Delivery Guarantees in Predictable Disruption Tolerant Networks

Jean-Marc François and Guy Leduc

Research Unit in Networking (RUN)
DEECS — Institut Montefiore, B28 — Sart-Tilman
University of Liège
4000 Liège, Belgium
{francois,leduc}@run.montefiore.ulg.ac.be

Abstract. This article 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 either *(a)* 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.

1 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. However, 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 method that has

This work has been supported by the Belgian Science Policy in the framework of the IAP program (Motion P5/11 project) and by the IST-FET ANA project.

been proposed in the literature is pretty radical and proposes to forward bundles in an "epidemic" way ([3, 4]), *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 ([5, 6]).

More advance heuristics have been introduced to cope with the nodes limited memory. Cache mechanisms have 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 ([7–10]).

Few papers explore how the expected delay could be more precisely estimated (notable exceptions are [11, 12]). It has been proved ([13]) that a perfect knowledge of the future node meetings allows the computation of an optimal bundle routing.

This short introduction emphasizes two shortcomings:

- Previous works suppose either that nodes contacts are perfectly deterministic, or that only the contact frequency is known for each pair of nodes. In this paper, we introduce a framework which generalizes those extreme cases and formalizes the nodes contact predictability. It allows one to compute the expected impact of a particular bundle forwarding strategy;
- Previous works only propose bundle forwarding *heuristics*. In what follows, we give an example of how the above-mentioned framework can be used to find a bundle routing strategy that fulfills delivery guarantees while limiting bandwidth/energy consumption.

2 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 happens 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[1].

2.1 Contact profiles

We expect the mobiles motion to be 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

[1] This is a major difference with [13] which does not neglect bundle transmission times.

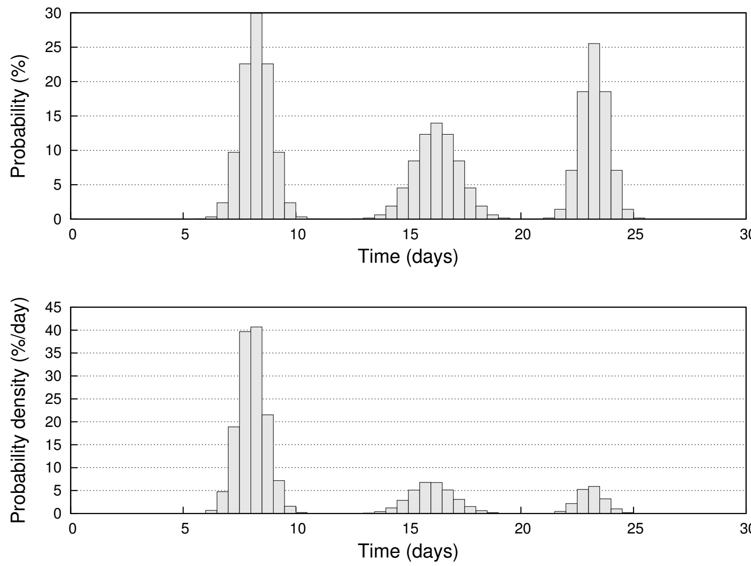


Fig. 1. Contact profile and first contact distribution: example. *Top:* A contact profile: the height of a bar gives the probability that two nodes meet (*at least* once) during the corresponding 12-hour time period. *Bottom:* The corresponding first contact distribution; each bar corresponds to a 12-hour period.

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 (discrete) each time step in the near future. We call it a *contact profile* and denote it $C_{ab} : \mathbb{N} \to [0, 1]$. 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 $C_{ab}(t) = 1/30$ (contact probability per day) corresponds to two nodes a and b meeting once a month on average.
- Perfect knowledge of 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 constant function equal to zero on its domain 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.

2.2 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 $(\forall a, b \in \mathcal{N})$:

$$d_{ab}(t) = C_{ab}(t) \prod_{i=0}^{t-1} \big(1 - C_{ab}(i)\big) \qquad \forall t \in \mathbb{N} \tag{1}$$

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.

Definition 1. *The first contact distribution set, $\mathcal{C}$, is the set of functions*[2] $f : \mathbb{R}^+ \to \mathbb{R}^+$ *such that* $\int_0^\infty f(x)\, dx \leq 1$*.*

Contact profiles do not allow us to express contact interdependencies; for example, they cannot model that two nodes are certain to meet exactly once during the weekend without knowing exactly which day (if a probability of .5 is assignated to Saturday and Sunday, there is a .25 probability that the nodes will meet twice). 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 1 gives an example contact profile C_{ab} (top) and the corresponding first contact distribution d_{ab} (bottom).

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.

