t time steps to meet for the first time after time step T. Since these distributions will be the building blocks that allow us to compute when a bundle can be delivered to its destination, we call them *delivery distributions*. D_{ab} can directly be derived from the contact profile C_{ab}:

$$D_{ab}(T,t) = C_{ab}(T+t) \prod_{i=T}^{T+t-1} \big(1 - C_{ab}(i)\big) \qquad \forall a,b \in \mathcal{N},\ \forall T,t \in \mathbb{N} \quad (2)$$

[3] $\mathbb{R}^+$ denotes the set of positive reals.

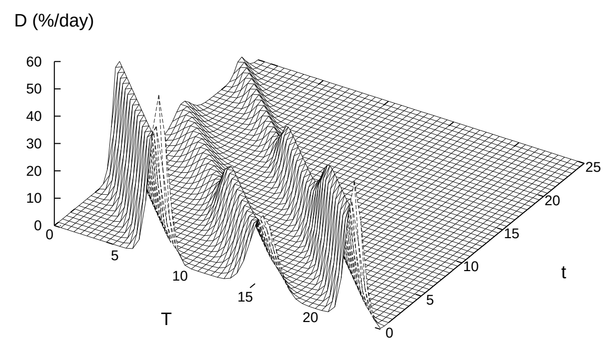


Fig. 5. **The $D_{ab}(T,t)$ function** matching the contact profile given in figure 1.

As before, the domain of these functions can be extended to ${\mathbb{R}^+}^2$.

Definition 2: The *delivery distribution set*, $\mathcal{D}$, holds all the functions $f : {\mathbb{R}^+}^2 \to \mathbb{R}^+$ such that $\forall T \in \mathbb{R}^+ :\ \int_0^\infty f(T,x)\,dx \leq 1$.
Notice the inequality.

The $D_{ab}(T,t)$ distribution corresponding to the contact profile given in figure 1 is plotted in figure 5. Figure 3 plots the function $D_{ab}(9,\cdot)$ (*i.e.* a section of $D_{ab}(T,t)$ in the $T = 9$ plane); the $D(T,\cdot)$ functions of course belong to $\mathcal{C}$ ($\forall T \geq 0$).

Notice that $D_{ab}(T,\cdot)$ is the expected delivery delay distribution for a bundle sent directly from a source a to a destination b if a decides to send it at time T.

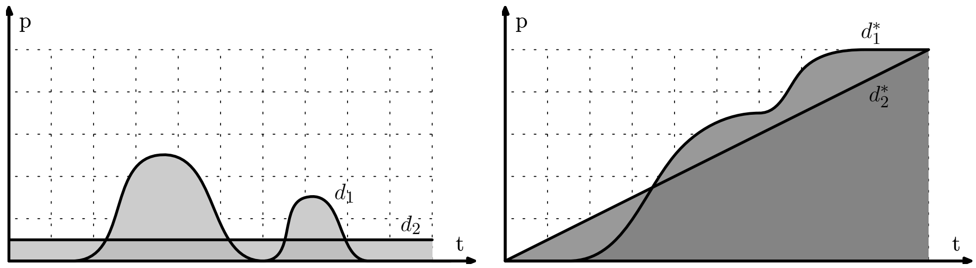
B. Order relation on distributions

We define an order relation between first contact distributions. Intuitively, this relation allows us to compare two distributions to find which one represents more frequent or predictable contacts. A rigorous definition is given below.

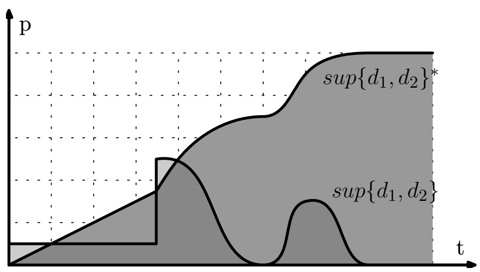
Definition 3: The first contact distributions $d_1 \in \mathcal{C}$ is *greater (or equal)* than $d_2 \in \mathcal{C}$ (denoted $d_1 \succeq d_2$) if and only if:

$$\forall x \geq 0 : \quad \int_0^x d_1(t)\,dt \geq \int_0^x d_2(t)\,dt \qquad (3)$$

This relation is a *partial* order (but not a total order as there exist $d_1, d_2 \in \mathcal{C}$ such that neither $d_1 \succeq d_2$ nor $d_1 \preceq d_2$).



Definition 3 specifies when two contact distributions d_1, d_2 are such that $d_1 \succeq d_2$. The plots show two distribution examples (left-hand plot) and their cumulative function (denoted d_1^* and d_2^*, right-hand plot). We have $d_1 \succeq d_2$ iff $\forall t \geq 0 : d_1^*(t) \geq d_2^*(t)$. Here, neither $d_1 \succeq d_2$ nor $d_2 \succeq d_1$ hold.



