A Survey of Message Diffusion Protocols in Mobile Ad Hoc Networks

Ahmad Al Hanbali
University of Twente
The Netherlands
hanbali@ewi.utwente.nl

Mouhamad Ibrahim
INRIA
Sophia Antipolis, France
mibrahim@sophia.inria.fr

Vilmos Simon and Endre Varga
Dept. of Telecommunications
Budapest University, Hungary
svilmos@hit.bme.hu
vendre@hit.bme.hu

Iacopo Carreras
CREATE-NET
Trento, Italy
iacopo.carreras@create-net.org

ABSTRACT

For the last twenty years, mobile communications have experienced an explosive growth. In particular, one area of mobile communication, the Mobile Ad hoc Networks (MANETs), has attracted significant attention due to its multiple applications and its challenging research problems. On the other hand, the nodes mobility in these networks has introduced new challenges for the routing protocols, especially when the mobility induces multiple disconnections in the network. In this survey, we present an overview of this issue and a detailed discussion of the major factors involved. In particular, we show how messages can be efficiently disseminated in different types of MANETs.

Keywords

Mobile computing, mobile ad-hoc networks, opportunistic communications, message delivery protocol, survey

1. INTRODUCTION

A Mobile Ad hoc Network (MANET) [8] is a self-configuring network consisting of mobile nodes that are communicating through wireless links. Nodes are free to move and the network transparently supports such movement by dynamically re-configuring itself whenever appropriate. The architecture that nodes form is fully distributed, since they don't assume any centralized network infrastructure to coordinate the communications among them, and each participating node can initiate a peer-to-peer data exchange with any other node through one-hop, or multi-hop paths.

The intrinsic distributed and self-configuring nature of this communication paradigm, combined with the ease and flexibility of deployment of such networks, make MANETs appealing for a wide range of application scenarios including, e.g., emergency situations, sensor networks for environmental monitoring [35] [57], vehicular ad hoc networks [12], and many others [13, 20].

The common denominator behind all these application scenarios is the fully distributed nature of the network infrastructure supporting them, together with the support of nodes mobility. In particular, this last characteristic is reflected in a network topology that can change over time, depending on the density and mobility of nodes. With this respect, it is possible to distinguish between more or less dense networks. The former case corresponds to networks that are connected most of the time, and for which a path almost always exists from a source to a destination. In this case, disconnections represent an exception, rather than the rule, and needs to be handled properly, although they do not represent a key requirement during the system design. To this category belong those application scenarios for which the combination of (i) nodes mobility (ii) nodes density and (iii) communication technology guarantee that the network is partitioned for relatively small time periods.

The latter case is constituted by those scenarios where the devices may be disconnected due to some physical constraint. The most evident case is interplanetary Internet [13], where the planets orbits drive the presence/absence of line-of-sight and hence the possibility of communications. Another similar case arises when we consider the use of buses and other public transportation means for carrying and disseminating information [12]. Such systems may be used, e.g., to diffuse information about traffic situation, parking availability, special events (conferences, fairs etc.), local advertisement, video-surveillance and similar tasks. In these situations, connectivity cannot be taken for granted, which determines the need to define an architecture able to handle disconnected operations. This category, generally referred as Delay Tolerant Networks (DTNs), is characterized by a disconnected network topology, and nodes use opportunistic forwarding for achieving network-wide communications .

Depending on the specific network topology characteristics

Inter-Perf 2008, October 20, 2008, Athens, GREECE
Copyright 2008 ICST ISBN # 978-963-9799-31-8.

– connected or disconnected – the way messages are diffused in the network may vary significantly. Broadcast techniques are more appropriate for the first case, where the density of nodes allows to maximally exploit the intrinsic broadcast nature of the wireless medium for reducing the number of messages being exchanged. Differently, in the case of disconnected networks, the main design goal of forwarding algorithms is shifted from performance to robustness and reliability. In this sense, it is of paramount importance to use redundancy in order to cope with the randomness of network dynamics.

Starting from this differentiation, in this survey we investigate the subtleties of various techniques that appeared in the literature, and provide a comprehensive survey of the related works. In Sec. 2, we analyze the case of broadcast diffusion techniques, classifying the different methods depending on the side-information they require, and providing a broad overview of the most utilized protocols. Sec. 3 addresses the case of opportunistic communication techniques, describing the main challenges related to the design of forwarding schemes, and detailing the most relevant methods in this area. In Sec. 4, we describe which are the main performance figures that are typically applied when evaluating diffusion processes in mobile ad hoc networks, and point out some of the most relevant models proposed in the literature. Finally, Sec. 5 concludes the survey by drawing the conclusions of this work, and by pointing out the most promising research directions that are currently being investigated.

2. BROADCAST AND DISSEMINATION PROTOCOLS

Many ad hoc applications rely on the existence of a broadcast medium for the dissemination of some control information. The naive first implementation of this was flooding: every node repeats the message after it is first received. However it was realized very soon, that this is very far from optimal, and collisions in the media can lead to serious congestion and loss of packets. To solve this problem many efficient broadcast techniques were designed, that take into account some information about their surroundings, instead of blindly repeating every packet. These algorithms differ in their assumptions about the environment (like assumption of a connected or disconnected network) and in the information available for decision (availability of Global Positioning System (GPS) for example). The central problem of broadcast algorithms is to decide when and who should retransmit messages. Nodes have to forward packets so the message reaches every part of the network, however the performance relies heavily on the set of nodes that do this. When nodes decide whether to retransmit or not, they actually decide if they are part of the forwarding set or not. Too many retransmissions cause collisions and waste the network bandwidth, but choosing the smallest forwarding set is not easy because a global view of the network is not available, and local information gets obsolete very quickly if the velocity of nodes is high. There is also a risk if the number of forwarding nodes is too small, because then the message may not reach every node. In this section we try to give an overview of the existing algorithms and approaches by giving a categorization and showing some of the interesting techniques. There are many possible ways to categorize dissemination protocols, one of the most used is in [67], which we also use as a basis, and extend it where it is needed. Giving a strict, orthogonal categorization, where an algorithm is part of exactly one class is in fact not feasible because there are many hybrid approaches that fall into many categories, and exploit different approaches simultaneously. Instead of this, we provide a usable, but not necessarily rigorous classification of algorithms. This way we can follow the conventions already established in the field. We try to show what approaches are common for dissemination protocols, and how the existing solutions relate to these approaches. In the following sections (2.1, 2.2 and 2.3) we give three classifications which capture three different aspects of dissemination algorithms. These classes are overlapping, and capture different aspects of the existing algorithms. In 2.1 we show what kind of information can be used to optimize broadcasting.

2.1 Categorization by the used information

The dissemination algorithms use different information about their environment to make their decisions. When one has to choose from the bag of existing algorithms, one has to first investigate if the needed information is available in the target network. Good examples for this are the dissemination methods that use location information. Another example can be the use of some beacon mechanism, where nodes explicitly notify others about their presence. If our network devices cannot acquire the information needed by a broadcast method, then we could not use that algorithm. However, when such an information is available, the efficiency of broadcasting can be greatly improved and the use of bandwidth can be reduced.

2.1.1 Simple heuristic based algorithms

These algorithms use very limited information about their environment. Usually they do not require periodically updated information about their neighbors, instead they watch the events of their surroundings, like successful transmissions, collisions, or the number of duplicate packets, and try to figure out whether the rebroadcast of data is needed or not. One of the most frequently used environment information is the number of received duplicates of a packet, like in the Counter Based Method [53]. Because these simple algorithms depend on heuristics, they often have some adjustable parameters, which are loosely based on the physical world, and incorporate the intuition of the designer. Determining the optimal value for these parameters is not easy. To solve this problem many of the algorithms in this category have an adaptive version, which try to figure out the optimal values for the internal parameters. Most of these algorithms are very simple, and they are usually outperformed by the more complex ones. It is also very hard to design good heuristics, that actually work in real world scenarios, too. However, in [17] the author shows, how simple learning algorithms — like a decision tree learning — are able to mimic the behavior of complex ones almost perfectly while using much simpler decision rules than the original ones. This suggests, that while good heuristics are very hard to produce "by hand" there are very good performing heuristics, that can be found by learning algorithms or genetic algorithms (or even by a combination of both).

2.1.2 Neighbor information based algorithms

These algorithms use some information about the local topology around the sender. To acquire this, these algorithms need to use periodic HELLO messages that indicate the presence of a node. These beacon messages may contain additional topology information on neighbors of the sender. Some of the algorithms collect knowledge about their immediate neighbors solely, others use k-hop information (where $k = 2$ in most of the cases). In this case, the algorithms know the local topology with higher precision, and so they can coordinate the forwarding of messages much more efficiently. Often these algorithms are sensitive to high nodes speed, because their local topology view gets outdated very quickly, so the efficiency of their forwarding policy drops. To overcome this, broadcast algorithms may choose to send topology updates more often, however this can lead to channel congestion.