3 Delivery distributions

3.1 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} $(\forall a, b \in \mathcal{N})$:

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

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

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.

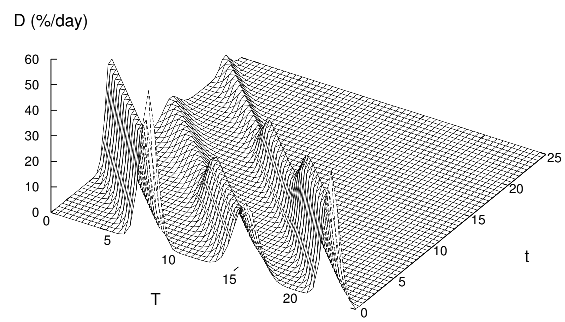


Fig. 2. Contact probability density. The $D_{ab}(T,t)$ delivery distribution matching the contact profile given in figure 1.

Figure 2 draws the $D_{ab}(T,t)$ distribution corresponding to the contact profile given in figure 1. Notice that 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.

3.2 Order relation on distributions

We define an order relation between first contact distributions. Intuitively, this relation allows one 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:*

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

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 $d_1 \preceq d_2$; see [14] for more details).

It appears difficult to define a total order on $\mathcal{C}$: comparing two distributions that cannot be ordered using the $\succeq$ relation 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.

4 Delivery distribution operators

4.1 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 $\otimes : \mathcal{D}^2 \to \mathcal{D}$:

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

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 $\{s, d\}$.

4.2 The *duplication* operator

Let $D_{s\substack{-d \\ \searrow d}}$ 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:

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

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

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

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. Operators $\otimes$ and $\oplus$ can be combined to consider more complex forwarding strategies, assigning a higher precedence to $\otimes$.

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.

It can be proven that we have both $D_1 \oplus D_2 \succeq D_1$ and $D_1 \oplus D_2 \succeq D_2$. 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.

Figure 3 shows an example of the distributions obtained using the "duplication" operator. As expected, duplicating bundles shortens the delays and increases the delivery probability.

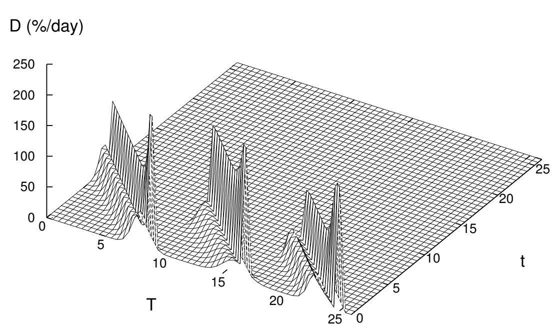


Fig. 3. *Duplication* ($\oplus$) operator: example. We denote D_1 the delivery distribution depicted in figure 2. Let D_2 be a distribution that corresponds to nodes that are certain to meet on day 9, 16 and 25. This plot depicts $D_1 \oplus D_2$.

4.3 The *scheduling* operator

Let $D_{s\overset{-d}{\underset{d}{\cdot}}}$ 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:

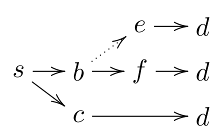
$$D_{s\overset{-d}{\underset{d}{\cdot}}} \equiv D_{s-d} \oslash D'_{s-d} \tag{8}$$

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. This new operator can be combined with the other two ($\otimes$ and $\oplus$) assigning it a lower precedence.

The following example involves all the operators defined above. Two plain arrows leaving a node depict a duplication. 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.



$$: \quad D_{s-d} \equiv \big(D_{sc} \otimes D_{cd}\big) \oplus \big(D_{sb} \otimes (D_{bf} \otimes D_{fd} \oslash D_{be} \otimes D_{ed})\big)$$

4.4 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.

5 Delivery guarantees

Knowing the delay distribution $d \in \mathcal{C}$ 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

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

imposes a maximum expected delay $d_{\max}$, while

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

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:* $C : \mathcal{C} \to \{true, false\}$ *iff* $\forall d_1, d_2 \in \mathcal{C}$ *such that* $d_1 \succeq d_2$*, we have* $C(d_2) \Longrightarrow C(d_1)$*.*

A condition C can be extended to a delivery distribution $D \in \mathcal{D}$:

$$C(D) \iff \forall T \geq 0 : C\big(D(T, \cdot)\big)$$

6 Delivering bundles with guarantees

6.1 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.