The distribution $sup\{d_1, d_2\}$ (left-hand plot) is called *supremum* (or *least upper bound*). Its cumulative function is the maximum of the d_1^* and d_2^* functions; its distribution is the derivative of the cumulative function.
By definition of $sup\{d_1, d_2\}$, if $d \succeq d_1$ and $d \succeq d_2$, then $d \succeq sup\{d_1, d_2\}$ $(\forall d \in \mathcal{C})$. The infimum (or *greatest lower bound*) is defined in a similar manner.
Since every element of $\mathcal{C}^2$ has a corresponding supremum and infimum, the $\succeq$ relation defines a *lattice* structure on $\mathcal{C}$ (and on $\mathcal{D}$).

Fig. 4. The $\succeq$ relation: example.

Figure 4 gives an example of incomparable first contact distributions.

It appears difficult to define a total order on $\mathcal{C}$: comparing the distributions d_1 and d_2 in figure 4 is a matter of choice and depends on the bundle delivery guarantees one wants to enforce. The $\succeq$ relation is thus a least common denominator, and could be replaced in what follows with a more restrictive order definition.

The worst (smallest) element of $\mathcal{C}$ is the $\perp$ (*bottom*) distribution: $\perp(t) = 0$ $(\forall t \geq 0)$. The best (greatest) first contact distribution is denoted $\top$ (*top*): $\top(t) = \delta(t)$ $(\forall t \geq 0)$; the δ symbol denotes the Dirac distribution.

The $\succeq$ relation can be extended to $\mathcal{D}$. For all $D_1, D_2 \in \mathcal{D}$:

$$D_1 \succeq D_2 \iff \forall T \geq 0 : \; D_1(T, \cdot) \succeq D_2(T, \cdot)$$

The $D_\perp$ delivery distribution is such that $\forall T \geq 0 \; : \; D_\perp(T, \cdot) \equiv \perp$. The definition of $D_\top$ follows immediately.

IV. Delivery distribution operators

A. *The* forwarding *operator*

Let D_{sbd} be the delivery distribution associated with the delivery of a bundle from a source node s to a destination d via node b. More precisely, if s decides to send a bundle at time T, it will reach d after a delay described by the $D_{sbd}(T, \cdot)$ distribution. D_{sbd} can be computed thanks to D_{sb} and D_{bd}:

$$D_{sbd} \equiv D_{sb} \otimes D_{bd} \tag{4}$$

The $\otimes$ (or *forwarding*) operator is a function defined for all distribution pair. We have $\otimes : \mathcal{D}^2 \rightarrow \mathcal{D}$:

$$\big(D_1 \otimes D_2\big)(T, t) = \int_0^t D_1(T, x)\, D_2(T + x, t - x)\, dx \tag{5}$$

It is easy to see that this operator is associative but not commutative.

Equation (5) simply states that since the total delivery delay is equal to t, if the delay to reach b is equal to x, then the delay from b to d is $t - x$.

Equation (4) can be generalized: a bundle could be forwarded through several intermediate hops before reaching its destination. We denote D_{s-d} (notice the dash) the delivery delay distribution for a bundle sent from a source s to a destination d at time T; from now on, $\otimes$ will thus be applied to any kind of delivery distributions.

For example, the graph below depicts a simple *delivery path*, *i.e.* a sequence of forwarding nodes; the corresponding delivery distribution is also given.

$$s \longrightarrow a \longrightarrow b \longrightarrow d \;:\quad D_{s-d} \equiv D_{sa} \otimes D_{ab} \otimes D_{bd}$$

We say that two delivery paths with a common source s and destination d are *disjoint* if the intersection of the set of nodes they involve is $\{s, d\}$.

B. *The* duplication *operator*

Let $D_{s\substack{\nearrow d \\ \searrow d}}$ be the delivery distribution associated with the delivery of a bundle from s to d if it is duplicated so as to follow the disjoint delivery paths described by the distributions D_{s-d} and D'_{s-d}. We have:

$$D_{s\substack{\nearrow d \\ \searrow d}} \equiv D_{s-d} \oplus D'_{s-d} \tag{6}$$

The $\oplus$ (or *duplication*) operator is a function $\oplus : \mathcal{D}^2 \rightarrow \mathcal{D}$, defined as follows:

$$\begin{aligned}\big(D_1 \oplus D_2\big)(T, t) = \left(1 - \int_0^t D_1(T, x)\, dx\right) D_2(T, t) + \\ \left(1 - \int_0^t D_2(T, x)\, dx\right) D_1(T, t)\end{aligned} \tag{7}$$

The expected delay computed is that of *the first* bundle to reach the destination d. It is easy to see that $\oplus$ is associative and commutative. We decide that $\otimes$ has a higher precedence than $\oplus$.