2.1.3 Location based algorithms

Location based algorithms use some spatial information to make their decision. In most of the cases this means that the device should have a Global Positioning System (GPS) to acquire this information. These methods use HELLO messages, just like the neighbor information (section 2.1.2) algorithms do, but they collect the location of the neighbors too. There are also algorithms that need to know only the distance to their neighbors [39], which may be measured by signal power. In this case, the use of HELLO messages may not be necessary. The location based algorithms can perform very well (especially when combined with neighbor information), because of their very precise view of the local topology. However, the performance of these algorithms is not well understood when the error of the positioning system cannot be neglected.

2.2 Categorization by strategy

In section 2.1 we showed what kind of information can be used by the different algorithms in the literature. However this information can be processed in different ways. Again, we must emphasize that these categories are not exclusive, and some of the algorithms are put into a separate class just because they are usually discussed together in the literature.

2.2.1 Stochastic

These algorithms inhibit some intrinsic random behavior that do not come from the randomness of the environment. The benefit of this can be the elimination of coupling between the decisions of neighbors, so the mathematical properties of these algorithms may become easier to derive (this is very similar to randomization in statistics). This way, the selection of an appropriate parameter value can be supported by some mathematical results, which is a clear benefit compared to the ad hoc methods sometimes used for setting parameters. The drawback is that in some cases a random behavior may destroy some information for the neighboring nodes, as the behavior of the algorithm is no longer deterministic. Usually the behavior is adaptable to different situations by adjusting the probabilities of different decisions. Many of the stochastic algorithms are heuristic based. Some of the algorithms have some Media Access Control that introduces non-deterministic behavior. While these algorithms could be treated as stochastic algorithms, we prefer to refer to them as deterministic.

2.2.2 Deterministic

These algorithms use usually some simple information about their environment (so they usually fall also in the Simple Heuristic Based category, see section 2.1.1) and behave always the same when the environment is the same. One of the simplest examples is the Counter Based Method already discussed in subsection 2.1.1. Among the deterministic algorithms, the ones based on graph theory have a special status, and are usually discussed separately. These algorithms also have their own literature [46, 68, 18, 43], they use more solid mathematical models instead of simple ad hoc heuristics, and usually rely heavily on graph theory results. Basically two types of such algorithms exist, depending on who makes the decision about being a forwarding node or not. Self-pruning methods let every node decide to retransmit or not, while nodes using designation protocols explicitly choose which neighbors should be forwarding nodes.

The self-pruning algorithms collect neighborhood information from other nodes, and make a local decision whether to forward the message or not [68]. Nodes can decide their forwarding status when the neighborhood information is updated (on-update) or when the broadcast packet is first received (on-the-fly). The usual goal of these algorithms is to approximate a minimal connected dominating set (MCDS) [38]. A node set is a Connected Dominating Set (CDS) if every node is in the set, or is the neighbor of a node in the set. An MCDS is the smallest of the possible Connected Dominating Sets. An approximation scheme has been adopted to distributed systems in [42, 16] primarily for the use in wireless ad hoc networks.

Designation protocols on the other hand try to choose the forwarding nodes among themselves by explicitly designating a node to be a forwarder. Usually a node selects a subset of nodes from his 1-hop neighbors to be forwarding nodes for its 2-hop neighbors. The list of the designated forward nodes is usually sent with the broadcast packet [68]. The goal for neighbor designation protocols is the same as for the self-pruning protocols to approximate a minimal connected dominating set (MCDS). In [68] the authors give a generalization of the idea that incorporates most of the self-pruning and neighbor designation algorithms as special cases.

2.3 Categorization by media access

In every problem which have to be solved over wireless medium we have to deal with the problem of media access. This is a big difference from wired networks, where much of the details of the network can be abstracted away. Unfortunately wireless networks are quite ”hostile” and unresponsible abstractions of the medium can lead to poorly performing designs. Broadcast algorithms are no different, so we dedicate this subsection to discuss some of the approaches that different algorithms use.

Some methods assume Media Access to be solved by a lower layer. These algorithms usually suppose a MAC mechanism that does not use an RTS/CTS (Ready To Send/Clear To Send) handshake because this can degrade the performance of the broadcasting algorithms. Most of the algorithms use instead a random jitter to minimize collisions with other nodes. This is enough in most of the cases to avoid packet loss. However some of the algorithms use a more elaborate random backoff algorithm. One of the most used approach is called Random Assessment Delay (RAD) which was first introduced in [53]. Algorithms using a RAD mechanism do not retransmit messages immediately, but instead they wait for a random time. During this time they are able to collect more information about their environment. After the RAD expires they can cancel or proceed with the retransmission, according to the events that happened during the RAD. Some algorithms vary the length of the possible RAD period according to local congestion levels. In other cases RAD is used as a prioritization method to make some of the nodes more likely to broadcast first (usually the ones with the most neighbors). We should note that the use of RAD creates non-deterministic behavior, but we prefer to discuss this under Media Access, and not to treat the resulting algorithms as stochastic. There are also broadcast algorithms, that adjust the signal levels of the network interfaces in a coordinated way, to improve the efficiency of broadcasts. Of course these approaches need specialized equipment at the nodes.

2.4 Survey of existing algorithms

There are many published comparisons of information dissemination methods [40, 18, 67, 2, 33], which provide us a quite detailed picture about the existing broadcast approaches. In this section we will try to give a quick overview of the most important methods.

The Counter Based Method, originally introduced in [53], is one of the first controlled broadcast methods, and it is a deterministic, heuristic based algorithm. It is based on a simple observation, that if a duplicate of a packet is received, then the probability of reaching any new node is low. To exploit this idea, the nodes do not immediately transmit on the receipt of a packet, but instead they wait for a random time, which is called Random Assessment Delay (RAD). If a duplicate is received during the RAD a counter is increased. If the counter reaches a threshold before the RAD expires, the node cancels the transmission. The original method has different adaptive versions [2], which try to adapt the length of RAD and the threshold of the duplicate counter to the current network conditions.

Another very simple broadcast method is the Gossiping algorithm which was also introduced in [53]. It is a very simple one: every node broadcasts the heard message with a predefined probability. The optimal probability can be calculated off-line, or can be learned adaptively. Some of these adaptive versions are covered in [15]. While the Counter Based method is a fine example of a simple heuristic based deterministic algorithm, the Gossiping is an example for the simple heuristic based stochastic methods. While it is very easy to implement, it is usually outperformed by other more sophisticated algorithms. Another problem can be, that while the optimal retransmission probability can be calculated off line, it heavily relies on the parameters of the environment. To overcome this limitation there are adaptive versions of the basic methods, like Hypergossiping.

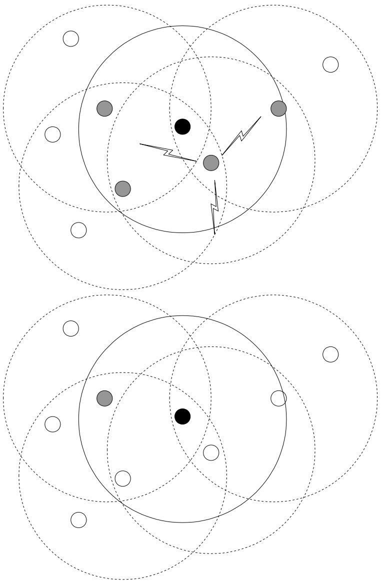
Hypergossiping [37] is one of the most recent algorithms discussed in this document. It is specifically designed for partitioned networks, where nodes are mobile, and partitions join and split from time to time. It is an advanced version of the Gossip algorithm, extended by neighbor information and partition join detection. The algorithm uses a simple adaptive gossiping strategy for in-partition forwarding, but rebroadcasts some of the packets if it detects a join with another partition. The join detection is based on the simple heuristic, that the nodes in the same partition received the same messages recently. Every node maintains a list, called LBR (Last Broadcasts Received), of the recently broadcast messages. They send HELLO messages periodically, to indicate their presence. When a new node is detected, one of the nodes includes its LBR in the next HELLO message. When the other node receives this LBR, it compares with its own LBR. If the overlap between the LBR of two nodes is smaller than a threshold, then the node is considered coming from another partition, so a new message is sent, called BR (Broadcasts Received), which contains the list of messages that the node already received. From this the other node knows that a partition join happened, and rebroadcasts all the messages that were not inside the other nodes BR. After this rebroadcast, dissemination proceeds using adaptive gossiping.