Algorithm 1: Probabilistic Bellman-Ford

Data: d is the destination node

$\forall\, x \in \mathcal{N} \setminus \{d\} :\ B_x \leftarrow D_\perp$;
$B_d \leftarrow D_\top$;
repeat
    stabilized $\leftarrow$ true;
    forall $x \in \mathcal{N}$ **do**
        forall $y \in \mathcal{N}$ **do**
            $D_{xy-d} \leftarrow D_{xy} \otimes B_y$;
            if $B_x \neq B_x \oslash D_{xy-d}$ **then**
                stabilized $\leftarrow$ false;
                $B_x \leftarrow B_x \oslash D_{xy-d}$;
until stabilized ;

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.

As mentioned before, this algorithm generalizes both [15] (*i.e.* converges to the *"shortest expected path"*) and [13][3] (*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.

6.2 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; see algorithm 2.

As mentioned in section 4.2, the distribution computed by the "duplication" ($\oplus$) operator is biased if its operands are not *independent* 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). More details can be found in [14].

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

7 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.

We have setup a framework that shows how such contact distributions can be combined to compute the bundle delivery delay distribution corresponding to

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

Algorithm 2: Constrained probabilistic delivery

Data: Delivery condition C
Data: Bundle source s and destination d

$B \leftarrow D_\perp$

repeat
  Using nodes in $\mathcal{N}$, compute $D \in \mathcal{D}$ *via* algorithm 1
  $B \leftarrow B \oplus D$
  $\mathcal{N} \leftarrow \big(\mathcal{N} \setminus \{\text{nodes involved in } D\}\big) \cup \{s, d\}$
until $C(B)$ **or** $\mathcal{N} = \{s, d\}$

a given delivery strategy (*i.e.* a description of the nodes forwarding decisions). This framework is formally defined and quite generic; it can be used to evaluate quantitatively the performance of new routing protocols. It could be expanded with new operators describing other (more subtle) forwarding schemes. A significant improvement would be to modify the framework so as to deal with bundles transmission delays.

As a future work, real network traces can be analysed so as to quantify their predictability; the delivery strategies elaborated using this framework could then be compared with the heuristics proposed in the literature.

To demonstrate the applicability of the framework, we have used it to build a new routing algorithm. It uses a modified Bellman-Ford algorithm adapted to DTNs and asks the source to duplicate bundles. It tries to compute a routing strategy that fulfills a given delivery condition without consuming too many resources.

References

1. Zhang, Z.: Routing in intermittently connected mobile ad hoc networks and delay tolerant networks. In: IEEE Surveys and Tutorials. Volume 8-1. (2006) 24–37
2. Burleigh, S., Hooke, A., et al.: Delay-tolerant networking — an approach to interplanetary internet. IEEE Communications Magazine **41**(6) (2003) 128–136
3. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Technical Report TR CS-200006, Duke University (April 2000)
4. Juang, P., Oki, H., et al.: Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In: ASPLOS, San Jose, CA. (October 2002)
5. Spuropoulos, A., Psounis, K., Raghavendra, C.: Single-copy routing in intermittently connected mobile networks. In: Proceedings of IEEE SECON. (October 2004)
6. Spyropoulos, T., Psounis, K., et al.: Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In: Proc. of SIGCOMM'05. (2005)
7. Wang, Y., Jain, S., Martonosi, M., Fall, K.: 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) 229–236

8. Lindgren, A., Doria, A., Schelén, O.: Probabilistic routing in intermittently connected networks. SIGMOBILE Mob. Comput. Commun. Rev. **7**(3) (2003) 19–20
9. Jones, E., Li, L., Ward, P.: Practical routing in delay-tolerant networks. In: Proc. of WDTN'05, New York, NY, USA, ACM Press (2005) 237–243
10. Leguay, J., Friedman, T., Conan, V.: DTN routing in a mobility pattern space. In: Proc. of WDTN'05, New York, NY, USA, ACM Press (2005) 276–283
11. Shen, C., Borkar, G., Rajagopalan, S., Jaikaeo, C.: Interrogation-based relay routing for ad hoc satellite networks. In: IEEE Globecom, Taipei, Taiwan (November 17-21 2002)
12. Musolesi, M., Hailes, S., Mascolo, C.: Adaptive routing for intermittently connected mobile ad hoc networks. In: Proc. of WoWMoM'05. (2005) 183–189
13. Merugu, S., Ammar, M., Zegura, E.: Routing in space and time in networks with predictable mobility. Technical Report GIT-CC-04-7, Georgia Tech. Inst. (2004)
14. François, J.M., Leduc, G.: Predictable disruption tolerant networks and delivery guarantees. Technical Report (arXiv:cs.NI/0612034 v1) (2006)
15. Tan, K., Zhang, Q., Zhu, W.: Shortest path routing in partially connected Ad Hoc networks. In: Proc. of IEEE GLOBECOM'03. Volume 2. (December 2003) 1038–1042