Equation (7) is the sum of two terms. Each term is the probability that the bundle reaches the destination after a delay t using one path and that the bundle following the other path is not arrived yet.

Notice that we have both $D_1 \oplus D_2 \succeq D_1$ and $D_1 \oplus D_2 \succeq D_2$ (appendix, corollary 1). This means that, contrary to what happens in deterministic networks, duplicating a bundle to send it along two paths can improve performance: it is not the case that the best path always delivers the bundle first.

The definition of this operator allows us to apply it to arbitrary independent distributions (for example, involving duplication and forwarding). This allows the computation of the distribution associated with a non trivial way to deliver a bundle, such as the one depicted below; the corresponding distribution formula is given on the right. Two arrows leaving a node depict a duplication.

$$\begin{tikzcd}[row sep=small]
 & & e \arrow[r] & d \\
s \arrow[r] \arrow[dr] & b \arrow[ur] \arrow[r] & f \arrow[r] & d \\
 & c \arrow[rr] & & d
\end{tikzcd} : \quad \begin{aligned} D_{s-d} \equiv \big(D_{sc} \otimes D_{cd}\big) \oplus \big(D_{sb} \otimes \\ (D_{be} \otimes D_{ed} \oplus D_{bf} \otimes D_{fd})\big)\end{aligned}$$

Figure 6 shows examples of the distributions obtained using those operators. As expected, the "duplication" operator shortens the delays and increases the delivery probability.

C. *The* scheduling *operator*

Let $D_{s\substack{-d \\ \searrow d}}$ be the delivery distribution that, every time a bundle has to be sent, chooses the best delivery strategy out of D_{s-d} and D'_{s-d}. We have:

$$D_{s\substack{-d \\ \searrow d}} \equiv D_{s-d} \oslash D'_{s-d} \tag{8}$$

The definition of $\oslash$ is straightforward. It is a function $\oslash : \mathcal{D}^2 \to \mathcal{D}$ such that:

$$\big(D_1 \oslash D_2\big)(T,t) = \begin{cases} D_1(T,t) & \text{if } D_2(T,\cdot) \not\succeq D_1(T,\cdot) \\ D_2(T,t) & \text{otherwise} \end{cases} \tag{9}$$

If s sends a bundle at time T, it is delivered using $D_2(T,\cdot)$ if and only if $D_2(T,\cdot) \succeq D_1(T,\cdot)$. This operator is not commutative since $\succeq$ is not a total order: when $D_1(T,\cdot)$ and $D_2(T,\cdot)$ cannot be compared, $D_1(T,\cdot)$ is chosen. We decide that $\oslash$ has a lower precedence than both $\otimes$ and $\oplus$.

The following example involves all the operators defined above. Two arrows leaving a node, one of them dotted, depict a scheduling operation. The dotted arrow leads to the second argument of $\oslash$, emphasizing the operator's non-commutativity.

$$\begin{tikzcd}[row sep=small]
 & & e \arrow[r] & d \\
s \arrow[r] \arrow[dr] & b \arrow[ur, dotted] \arrow[r] & f \arrow[r] & d \\
 & c \arrow[rr] & & d
\end{tikzcd} : \quad \begin{aligned} D_{s-d} \equiv \big(D_{sc} \otimes D_{cd}\big) \oplus \big(D_{sb} \otimes \\ (D_{bf} \otimes D_{fd} \oslash D_{be} \otimes D_{ed})\big)\end{aligned}$$

D. *Delivery schemes*

We have defined a *delivery path* as a delivery strategy that only involves forwarding.

A *delivery scheme* with source s and destination d is a general delivery strategy that allows a bundle to be delivered from s to d. It can use an arbitrary number of forwarding, duplication and scheduling operations. A delivery path is thus a particular delivery scheme.

Two delivery schemes from s to d are *disjoint* if the intersection of the set of nodes they involve is $\{s, d\}$.

V. Delivery guarantees

Knowing the delay distribution $d_{s-d} \in \mathcal{C}$ associated with the delivery of a bundle allows us to verify a large range of conditions on permissible delays or on delivery probabilities.

For example, the condition

$$\int_0^\infty d_{s-d}(t)\, t\, dt \leq d_{\text{max}}$$

imposes a maximum expected delay d_{max}, while

$$\int_0^{\text{1h}} d_{s-d}(t)\, dt \geq .9 \quad \text{and} \quad \int_0^{\text{24h}} d_{s-d}(t)\, dt \geq .99$$

matches distributions delivering a bundle in less than one hour nine times out of ten, and in less than a day with a probability of 99%.

We naturally impose that a condition fulfilled for a certain delivery scheme must be fulfilled for better schemes.