One example for location based protocols is the Optimized Flooding Protocol (OFP) which is a deterministic dissemination algorithm, that uses a geometric approach instead of the usual graph theory solutions. The algorithm tries to cover the 2D space efficiently with R radius circles. We do not detail the algorithm here, mostly because we do not think circles are good approximations of transmission ranges in urban and in-building environments. Details can be found in [62].

The Distance Adaptive Dissemination (DAD) algorithm in [39] uses distance information instead of exact positions. The authors propose a scheme that chooses forward nodes according to their distance, using the signal strength as a measure for distance. The goal of the algorithm is to try to get the outermost neighbors of a node rebroadcast, thus minimizing overlap of transmission ranges. It uses 1-hop neighbor information and records signal levels from the neighbor nodes. The authors propose two variants called DAD-NUM and DAD-PER. DAD-NUM chooses a signal strength S_{thres} so that there are k number of neighbors that have transmitted with a signal strength lower than S_{tresh}. On arrival of a new packet, the node checks if the signal strength is greater of S_{tresh} or not. If it is smaller then the node rebroadcasts. DAD-PER is very similar, but instead of finding the k farthest nodes it chooses p percent of them.

A fine example of a self-pruning algorithm is the Scalable Broadcast Algorithm (SBA) algorithm (introduced in [45]). It requires 2-hop neighbor information and the last sender ID in the broadcast packet. When a node v receives a broadcast packet from a node u it excludes the neighbors of u, $N(u)$ from the set of its own neighbors $N(v)$. The resulting set $B = N(v) - N(u)$ is the set of the potentially interested nodes. If $|B| > 0$ then the node will start a Random Assessment Delay (RAD). The maximum RAD is calculated by the $\left(\frac{d_v}{d_{max}}\right) \cdot T_{max}$ formula, where $d_v = |N(v)|$ and d_{max} is the degree of the node with the largest degree in $N(v)$, and T_{max} controls the length of the RAD. Nodes choose the time of transmission uniformly from this interval. This ensures that nodes with higher degree often broadcast packets before nodes with fewer neighbors.

The basic idea of the SBA algorithm was the RAD process, which delays transmission of packets by a random interval. The first time SBA receives a packet, it starts the RAD procedure. During this interval the SBA listens to its neighbors. When one of them starts broadcasting, SBA removes the neighbors of the broadcasting nodes from its own 1-hop neighbor set. This process is demonstrated on figure 1. The length of the RAD depends on the number of the immediate (1-hop) neighbors, so nodes with more neighbors are more



(a) Neighbor node starts broadcasting (b) SBA updates the set of interested nodes

Figure 1: Demonstration of SBA context update when neighbor node broadcasts

likely to transmit first. When the RAD ends, SBA checks if anybody remained, who might be interested in the message. If all of the neighbors are covered by another nodes, SBA cancels transmission, and the algorithm stops.

Multi-Message Scalable Broadcast Algorithm (MMSBA) is a modified version of the original SBA algorithm which is adapted to partitioned networks, and allows the dissemination of multiple messages simultaneously. One of the improvements over SBA is that MMSBA triggers a RAD not only on the first reception of a message, but on any event that changes the local neighbor information. Every time a HELLO message is heard, MMSBA updates the neighbors list. When the number of interested nodes becomes larger than zero because of the detection of a previously unseen node, MMSBA starts the RAD which works exactly the same way as in SBA. There is a little problem though. Every time a node receives a HELLO message from an unseen node, the algorithm in this form will add him to the list of interested nodes, even if it already had the broadcast message. This problem is not present in the original SBA, as nodes broadcast the message at most once, when it is first received. To overcome this problem, the nodes include the list of messages they have already received in their HELLO packets. This also gives a feedback to MMSBA if a broadcast message was lost during transmission. To support multiple messages, the RAD process also needs to be updated. When a neighbor node broadcasts MMSBA removes from its context the nodes that are interested *only* in that broadcast message. However, nodes that are interested in other messages remain in his list. This mechanism can be imagined as overlapping independent networks, where different messages are disseminated independently using the SBA RAD in the overlapping networks.

The algorithm described in [42], referred to as Wu and Li's algorithm in the literature is a self-pruning algorithm based on a marking process. First, every node is marked as gateway if it has two neighbors that are not connected to each other. To reduce this redundant Connected Dominating Set (CDS), the algorithm uses two rules to prune out unnecessary forward nodes.

Rule 1 A node v can be unmarked if it knows that there is a node u with higher priority that covers all of its neighbors.

Rule 2 A node v can be unmarked if it knows that there are u, w nodes, that are connected, have higher priority than v, and cover all of the neighbors of node v.

The algorithm does not specify how much detail is available to the nodes about their surroundings. In [18] the authors use 2-hop information to compare the performance of the algorithm with other self-pruning methods.

Stojmenovic's method [18, 43] is a variant of Wu and Li's algorithm. There are two important improvements over the original algorithm: it uses 1-hop information coupled with position information to implement the marking process and rules 1, 2. The other difference that it also implements a random backoff scheme, similar to SBA. The nodes do not broadcast immediately, but rather wait for a random time. If a node v hears a transmission during this interval from a node u then he removes $N(u)$, the neighbors of u, from its own neighbor set $N(v)$.

Multipoint relaying [64] is a neighbor designation protocol. The designated nodes that relay the messages are called Multipoint Relays (MPR). The nodes send HELLO messages to discover their 2-hop neighborhood and they try to choose as MPR the node that is able to reach most nodes among the 2-hop neighbors. The algorithm first chooses the nodes from its 2-hop neighbors that are reachable by only one node from the 1-hop neighbors, and assigns MPR status to these 1-hop neighbors. From the remaining set of 1-hop neighbors it chooses the one that covers most of the uncovered 2-hop neighbors. This step is repeated until all of the 2-hop neighbors are covered. This algorithm is also part of the Optimized Link State Routing (OLSR) Internet draft.

The Ad Hoc Broadcast Protocol (AHBP) algorithm (introduced in [46]) is another designation protocol, which is similar to Multipoint relaying but introduces some new ideas. First, designated neighbors (Broadcast Relay Gateway or BRG in AHBP terminology) are not informed in a separate HELLO message, but in the header of the broadcast data. The other difference is that when a node receives a BRG designation, it also checks which neighbors have received the message with the same transaction, and considers these nodes covered when it chooses the next hop BRGs.

A generalization of self-pruning and neighbor designation protocols was introduced first as two general rules in [68] and then specific versions of the rules were used in [18] to make a comparison with other algorithms. The algorithm is referred to as Generic Self-pruning. In its general form the method relies on k-hop neighborhood and k-hop routing information. The class of algorithms they describe use one of the versions of the so-called Coverage Condition. The most used case is when 2-hop neighbor and 2-hop routing information is used, and the self-pruning made according to the static version of Coverage Condition I[1]: Node v has a non-forwarding status

[1] Coverage Condition II is a computationally less expensive approximation of Condition I for very simple devices

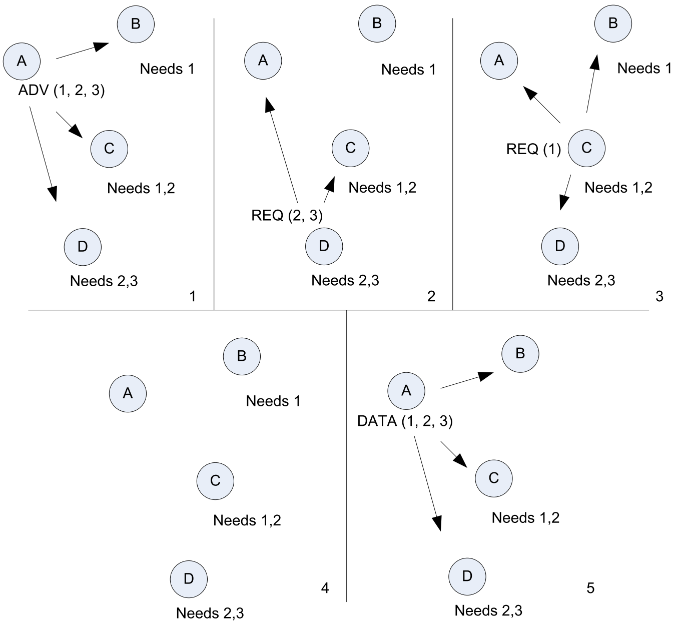


Figure 2: IOBIO handshake sequence

if for any two neighbors u and w a so called *replacement path* exists that connects u and w via several immediate nodes (if any) with either higher priority values than the priority of v or with the visited node status. Generic self-pruning contains many existing algorithms as special cases of Coverage Condition I or II (both of them are detailed in [68]), for example Lightweight and Efficient Network-Wide Broadcast (LENWB), another neighbor based self-pruning algorithm is in fact a special case of the Coverage Condition from the General Self-Pruning algorithm where the priority of the nodes are given by the number of their neighbors. It uses 2-hop neighbor and 1-hop routing information.

A quite different approach from the algorithms discussed so far is the IOBIO algorithm[63]. It is a variation of the SPIN [27] dissemination protocol. It uses a simple 3-stage handshake to discover neighbors that are interested in one of the carried messages. The goal of the protocol is to reduce the unnecessary load of neighboring nodes by duplicate or unneeded data ("spamming"). There are three IOBIO message types that are used by the protocol. The ADV (Advertisement) messages are sent periodically, and they contain the list of messages that the sending node has. Neighbor nodes indicate their interest in the advertised messages by sending a REQ (Request) packet. In response to the REQ, the originator node sends the required DATA packets. The transmission of a REQ after an ADV is not done immediately, but after a random delay. During this delay, the nodes listen to each other, and they only request packets that were not requested before. This process is demonstrated on Figure 2. Node A sends an advertisement indicating that it has the messages 1, 2 and 3. After receiving this ADV, nodes B, C, D start a random delay. At step 2 node D sends a REQ packet, indicating, that he needs messages 2 and 3. At step 3, node C sends a REQ packet. However, he heard the REQ packet of D, so he knows, that message 2 is already requested, and only puts the ID of message 1 in the REQ packet. At step 4, the random delay of B is over, however, message 1 is already requested, so no REQ is sent. At step 5 the wait interval of node A is over, and it broadcasts all requested messages.

2.5 General comparison and discussion

Table 1 summarizes the algorithms discussed in the previous sections. The main aspect of classification of dissemination protocols in dense mobile ad hoc network is the strategy used to effectively propagate information in the system. Simple heuristic based and stochastic methods are usually outperformed by the more sophisticated approaches like neighbor or location based strategies. An important constraint can be the availability of special hardware e.g. GPS devices for location based methods meanwhile most of the neighbor based schemes do not need any additional support. In dense scenarios neighbor based protocols can dramatically decrease redundancy of dissemination at the cost of increasing the overall amount of control messages. Another drawback could be the sensitivity to fast topology changes caused by high velocity nodes. A different aspect of comparison could be where is the forwarding status decided: at the node itself, or by the previous node. An exception to this is MIOBIO which uses a handshake mechanism which means that a negotiation process is carried out among the interested and forwarding nodes. Many times it is useful to be able to disseminate messages of different services parallelly, only Hypergossip, MMSBA and MIOBIO provides this feature.

3. ROUTING APPROACHES FOR INTERMITTENTLY CONNECTED NETWORKS

Intermittently connected networks are a new class of wireless networks that started to emerge recently and to gain extensive efforts from the networking research community. In the literature, these networks are found under different terminology such as sparse or extreme wireless networks, or under another commonly used term disruption/delay tolerant networks (DTNs) [1],[20]. These schemes arise in areas where the network spans over large distances with a low and heterogeneous node density and where the presence of a fixed infrastructure has no great impact on the lack of connectivity of the network. Examples of such networking scenarios include disaster healing and military networks, vehicular networks [54], deep space networks [13], communication between rural zones in toward development countries [11],[26], sensor networks for environmental monitoring [35],[57], and many other networks. Nodes participating to these networks move according to some random or particular mobility model and are generally characterized by scarce resources such as small buffer sizes, limited power and transmission capabilities. Consequently, low throughput, high end-to-end delay and high loss rates describe the default performance features of these networks.

Due to frequent partitions in these networks, instantaneous end-to-end routes do not exist between most of the node pairs, and hence most of the traditional Internet and/or mobile ad hoc routing protocols fail. However, end-to-end routes may exist over time if the nodes can take advantage of their mobility by exchanging and carrying other node messages upon meetings, and by delivering them afterward to their destinations. The latter concept have gave rise to a novel routing paradigm in these networks called the store-

Table 1: Comparison of broadcast algorithm

Algorithm	Multi-message	Forwarding Decision	Control Messages	Special Hardware	Strategy	Sensitivity on Mobility
Counter Based [53]	no	self	none	none	Simple heuristic	Speed and Connectedness
Gossiping [53]	no	self	none	none	Simple heuristic / Stochastic	Speed and Connectedness
Hypergossip [37]	yes	self	LBR and BR	none	Stochastic	Speed
OFP [62]	no	self	HELLO with location	circular radio range; GPS	Geometry based	Speed and Connectedness
DAD [39]	no	self	HELLO	signal strength measurement	Simple heuristic / Location based	Speed and Connectedness
SBA [45]	no	self	2-hop HELLO	none	Neighbor based	High speeds and Connectedness
MMSBA [56]	yes	self	2-hop HELLO with BR	none	Neighbor based	Speed
Wu and Li's algorithm [42]	no	self	2-hop HELLO	none	Neighbor based	Speed and Connectedness
Stojmenovic [18, 43]	no	self	2-hop HELLO	GPS	Neighbor based / Location based	High speeds and Connectedness
MPR [64]	no	designated	2-hop HELLO	none	Neighbor based	Speed and Connectedness
AHBP [46]	no	designated	2-hop HELLO	none	Neighbor based	High speeds and Connectedness
Generic [68]	no	self	k-hop HELLO	may use for prioritization	Neighbor based	Speed and Connectedness
LENWB [68]	no	self	2-hop HELLO	none	Neighbor based	Speed and Connectedness
MIOBIO [63]	yes	handshake	BR and REQ	none	Handshake based	Speed

carry-and-forward approach, in which the nodes will basically serve as relays for each others, thus, the term "mobility-assisted routing approach" that is used in conjunction to describe these approaches.

This part will survey and classify various research works that have considered routing schemes for intermittently connected networks. Actually, there are different ways to categorize these approaches. Hereafter, we propose a classification that is based on the degree of knowledge that the nodes have about their future contact opportunities [2] with other nodes. Specifically, depending on whether these contact opportunities are scheduled, controlled, predicted or opportunistic, these approaches can be grouped into one of the four following families.

3.1 Scheduled-contact based routing

This section surveys the routing approaches that attempt to improve the performance of a sparse network when its dynamics are known in advance such as for instance Low-earth Orbiting satellites (LEO) based networks. In a given network scenario, the most important metrics of interest are the following. The contact times between nodes (their starting times and durations), queue lengths of the nodes, and the network traffic load. The complete knowledge of these three metrics by the routing protocol allows to select optimal routes between the nodes. Despite that the implementation of the complete knowledge in a distributed environment is a very hard task, its evaluation is important as it constitutes the best case scenario compared with other case where only a partial knowledge is available to the routing protocol. On the other side, the approaches that use zero knowledge constitute the worst case scenario.

Jain et al. in [31] use the delay of a link as a cost function, and define the cost of a route to be the sum of its link costs. The authors propose four different techniques that utilize different degrees of knowledge. The first proposal is the Minimum Expected Delay (MED) where only the expectation of the link delay (excluding queueing delay) is known by the routing protocols. The second is the Earliest Delivery (ED) where the instantaneous link delay is available. The third is the Earliest Delivery with Local Queueing (EDLQ) where in addition to the use of the instantaneous delay, the delay at the local queue node is known. The last is the Earliest Delivery with All Queues (EDAQ) where in addition to the link delays, all the delays of the nodes queues are known. All these approaches were evaluated using simulation and compared to the zero knowledge and the complete knowledge cases. Their conclusion is that in networks with plentiful communication opportunities, the need for smart algorithms that require more knowledge is minimal. In situation where resources are limited smarter algorithms (EDLQ

[2] Two nodes are in contact if they are within transmission range of one another.

and EDAQ) may provide a significant benefits.

3.2 Controlled-contact based routing approaches

In this section, we discuss some routing approaches in DTNs which control the mobility of some dedicated additional mobile nodes in order to improve the network performance by increasing the contact opportunities between participating nodes. The additional mobile nodes can either have fixed predetermined paths conceived in a way to permit them to meet a large number of nodes, or their paths can be adjusted dynamically to meet traffic flows between the nodes. Their main task is to relay packets between the participating nodes by providing a store-carry-forward service. Indeed, by controlling the mobility of the additional nodes, a DTN network administrator would be able to limit the delivery delay and to provide bounds on some other performance metrics of the network. In the literature, several research works have discussed the integration of some special mobile nodes and the design of travel paths of these nodes to meet certain optimization criteria.