Definition 4: A *delivery condition* C is a predicate: $C : \mathcal{C} \to \{\text{true}, \text{false}\}$ with $\forall d_1, d_2 \in \mathcal{C}$ such that $d_1 \succeq d_2$: $C(d_2) \Longrightarrow C(d_1)$.

A condition C can be extended to a delivery distribution $D \in \mathcal{D}$: $C(D) \iff \forall T \geq 0 : C\big(D(T,\cdot)\big)$.

VI. Delivering bundles with guarantees

A. *Probabilistic Bellman-Ford*

Algorithm 1 adapts the Bellman-Ford algorithm to predictable disruption tolerant networks. In this section, *we do not allow bundle duplication*. Notice that, in general, the concept of "shortest path" is meaningless since the $\preceq$ relation is a *partial* order.

Similarly to the Bellman-Ford algorithm, algorithm 1 computes, for every node $n \in \mathcal{N}$, the best distribution leading to the destinations found so far (B_n). This distribution is propagated to its neighbours (*i.e.* all the other nodes since the network is infrastructureless).

Once node x receives the best delivery distribution B_y found by y, it computes the delivery distribution obtained if it would send the bundle directly to y, and if y would forward it according to B_y. The resulting distribution is denoted D_{xy-d} (line 6).

D_{xy-d} is compared to the best known distribution to the destination (B_x) by means of the $\oslash$ operator. If D_{xy-d} is better than B_x on some time intervals, B_x is updated (line 9).

The algorithm terminates once no more B_x distribution is updated.

Figures 7 and 8 demonstrate how the algorithm works by means of a small example.

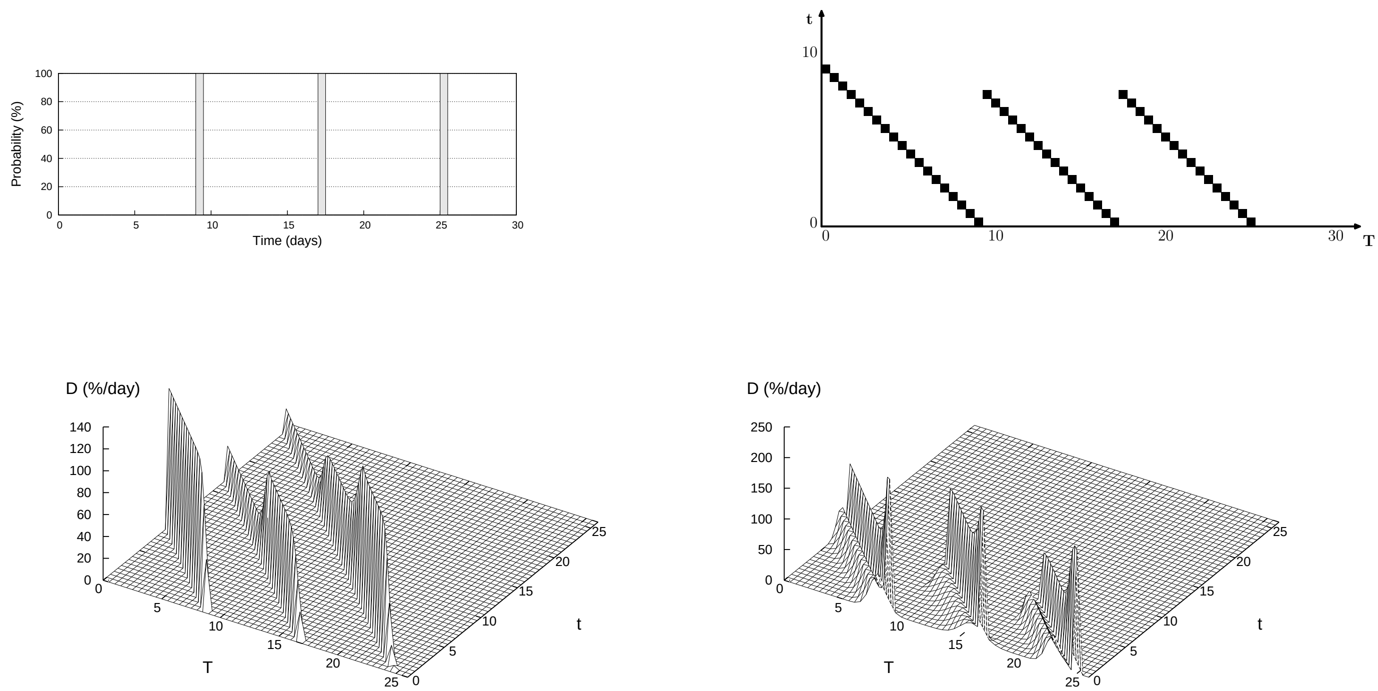


Fig. 6. ***Forwarding* ($\otimes$) and *duplication* ($\oplus$) operators: example**. We denote D_1 the delivery distribution depicted in figure 5. The top part of this figure depicts a contact profile (left) and the associated delivery distribution D_2 (dark squares represent a probability equal to 1). The left 3D plot depicts $D_1 \otimes D_2$, the right one $D_1 \oplus D_2$.

Algorithm 1: Probabilistic Bellman-Ford

Data: d is the destination node

1 $\forall\, x \in \mathcal{N} \setminus \{d\} :\ B_x \leftarrow D_\perp$;
2 $B_d \leftarrow D_\top$;
3 **repeat**
4   stabilized $\leftarrow$ true;
5   **forall** $x \in \mathcal{N}$ **do**
6   **forall** $y \in \mathcal{N}$ **do**
7       $D_{xy-d} \leftarrow D_{xy} \otimes B_y$;
8   **if** $B_x \neq B_x \oslash D_{xy-d}$ **then**
9   stabilized $\leftarrow$ false;
10         $B_x \leftarrow B_x \oslash D_{xy-d}$;
11   **end**
12   **end**
13   **end**
14 **until** stabilized ;

As mentioned before, this algorithm generalizes both [12] (*i.e.* converges to the "*shortest expected path*") and [17][4] (*i.e.* finds the exact shortest path in the case of perfectly predictable networks).

The delivery computed by this algorithm depends on the order at which the elements of $\mathcal{N}$ are picked up (lines 5 and 6). In practice, it might be preferable to rely on a heuristic to choose the preferred elements first.

[4]To be fair, this work also deals with message transmission delays, which are not considered here.

B. Guarantees

Our aim is now to find a way to deliver bundles that fulfills a given condition C as specified in definition 4, while trying to minimize the network's bandwidth/energy/memory consumption.

Ideally, the DTN is predictable enough to enforce condition C without duplicating any bundle. We thus propose to rely on algorithm 1 to find a first delivery scheme (and, thus, a first delivery distribution D_1).

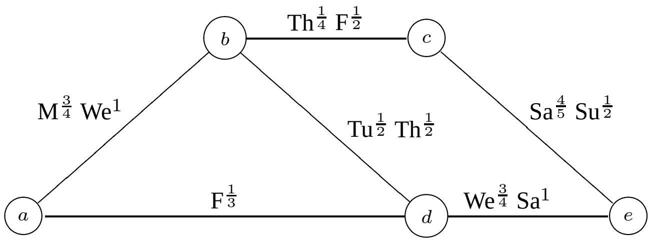
If C is not fulfilled by D_1, we search for another fast bundle forwarding scheme using algorithm 1; let D_2 be its delivery distribution. We then duplicate the bundle on both delivery schemes, yielding a distribution $D_1 \oplus D_2$. We have already pointed out that $D_1 \oplus D_2 \succeq D_1$, thus $C(D_1 \oplus D_2)$ is more likely to be *true* then $C(D_1)$.

This process is iterated until C is finally fulfilled.

As mentioned in section IV-B, the distribution computed by the "duplication" ($\oplus$) operator is biased if its operands are not *independent* distributions. The simple distribution formula $(D_{sb} \otimes D_{bd}) \oplus (D_{sb} \otimes D_{bd})$ brings to light the problem caused by dependent distributions.

To avoid this bias, we ensure that D_1 and D_2 are independent by forbidding D_2 to rely on the nodes involved in D_1 (source and destination nodes excluded, line 5).

The resulting algorithm is given below.



This graph gives the contact profiles of the nodes $a, b, c, d, e \in \mathcal{N}$.
Unconnected nodes never meet each other: they have a null contact profile (and a corresponding delivery distribution $D_\perp$).
The label connecting the other nodes describes which days they might have a contact. For example, there is one chance out of four that b and c meet on Thursday, and one out of two on Friday.

Fig. 7. **A predictable network.**

The opposite table shows how our probabilistic Bellman-Ford algorithm behaves. This example is based on a simple network made of 5 nodes. The nodes contact profiles are given in figure 7. In this example, a is the source node and e is the destination.

At first, all the nodes (but the destination) have no knowledge of any path to the destination; their best distribution is thus set to $D_\perp$. The destination's delivery distribution to itself is of course $D_\top$.

Line 2 depicts the results obtained after the first iteration. Since only c and d have contacts with the destination, only B_c and B_d are modified. They are set to the direct contact with the destination distribution since, for example, $D_{c-e} \otimes D_\top = D_{c-e}$. The delivery distributions are depicted as a square plot; the x-axis is the bundle sending time, the y-axis is the delay to reach the destination. Each square represents a 24 hour period, the first column matches bundles sent on Monday.