Jain et al. in [32] have introduced and modeled an architecture of a sparse network constituted by fixed sensors and extended by mobile nodes called *MULEs* (Mobile Ubiquitous LAN Extensions). MULEs move in the network area according to a random mobility model. Their task is to collect data from sensors, buffer it and drop it off later to a set of fixed base stations representing data sinks. The main objective of the architecture is to enhance power saving by allowing sensor nodes to exploit the random mobility of MULEs by transmitting their data to these mobile nodes over short range radio links when they pass nearby. To characterize data success ratio and queueing delay at the sensor buffer, the authors introduce a simple stochastic model based on renewal theory and bulk queueing theory. Through simulations, they have also investigated other performance metrics when the system parameters, the number of access points and the number of MULEs scale. Their basic observation confirms that an increase in the MULE density will improve system performance and leverage resource consumption.

Another controlled-contact routing work that is based on a proactive approach has been introduced in [70]-[71] by Zhao et al.. The approach is termed proactive in the sense that the trajectories of the special mobile nodes, termed as message ferries (MF), are already determined and fixed. Under the assumption of mobility of network nodes, the authors consider two schemes of messages ferries, depending on whether nodes or ferries initiate the proactive movement. In the Node-Initiated Message Ferrying (NIMF) scheme, ferries move around the area according to known routes, collect messages from the nodes and deliver the messages later to their corresponding destinations. Aware of the ferries routes, the mobile nodes can adapt their trajectories to meet the ferries in order to transmit and receive messages. In the Ferry-Initiated Message Ferrying (FIMF), the ferries will move upon service requests to meet the nodes. Specifically, when a node has packets to send or to receive, it generates a service request and transmits it to a chosen ferry using a long range radio. When the ferry receives the request, it adapts its trajectory to meet with the node for packet exchanging using short range radio.

In their former work [70], the focus was on the design of ferry routes to meet certain constraints on throughput requirement and delivery delay in networks with stationary nodes using a single ferry. By formulating the problem as two optimization sub-problems, they developed algorithms to design the ferry route. In a recent work [72], they considered the case of multiple ferries with the possibility of interaction between the ferries. The addition of multiple ferries has the advantages of improving the system performance and robustness to ferry failure at the cost of increasing the complexity of the problem. Based on several assumptions regarding whether the ferries follow the same or different routes and whether they interact with each others, they investigated four different route design algorithms that attempt to meet the traffic demand and minimize the delivery delay. Simulation results showed that when the traffic load is low, the impact of increasing the number of ferries on the delivery delay is minor. However, for high traffic load scenarios, the impact is significant.

In [14], the authors propose an algorithm called MV routing which, on one side, exploits the movement patterns of participating nodes, that is the *meeting* and *visit* behaviors, and on the other side, attempts to control the motions of some additional external nodes. Their aim is to improve network efficiency in terms of bandwidth and latency of message delivery. The algorithm is seen as being constituted by two separate mechanisms. Building on their previous work in [19], the first mechanism is a slightly modified variant of the Drop-Least-Encountered technique that is used as a routing strategy instead of a buffer management technique as it has been used in [19]. The second mechanism of the algorithm consists in adapting dynamically the movement paths of some additional nodes to meet the traffic demands in the network while optimizing some performance criterion. Travel path adjustment is carried out through multi-objective control algorithms with the objective of optimizing simultaneously several metrics related to bandwidth and delay. Simulation results demonstrate that exploiting node mobility patterns in conjunction with multi-objective control for autonomous nodes have the most significant performance improvements.

3.3 Predicted-contact based routing approaches

Predicted routing techniques attempt to take advantage of certain knowledge concerning the mobility patterns or some repeating behavioral patterns. Based on an estimation of that knowledge, a node will decide on whether to forward the packet or to keep it and wait for a better chance. Basically, each node is assigned a set of metrics representing its likelihood to deliver packets to a given destination node. When a node holding a packet meets another node with a better metric to the destination, it passes the packet to it, hence increasing the packet likelihood of being delivered to its corresponding destination. According to the nature of knowledge, we propose to reclassify the algorithms falling under this category as based on mobility-pattern or based on history.

3.3.1 Mobility-pattern based approaches

Approaches falling under this section attempt to take advantage of common behaviors of node mobility patterns in the network in order to derive decisions on packet forwarding. In fact, by letting the nodes learn the mobility pattern characteristics in the network, efficient packet forwarding decisions can be taken. Two main issues are related to these approaches. The first issue concerns the definition and the characterization of the node mobility pattern where several ways can exist to characterize and acquire such a pattern. For instance, the appearance of stable node clusters in the network, or the acquisition of statistical information related to meeting times or to the visit frequencies of nodes to a given set of locations are examples of mobility patterns that can be exploited by the nodes. The second issue is related to the way through which a node can learn and acquire its own pattern as well as those of other nodes. In particular, the presence of some external signals to the nodes such as GPS coordinates or some fixed beacons help greatly the nodes to acquire easily the mobility patterns in the network. Alternatively, nodes can also learn their own mobility patterns without any external signal by relying only on previous observations and measurements, or by exchanging pattern information with other nodes. Several routing works in DTNs that use mobility patterns to derive forwarding decisions have appeared in the literature.

In [55], the authors develop a routing algorithm that exploits the presence of concentration points (CPs) of high node density in the network to optimize forwarding decisions. The appearance of CPs is seen as the result of a general mobility model where nodes will have a high concentration inside these CPs with random movements over time between these islands of connectivity. The basic idea of their algorithm is to make use of the neighbor set evolution of each node without using any external signals. Specifically, nodes that belong to a given concentration point will collaborate between them to assign a label to their CP. Nodes will learn the labels of other nodes when they move in the network between the different CPs. Using the knowledge of the CP graph, and the positions of the source and destination nodes in the graph, the message is forwarded from the source to its destination through a sequence of CPs using the shortest path between the respective CPs. Even though the algorithm performs well, the need to manage and update the labels introduces some complexity in the algorithm mechanism.

In another work [41], the authors introduce a virtual-location routing scheme which makes use of the frequency of visit of nodes to a discrete set of locations in the network area in order to decide on packet forwarding. Specifically, they define a virtual Euclidean space, termed as *MobySpace*, where the dimension degree and the type of the coordinate space depend on the mobility pattern of the nodes. For instance, for a network with L possible node locations, the MobySpace is an n-dimensional space where $n = |L|$. Each node is represented in that space by a virtual coordinate termed as *MobyPoint*. A source node X with a message to send at time t will forward its message to a node Y among the set of its neighbors $W_X(t)$ for which the Euclidean distance to the destination is the smallest. Observe that the MobyPoint of a node is not related to its physical GPS coordinate. The acquisition of the visit frequencies of the nodes to the location set is obtained by computing the respective fraction of time of being in a given location.

Another subclass of mobility-pattern based approach consists in exploiting the underlying structure of social aspects of the network, whether in terms of contact patterns as well as set of interests, in order to derive decisions on packet forwarding. Actually, accounting for the social interactions and the social structure of the network to which the mobile users belong was proved to significantly influence the routing performance of the network. Various groups have recently started investigating the impact of social aspects on forwarding protocol design and routing performance.

In [28], the community structure behind the social interactions has been studied in order to improve the forwarding algorithms in the network. The authors showed that there exists a limited set of nodes, called *hubs*, which play a central role in the diffusion of information. Being aware of the community structure, the authors showed that an extremely efficient trade-off between resources and performance can be achieved.

In [48], the impact of different social-based forwarding schemes were evaluated on real world mobility patterns obtained from Bluetooth proximity measures. The authors showed that incorporating a friend/stranger classification in the forwarding policies can be beneficial in different application scenarios.

3.3.2 History based approaches

History based approaches are developed mainly for heterogeneous mobility movements. They rely on the observation that the future node movements can be effectively predicted based on repeating behavioral patterns. For instance, if a node had visited a location at some point in time, it would probably visit that location in another future time. Actually, if at any point in time a node can move randomly over the network area, an estimate based on previous contacts is of no help to decide on packet forwarding. However, if the mobility process has some locality, then last encounter times with other nodes can be associated with some weights that can be ranked based on their likelihood to deliver the messages to the corresponding destinations. The following works illustrate the working mechanisms of some of these approaches.