During the next iteration (line 3), a discovers it might meet with d before d meets e. B_a is thus changed to $D_{a-d} \otimes D_{d-e}$. The bundles received by b can be forwarded to c or d. The distributions D_{b-c-e} and D_{b-d-e} are thus compared; bundles sent Tuesday or before are sent *via* d, those sent after Tuesday are sent *via* c.

The last iteration allows a to decide when bundles should be sent to b or d. The distributions B_a and $D_{a-b} \otimes B_b$ are thus compared; the latter is given between parentheses. Neither c nor d should forward bundles to b, thus B_c and B_d are left untouched.

The algorithm is stabilized since neither b, c, or d should forward bundles *via* a.

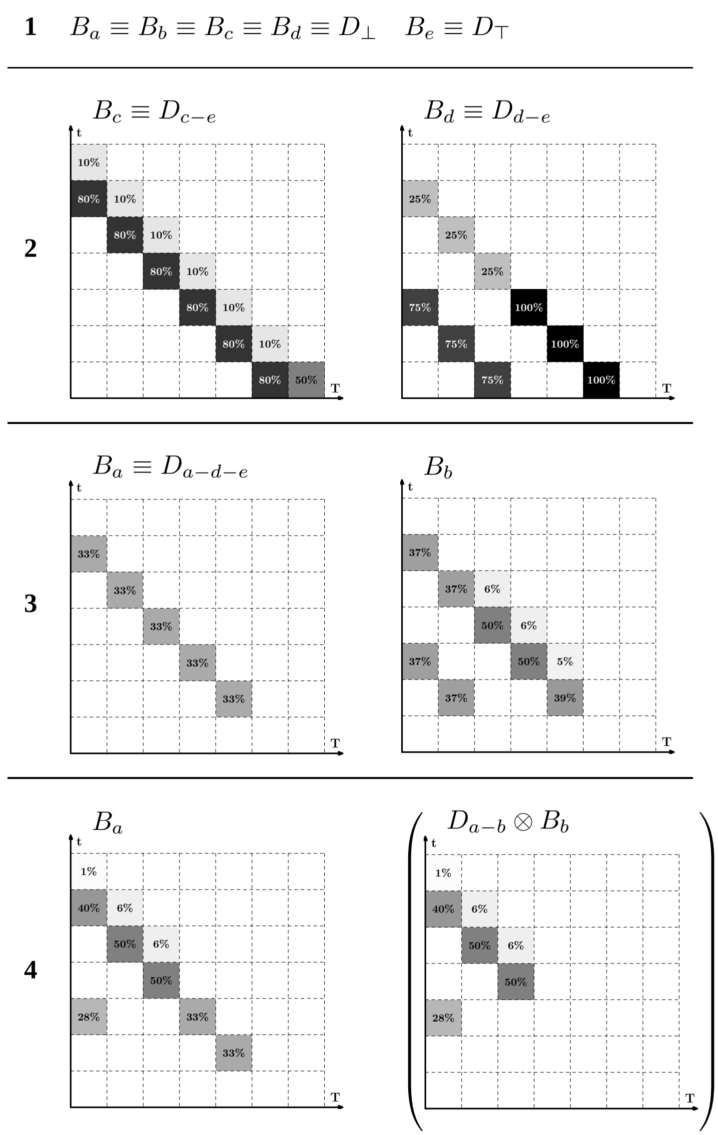


Fig. 8. **Probabilistic Bellman-Ford:** example.

Algorithm 2: Constrained probabilistic delivery

Data: Network nodes $\mathcal{N}$; delivery condition C
Data: Bundle source s and destination d

```
1 B ← D_⊥
2 repeat
3 |  Using nodes in 𝒩, compute D ∈ 𝒟 via algorithm 1
4 |  B ← B ⊕ D
5 |  𝒩 ← 𝒩 \ {nodes involved in D} ∪ {s, d}
6 until C(B) or 𝒩 = {s, d}
```

Nothing guarantees of course that there exists a way to deliver bundles that satisfies C: even an epidemic broadcasting might not suffice.

C. More on disjoint delivery schemes

The *constrained probabilistic delivery* algorithm above computes a delivery scheme that consists of duplicating the bundle to multiple, independent, non-duplicating delivery schemes.

To ensure independence, algorithm 2 enforces those non-duplicating delivery schemes to operate on completely distinct node sets. This might be too stringent if the network is small or sparse. We thus propose to allow such a delivery scheme to use nodes that are *unlikely* to receive a bundle according to the other schemes. The resulting delivery distributions will thus be *almost* independent.

Line 5 of algorithm 2 is thus changed: only the nodes involved in D with a probability higher than a given threshold are removed. The specific value of this threshold is a function of the network considered.

The rest of this section explains how to compute the probability that a given node receives a bundle, given a (non-duplicating) delivery scheme computed by algorithm 1.

We have seen that the proposed modified Bellman-Ford algorithm does not lead to a simple routing table: if a bundle reaches a given node at time T, its next hop depends on its destination *and on* T. Each node n divides time in intervals $I_1^n, I_2^n, \ldots$ (by means of the $\oslash$ operator, algorithm 1 line 7), and each interval I matches a given next hop $H_n(I)$. In the example figure 8, node b has defined two intervals: $I_1^b = [\text{Monday}, \text{Tuesday}]$ and $I_2^b = [\text{Wednesday}, \text{Sunday}]$; $H_b(I_1^b) = d$ and $H_b(I_2^b) = c$.

A bundle crosses a number of nodes on its way to its destination. We compute the probability P_n that a given node n is one of them.

Let s be the bundle source node and d the destination. If the bundle is ready to be sent at time T, it should reach $n = H_s(I)$, where I is the time interval of s such that $T \in I$. The bundle arrival time at n follows the contact distribution $N(x) = D_{sn}(T, x)$, thus $P_n = \int_0^\infty N(x)\,dx$.

Once the bundle has been received by n, each time interval

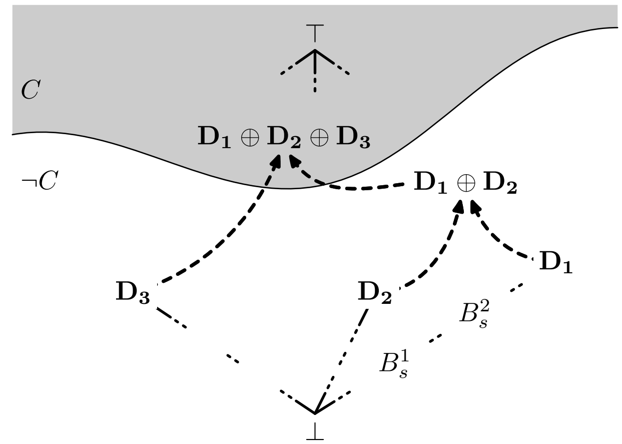


Fig. 9. **Finding a fast delivery scheme that fulfills condition C using algorithms 1 and 2**. This figure represents the delivery distribution lattice (introduced in figure 4); it is depicted the usual way (basically, an element is greater than another one if it is placed above). The greyed area corresponds to elements that satisfy condition C. **1.** The adapted Bellman-Ford algorithm is used to find a distribution (D_1) that characterizes a fast way to deliver bundles; B_s^i denotes the source node's best distribution found after i iterations. We have $\perp \preceq B_s^1 \preceq B_s^2 \preceq \cdots \preceq D_1$. **2.** Since $\neg C(D_1)$, another disjoint delivery distribution, D_2, is computed using algorithm 1. Combined with D_1, it leads to $D_1 \oplus D_2$ which still does not satisfy C. D_3 is thus computed, and combined with D_1 and D_2, gives a satisfactory delivery scheme $D_1 \oplus D_2 \oplus D_3$. We have $D_1 \preceq D_1 \oplus D_2 \preceq D_1 \oplus D_2 \oplus D_3$.

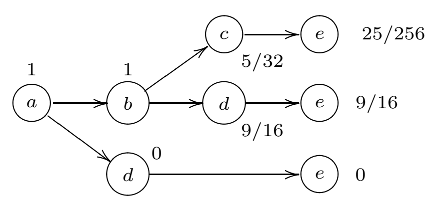
I_i^n matches a potential next hop $n_i = H_n(I_i^n)$. We have:

$$N_i^n(x) = \int_{\{t \in I_i^n | t \leq x\}} N(t)\, D_{nn_i}(t, x - t)\, dt \qquad (10)$$

$$P_{n_i} = \int_0^\infty N_i^n(x)\, dx \qquad (11)$$

Equation (10) gives the bundle time arrival distribution at the next hop n_i. This process can be continued recursively until the probability of receiving the bundle is known for all nodes.

The bundle forwarding process can be represented by a graph. The children of a node are the potential next hops. The graph obtained for the example depicted in figure 8 is given below.



The numbers labelling the nodes are the probabilities of receiving the bundle, as given by (11). The destination e thus receives the bundle with a probability of $\frac{25}{256} + \frac{9}{16}$.

VII. Conclusion and Future Works

We propose to model contacts between a disruption tolerant network's mobile nodes as a random process, characterized by contact distributions. Such a description is more general than those generally encountered in the literature, and allows, for example, to model a perfectly deterministic network.