One of the pioneer work that considered history-based routing in sparse mobile networks is the work of Davis et al. in [19]. The objective of their work is to study the impact of different buffer management techniques on an extended variant of the epidemic protocol [61] on nodes with limited buffer size. Even though their work is not related to routing, the way by which the packets are sorted upon a contact influences implicitly the performance of the routing protocol. More precisely, when two nodes meet, they will first transfer the packets destined to each other, then they will exchange the lists of their remaining stored packets. The combined list of remaining packets is next sorted according to the used buffer management strategy, and each node will request the packets it does not have among the top K sorted packets. The authors have explored four different buffer management techniques, among them the *Drop-Least-Encountered* (DLE) technique which makes use of previous contacts with other nodes to decide on packet ranking. Basically, nodes using the DLE technique keep a vector indexed by addresses of other nodes where each entry estimates the likelihood of meeting the corresponding node. At each time step, a given node A updates its likelihood meeting values for every other node C with respect to the co-located node B according to the temporal difference rule (see [60]). If node A meets B, it is likely that A meets B again in the future, and hence A is a good candidate for passing the packets to B. Thus, node A should increase its likelihood for node B. If

B has a high encounter for node C, then A should increases its likelihood of meeting C by a factor proportional to the likelihood of meeting between B and C. Last, if at a given time step, node A did not meet any other node, the different likelihood values decrease in a constant rate.

In [35], the authors propose a wireless peer-to-peer networking architecture, called ZebraNet system, which is designed to support wildlife tracking for biology research. The network is basically a mobile sensor network, where animals equipped with tracking collars act as mobile nodes which cooperate between them in a peer-to-peer fashion to deliver collected data back to researchers. Researcher base stations, mounted on cars, are moving around sporadically to collect logged data. The design goal is to use the least energy, storage and other resources necessary to maintain a reliable system with a very high data delivery success rate. To attain these objectives, they propose the use of a history-based protocol to handle packet transfer between neighbor peer nodes. More precisely, each node will be assigned a hierarchical level based on its past successes of transferring data to the base station. The higher the level of the node, the higher the probability that this node is within range of base station or within range of some other nodes near the base station. Therefore, it has a high likelihood of relaying the data back to the base station either directly or indirectly through minimal number of other nodes. The mechanism works as follows: each time a node scans for peer neighbors, it requests the hierarchy level of all of its neighbors. Collected data is then sent to the neighbor with the highest hierarchy level. Whenever a node comes within range of the base station, its hierarchy level is increased while it is decreased over time at a given rate when it is out-of-range.

Jones et al. in [34] propose a variant of the MED approach of [31] called Minimum Estimated Expected Delay (MEED). Alternatively to the MED approach where the expected delay of a link is computed using the future contact schedule, MEED uses an estimation of the observed contact history. The estimator implements a sliding history window with an adjustable size. To minimize the overhead induced by the frequent updates of the estimated link delay, the authors propose to filter update samples having small difference with the actual information in the network. Through simulations, MEED has shown to overcome the performance of MED as it is more responsive to network changes, and its performance approaches that of the epidemic protocol. However, the algorithm lacks the presence of an adjustment mechanism of its window size.

The authors in [44] propose PROPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity), a single copy history-based routing algorithm for DTNs. Similarly to [19], each node in PROPHET will attempt to estimate a delivery predictability vector containing an entry for each other node. For a given node X, the entry $P(X, Y) \in [0, 1]$ will represent the probability of node X to deliver a message to a given node, for instance node Y in this case. The entries of the predictability vectors will be used to decide on packet forwarding. Specifically, when two nodes meet, a message is forwarded to the other node if the delivery predictability for the destination of the message is higher at the other node. In addition to the predictability vector, a summary vector of stored packets will be also exchanged upon contact. The information in the summary vector is used to decide on which messages to request from the other node. The entry update process occurs upon each contact and works as follows. Nodes that are often within mutual ranges have a high delivery predictability for each other, and hence they will increase their corresponding delivery predictability entries. Alternatively, nodes that rarely meet are less likely to be good forwarders of messages to each other, and hence they will reduce their corresponding delivery predictability entries.

3.4 Opportunistic-contact based routing approaches

Opportunistic based approaches are generally characterized by random contacts between participating nodes followed by potential pair-wise exchanges of data. Given that connectivity, and consequently, data exchanges are subject to the characteristics of the mobility model which are in general unpredicted, these approaches rely on multi-copy schemes to speed up data dissemination within the network. In the following, we subdivide these approaches into epidemic-based approaches and coding based approaches.

3.4.1 Epidemic based approaches

Epidemic based approaches imitate the spread of contagious disease in a biological environment. Similarly to the way an infected individual passes on a virus to those who come into contact, each node in an epidemic-based system will spread copies of packets it has received to other susceptible nodes. The number of copies that an infected node is allowed to make, termed as the fan-out of the dissemination, and the maximum number of hops that a packet is allowed to travel between the source and the destination nodes, represented by a hop count field in the packet, define the epidemic variant of the algorithm. These two parameters can be tuned to trade delay for resource consumption. Clearly, by allowing the packet to spread throughout the mobile nodes, the delay until one of the copies reaches the destination can be significantly reduced. However, this comes at the cost of introducing a large overhead in terms of bandwidth, buffer space and energy consumption. Several variants of epidemic-based approaches have been proposed and their performance in terms of delay and resource consumption have been evaluated.

One of the pioneer work in this domain is the epidemic routing protocol of Vahdat and Becker [61]. The protocol is basically a flooding mechanism accommodated for mobile wireless networks. It relies on pair-wise exchanges of messages between nodes as they get in contact with each other to eventually deliver the messages to their destinations. Each node manages a buffer containing messages that have been generated at the current node as well as messages that has been generated by other nodes and relayed to this node. An index of the stored messages called a summary vector is kept by each node. When two nodes meet, they will exchange their summary vectors. After this exchange, each node can determine then if the other node has some messages that was previously unseen by it. In this case, it will request the missing messages from the other node. To limit the resource utilization of the protocol, the authors propose to use a hop count field at each message that specifies the total number of epidemic exchanges that a particular message may be subject to. They showed that by appropriately choosing the maximum hop count, delivery rates can still be kept high while limiting resource utilization.

In [24], Grossglauser and Tse introduce a one copy two-hop relay protocol. Basically, at any time, they will be one copy of the packet in the network, however, the copy can make at most two hops between the source node and the destination node. Their packet dissemination algorithm can be seen as an epidemic-like protocol with a fan-out of one and a hop count of two. The key goal of their work is to show that the capacity of a mobile network can scale with the number of nodes by exploiting the mobility of these nodes through a two-hop relay protocol.

Building on [61] and [24], several research works have appeared subsequently which proposed analytical models to evaluate the performance of these protocols. In [23] Groenevelt et al. introduce a multicopy two-hop relay protocol (MTR), a variant of the two-hop relay protocol. In MTR, the source forwards a copy of the packet to any other relay node that it encounters. Relay nodes are only allowed to forward the packets they carry to their destinations. By modeling the successive meeting times between any pair of mobile nodes by Poisson processes, the authors characterize the distribution of the delivery delay and that of the total number of copies generated until the packet delivery. This work was extended in [5] by Al Hanbali et al. under the assumption of limited lifetime of the packets, and in [4] under the assumption of general distribution of inter-meeting times. Zhang et al. in [69] extend the work in [23] by evaluating several variations of the epidemic protocol and some infection-recovery schemes. Inspired by [23], the authors of [29] consider a sparse mobile ad hoc network equipped by throwboxes. Throwboxes are small and inexpensive wireless devices that act as fixed relays and that are deployed to increase contact opportunities among the nodes. By modelling the meeting times between a mobile and a throwbox as a Poisson process, the authors characterize the delivery delay and the total number of copies generated under the MTR and the epidemic protocol for the cases where throwboxes are fully disconnected or mesh connected.

A biological acquisition system termed as the shared wireless infostation model (SWIM) has been introduced in [57] as a way of routing collected measurement traces between a set of sensors attached to whales and a set of fixed infostations acting as collecting nodes. Infostations act as base stations which connect the users to the network. Mobile nodes represented by the tagged whales move randomly within the area and connect to the infostations when they are within range to offload their data. When two tagged whales meet, an epidemic exchange mechanism takes place in order to accelerate the delivery of the packets at the cost of increasing the storage space at the nodes. Through simulations, the authors showed that sharing the data among the whales as well as increasing the number of SWIM stations reduce significantly the end-to-end delay. The positions of infostations as well as the mobility of whales greatly affect the system performance.

Spyropoulos et al. introduce a new routing algorithm for sparse networks in [59], termed Spray and Wait algorithm. The algorithm disseminates a number of copies of the packet to other nodes in the network, and then waits until one of these copies meets the destination. It consists of two phases. In the first phase, the source node will generate a total of L copies of the message it holds, then spreads these copies to other nodes for delivery to the destination node. The spreading process works as follows. When an active node holding $n > 1$ copies meets another node, it hands off to it $F(n)$ copies and keeps for itself the remaining $n-F(n)$ copies and so forth until a copy of the message reaches the destination. F is the function that defines the spreading process. For instance, for binary spray and wait, $F(n) = \frac{n}{2}$. In the second phase, the wait phase, if the destination is not found among the L copy-carrying nodes, then these latter nodes will perform direct transmissions to the destination node. Using simulations, the authors show that this technique can achieve a trade-off between efficient packet delivery and low overhead if the parameters are carefully designed.

3.4.2 *Coding based approaches*

The approaches in Section 3.4.1 are primarily based on packet flooding in order to improve the efficiency of packet delivery. Unfortunately, these improvements come at the expense of introducing large overhead in the network due to redundant packet transmissions. The approaches presented in this section alleviates the effect of flooding through the use of smarter redundant algorithms that are based on coding theory. In the following, we consider two main coding algorithms that appeared in the literature and which have shown their suitability to the opportunistic contact networks, namely the erasure coding and the network coding.

In the erasure coding scheme, upon receiving a packet of size m, the source produces n data blocks of size $l < m$. The coding algorithm composes these blocks in a such way to allow the destination to retreive the original message on receiving any subset of these blocks [49]. More precisely, the transmission of the packet is completed when the destination receives the kth block, regardless of the identity of the $k \approx m/l < n$ blocks it has received. The blocks are forwarded to the destination through the relay nodes according to store-carry-and-forward approach. The performance analysis of this approach in opportunistic contact network has shown to improve significantly the worst case delay with fixed amount of overhead [4, 65]. Further, in [30] it has been shown that erasure coding improve the probability of packet delivery in DTNs with transmissions failures.

In the network coding scheme, instead of simply forwarding the packets, nodes may transmit packets with linear combinations of previously received ones. For example, consider the three nodes case where nodes A and C want to exchange packets via the intermediate node B. A (resp. C) sends a packet a (resp. c) to B, which in turn broadcasts $a\ xor\ c$ packet to A and C. Both A and C can recover the packet of interest, while the number of transmissions is reduced. In [66], different aspects of the operability of network coding with limited storage resources have been discussed and different techniques have been proposed. The main result is that network coding benefits more from node mobility and performs well in scenarios of high packet drop rate where simple flooding approaches fail.

3.5 General comparison and discussion

Table 2 compares the various proposals that have been addressed in Section 3 by summarizing the main distinguishable features of each one. Our comparison is based on four features. The first feature defines the degree of knowledge that the nodes have about their future contact opportunities. Future contact opportunities are identified as being scheduled, controlled, predicted or opportunistic. The second feature lists the key relevant performance metrics that

Table 2: Summary of the routing approaches in DTNs and their main properties.

Proposal	Contact opportunities	Metric to optimize	Mobility pattern of network nodes	Mobility pattern of special nodes
Jain et al. protocol [31]	Scheduled	Delay	Random	–
MULE protocol [32]	Controlled	Power usage, Buffer overhead	Stationary	Random
MF protocol [70]	Controlled	Delivery rate, Power usage	Stationary	Predetermined paths
Extended MF protocol [71]	Controlled	Delivery rate, Power usage	Random, Stationary	Predetermined, Dynamic paths
MV protocol [14]	Controlled	Delivery rate, Delay	Meeting and Visit dependant	Metric dependant paths
Island hopping protocol [55]	Predicted	Delay, Transmission overhead	Heterogeneous mobility	–
MobySpace protocol [41]	Predicted	Delivery rate, Power usage	Location dependant	–
DLE protocol [19]	Predicted	Buffer usage	Heterogeneous mobility	–
ZebraNet [35]	Predicted	Delivery rate, Power, Storage	Heterogeneous mobility	–
MEED protocol [34]	Predicted	Delay, Transmission overhead	Random	–
PROPHET [44]	Predicted	Delivery rate, Power usage	Heterogeneous mobility	–
Epidemic protocol [61]	Opportunistic	Delivery ratio, Delay	Random	–
Two-hop protocol [24]	Opportunistic	Network capacity	Random	–
MTR protocol [23]	Opportunistic	Delay, Transmission overhead	Random	–
SWIM protocol [57]	Opportunistic	Delivery rate, Delay	Random, Stationary	–
Spary and Wait [59]	Opportunistic	Delivery rate, Power usage	Random	–
Erasure coding [65]	Opportunistic	Delivery rate	Random	–
Network coding [66]	Opportunistic	Delivery rate	Random	–

each proposal attempts to optimize. For instance, these metrics range from increasing the packet delivery ratio to reducing the end-to-end delivery delay, energy consumption and/or buffer occupancy of the nodes. The third and fourth features list the characteristics of the mobility patterns of the network nodes and the dedicated special nodes, whenever employed. Precisely, nodes of the network could be stationary where in this case they are the special nodes that move around according to some predetermined or dynamic paths to assist in packet routing to the fixed nodes. Alternatively, node mobility pattern could be either random, where there is no means to predict the potential future contacts of a node, or heterogeneous with some location dependency or some correlated meeting among the nodes. Observe that the properties we have listed are not exhaustive and other properties can be included in addition. For instance, the complexity of the proposal in terms of implementation or computation, or the requirement to exchange some control information can also be considered. However, we restricted the comparison to the previous four features, which we think are the most relevant according to the classification that we made before.

4. MODELLING APPROACHES

In the absence of predictable mobility and network topology, the notion of "route" for a message to follow looses its significance, and it becomes imperative to employ some kind of "epidemic spreading" mechanisms [36, 69]. Depending on the application scenario considered, such epidemic spreading can occur over large periods of time, as in the case of sparse networks where the delivery of messages is obtained from the physical mobility of nodes, or shorter ones, where the dense nature of the network allows to exploit the use of broadcasting algorithms. In both cases, it is of paramount importance to use redundancy in order to cope with the randomness of network dynamic. At the same time, forwarding operations rely on the ability of a node to keep (even for a rather long time) a message in its internal memory. This is justified by the fact that a node may be doomed to remain isolated for a long time, but should still be able to forward the messages it received. In this sense, the redundancy encompassed by the algorithm stresses the existence of a performance/robustness vs. storage/energy consumption tradeoff. Indeed, the larger the number of copies of a message in the system, (i) the faster it reaches its destination (ii) the more it is robust with respect to the nodes mobility and node/link failures. On the other hand, in order to have more copies of the same message traveling in the network at the same time, a larger amount of network resources has to be exploited. Resources are intended in terms of both (i) storage, necessary to keep the message in the nodes' memory for a longer time (ii) energy consumption, in that a larger number of transmissions of the same message is needed.

From these considerations emerges how the performance of message diffusion in MANETs is always a trade off between different requirements, and single aspects can not be considered in isolation. As an example, end-to-end delay should always be considered as a function of the resources, e.g., storage, allocated to run a specific forwarding or broadcasting algorithm.

It also clear the need to perform, where possible, an accurate modeling of the system and of the various processes occurring in the network, able to efficiently account for the various system parameters and to provide useful insights into the design space of such systems. This need is confirmed by the many models and modeling techniques appeared in the literature over the past years.

The traditional store-and-forward routing protocols, which require the existence of a connected path between a source and a destination, do not achieve good performance in intermittently connected ad hoc networks. A solution for this problem is to exploit the mobility of nodes present in the network. Such an approach is known as store-carry-and-forward and it has been proposed in the pioneering paper of Grossglauser and Tse [24].

The important aspects in the store-carry-and-forward solutions are the so-called contact opportunity and inter-contact time between nodes that mainly depend on the mobility of the nodes. In the following we will first introduce the performance metrics of interest before surveying the performance evaluation tools used in the literature. We should emphasize that most of the performance models developed in the literature focus on the opportunistic networks in Section 3. The key performance metrics in intermittently-connected networks are the following: **(i)** the network throughput known also network capacity, **(ii)** the delivery rate defined as the percentage of packets that successfully reach the destination, **(iii)** the packet delay denoted as the time that a packet requires to reach the destination, **(iv)** the energy consumption of the network in order to deliver a packet to its destination. The latter metric is especially important for the multicopy relay protocols that belong to the opportunistic class in Section 3.