We show how such contact distributions can be combined to compute the bundle delivery delay distribution corresponding to a given delivery strategy (*i.e.* a description of the nodes forwarding decisions). We show how the Bellman-Ford algorithm can be adapted to cope with such stochastic networks.

There is a tradeoff between a bundle's delivery probability/delay and the consumption of network resources. We propose to duplicate bundles along disjoint "shortest" path so as to meet a given delivery guarantee without consuming too many resources. The corresponding algorithms are given.

This work can be continued along several lines.

We have proposed a way to route bundles through the network; other routing strategies should be explored and compared.

Three operators on delivery distributions have been defined. Others could be added so as to describe more subtle routing decisions, or to deal with bundles' transmission delays.

Real network traces should be analysed so as to quantify their predictability, to compare delivery strategies, and to measure how predictability impacts performance.

APPENDIX

Lemma 1: $\forall D_1, D_2, D_3 \in \mathcal{D}$, we have $D_2 \succeq D_3 \Rightarrow D_1 \oplus D_2 \succeq D_1 \oplus D_3$.

Proof: Given the definition of $\oplus$ and $\succeq$, one must prove that, $\forall D_1, D_2 \in \mathcal{D}$, $\forall T \geq 0, t \geq 0$, given $D_2 \succeq D_3$:

$$\begin{aligned}\int_0^t \left[1 - \int_0^x D_1(T,y)\,dy\right] D_2(T,x) + \\ \left[1 - \int_0^x D_2(T,y)\,dy\right] D_1(T,x)\,dx \\ \geq \int_0^t \left[1 - \int_0^x D_1(T,y)\,dy\right] D_3(T,x) + \\ \left[1 - \int_0^x D_3(T,y)\,dy\right] D_1(T,x)\,dx \end{aligned} \tag{12}$$

The left-hand part can be written as:

$$\begin{aligned}\int_0^t D_1(T,x)\,dx + \int_0^t D_2(T,x)\,dx - \\ \int_0^t \int_0^x \left[D_1(T,x)D_2(T,y) + D_1(T,y)D_2(T,x)\right]\,dy\,dx\end{aligned} \tag{13}$$

Changing the double integral's integration order, the last term of (13) is equal to:

$$\begin{aligned} & \int_0^t D_1(T,x) \int_0^x D_2(T,y)\,dy\,dx + \\ & \quad \int_0^t D_1(T,x) \int_x^t D_2(T,y)\,dy\,dx \\ = & \int_0^t D_1(T,x)\,dx \int_0^t D_2(T,y)\,dy \end{aligned} \tag{14}$$

The same procedure can be applied to the right-hand part of (12). (12) is thus equivalent to:

$$\int_0^t D_2(T,x)\,dx \geq \int_0^t D_3(T,x)\,dx \tag{15}$$

Which holds by hypothesis. ■

Corollary 1: $\forall D_1, D_2 \in \mathcal{D}$, we have $D_1 \oplus D_2 \succeq D_1$.

Proof: From Lemma 1, $D_1 \oplus D_2 \succeq D_1 \oplus \perp = D_1$. ■

REFERENCES

[1] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay toerant networks: overview and challenges," in *IEEE Communications Surveys and Tutorials*, vol. 8-1, 2006, pp. 24–37.

[2] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking - an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[3] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. TR CS-200006, April 2000.

[4] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2003, pp. 233–244.

[5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *ASPLOS, San Jose, CA*, Oct. 2002.

[6] F. Tchakountio and R. Ramanathan, "Tracking highly mobile endpoints," in *WOWMOM '01: Proceedings of the 4th ACM international workshop on Wireless mobile multimedia*. New York, NY, USA: ACM Press, 2001, pp. 83–94.

[7] A. Spuropoulos, K. Psounis, and C. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proceedings of IEEE SECON*, October 2004.

[8] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proceedings of SIGCOMM 2005*, 2005.

[9] D. Nain, N. Petigara, and H. Balakrishnan, "Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks," in *WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, March 2003.

[10] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. New York, NY, USA: ACM Press, 2005, pp. 229–236.

[11] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[12] K. Tan, Q. Zhang, and W. Zhu, "Shortest path routing in partially connected Ad Hoc networks," in *Proc. of IEEE GLOBECOM'03*, vol. 2, December 2003, pp. 1038–1042.

[13] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. New York, NY, USA: ACM Press, 2005, pp. 237–243.

[14] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. New York, NY, USA: ACM Press, 2005, pp. 276–283.

[15] C. Shen, G. Borkar, S. Rajagopalan, and C. Jaikaeo, "Interrogation-based relay routing for ad hoc satellite networks," in *IEEE Globecom*, Taipei, Taiwan, November 17-21 2002.

[16] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 183–189.

[17] S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in networks with predictable mobility," Georgia Institute of Technology, Tech. Rep. GIT-CC-04-7, 2004.