A significant research work spawned exploring the tradeoffs between the capacity and the delay of the two-hop relay protocol and other similar schemes, especially their scaling laws when the number of nodes is large [21, 22, 24, 25, 51]. It is important to mention that most of these studies assume a uniform spatial distribution of nodes, which is the case, for example, when nodes perform a symmetrical Random-Walk over the region of interest [22], or when nodes move according to the Random Direction model [50]. Using a queueing analysis the authors in [3] prove that the uniform mobility models achieves the minimal relay throughput as compared with non-uniform models such as the Random-Waypoint model [9]. On the other hand, the authors in [23] show that the distribution of the inter-meeting times between any mobile nodes pair is approximately an exponential distribution. This finding has been noticed for a number of mobility models (Random Walk, Random Direction, Random Waypoint) in the case when the node transmission range is small with respect to the area where the nodes move. Exploiting this property, a batch of Markovian models of the number packet copies has been proposed recently in the literature to evaluate the delay and the energy consumption of a class of multicopy relay protocols, e.g. MTR, Epidemic Routing, for both the cases of finite and infinite number of nodes [5, 4, 23, 29].

Another tool that was used to evaluate the performance of multicopy relay protocols is the so-called fluid approach also know as the mean field approach. In disconnected mobile networks the fluid quantity represents the mean number of packet's copies in the network. The dynamics of these quantities in time can be written as a set of ordinary differential equations (ODEs). Using this tool Small and Haas in [57] provide a model, to evaluate the performance of disconnected mobile networks embedded in an infostation network architecture. They consider the case where the Epidemic Routing protocol is used to relay data from the mobile nodes to the infostations. An infostation can be seen as a wireless access port to the Internet or to some private networks. Zhang et al. in [52] extend the work in [57] and showed that the ODEs can be derived as limits of Markovian models under a natural scaling as the number of nodes increases. Moreover, they studied variations of the Epidemic Routing protocol, including probabilistic routing and recovery infection schemes.

Once the performance metrics of interest are computed, e.g. the expected delay and the expected energy consumed, one can construct a number of optimization problems. To this end, certain metrics should be first parameterized such as the maximal number of packet transmissions or the maximal number of packet copies in the case where packet's copies have limited lifetime. Based on this idea the authors in [59, 58] proposed to limit the maximal number of packet's copies of forwarding protocols using token based solution. Building on these studies Neglia and Zhang in [52] identify the best policy that a node should employ in order to minimize the linear cost function of the expected delay and the expected energy consumption. This is done with the help of the Dynamic Programming theory with a centralized controller.

In the case of dense ad hoc networks most of the modeling approaches of epidemics can not be applied. The most important limitation is that mobility can not be modeled by exponential intermeeting times because of the significant probability of having a node already in range. Also meetings can not be assumed pairwise anymore because small connected islands can be formed time to time. Because of connectivity the dissemination delays inside islands are much lower than in DTNs, therefore if the movement speed of nodes is small (pedestrian) the underlying connection graph can be assumed to be fixed during the dissemination in the island. The connections will change significantly only in the timescale of island intermeeting times. This naturally leads to several graph-theory based approaches. One significant

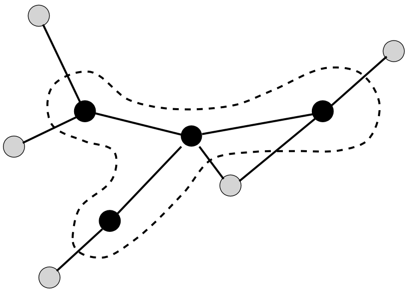


Figure 3: A Minimum Connected Dominating Set in a graph

use is to model the connection between nodes with Unit Disk Graphs (UDG) [10]. A UDG is constructed by placing unit radius disks on the plane associating a vertex to each circle and connecting the vertices if the corresponding disks overlap. It was shown by Breu and Kirkpatrick in [10] that the problem of deciding that a given graph is a Unit Disk Graph is NP hard.

Random UDGs share many of the properties of Bernoulli random graphs [6, 7]. The most important property of UDGs however is that many hard optimization problems on graphs can be approximated effectively on UDGs [47]. These problems include the approximation of a Maximum Independent Set, Minimum Dominating Set and Minimum Connected Dominating Set [38, 42, 16] (MCDS) that has very important applications for multi-hop broadcasting. A set of vertices is called a Connected Dominating Set if every vertex is in the set or has a neighbor in the set and the vertices of the set form a connected subgraph. An MCDS is the smallest of the possible Connected Dominating Sets. Figure 3 shows an example of an MCDS. The cardinality of an MCDS is a lower bound on the number of transmissons that are needed to disseminate a message in a connected island, therefore many algorithms try to approximate an MCDS in a distributed way.

5. CONCLUSION

In this survey, we have investigated several techniques for packet dissemination in mobile ad hoc networks. By referring to the type of applications which these techniques are designed for, we have categorized them into two generic classes where the first class includes reliable dissemination mechanisms using broadcast as a central means for packet delivery while the second class includes techniques that are designed for networks tolerating high delivery latency where store-carry-and-forward paradigm is the commonly used mechanism. For each class, we have reviewed a large part of recent research works that have appeared and proposed further categorizations of the different techniques according to some distinguishing features.

6. ACKNOWLEDGMENTS

This work of Vilmos Simon, Endre Varga and Iacopo Carreras, has been supported by the European Commission within the framework of the BIONETS project IST-FET-SAC-FP6-027748, www.bionets.eu.

7. REFERENCES

[1] Delay tolerant networks (DTN) research group. http://www.dtnrg.org.

[2] Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. In *Proc. of the ICDCS*, Washington, DC, USA, 2001. IEEE Computer Society.

[3] A. Al Hanbali, A. A. Kherani, R. Groenovelt, P. Nain, and E. Altman. Impact of mobility on the performance of relaying in ad hoc networks- extended version. *Elsevier Computer Networks*, 51(14):4112–4130, Oct. 2007.

[4] A. Al Hanbali, A. A. Kherani, and P. Nain. Simple models for the performance evaluation of a class of two-hop relay protocols. In *Proc. of IFIP Networking*, Atlanta, GA, USA, May 2007.

[5] A. Al Hanbali, P. Nain, and E. Altman. Performance of ad hoc networks with two-hop relay routing and limited packet lifetime. In *Proc. of IEEE/ACM ValueTools*, Pisa, Italy, Oct. 2006.

[6] M. J. Appela and R. P. Russob. The connectivity of a graph on uniform points on $[0, 1]^d$. *Statistics and Probability letters*, 60:351–357, 2002.

[7] S. R. Ashish Goel and B. Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. *ACM Symposium on Theory of Computing*, 2004.

[8] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovi. *Mobile Ad Hoc Networking.* IEEE Press John Wiley, 2004.

[9] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. *ACM/Kluwer Wireless Networks, Special Issue on Modeling and Analysis of Mobile Networks*, 10(5):555–567, Sept. 2004.

[10] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry. Theory and Applications*, 9(1-2):3–24, 1998.

[11] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall. The case for technology in developing regions. *IEEE Computer*, 38:pp. 25–38, May 2005.

[12] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. of IEEE INFOCOM*, April 2006.

[13] 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*, 41:pp. 128–136, June 2003.

[14] B. Burns, O. Brock, and B. Levine. MV routing and capacity building in disruption tolerant networks. In *Proc. of IEEE INFOCOM*, Miami, Florida, USA, Mar. 2005.

[15] J. Carle, J. Cartigny, and D. Simplot. Stochastic flooding broadcast protocols in mobile wireless networks. Technical Report 2002-03, LIFL Univ. Lille 1, France, May 2002.

[16] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[17] M. D. Colagrosso. Intelligent broadcasting in mobile ad hoc networks: Three classes of adaptive protocols. *EURASIP Journal on Wireless Communications and Networking*, 2007. Article ID 10216.

[18] F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):1027–1040, 2004.

[19] J. Davis, A. Fagg, and B. Levine. Wearable computers as packet transport mechanisms in highly partitioned ad hoc networks. In *Proc. of 5 IEEE Intl. Symp. on Wearable Computers*, Zurich, Switzerland, 2001.

[20] K. Fall. A delay tolerant network architecture for challenged internets. In *Proc. of ACM Sigcomm*, Karlsruhe, Germany, Aug. 2003.

[21] R. M. G. Sharma and N. Shroff. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. In *Proc. of IEEE Infocom*, Barcelona, Spain, Apr. 2006.

[22] A. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *Proc. of IEEE INFOCOM*, Hong Kong, Mar. 2004.

[23] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62(1-4):210–228, Oct. 2005.

[24] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *ACM/IEEE Transactions on Networking*, 10(4):477–486, Aug. 2002.

[25] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, Mar. 2000.

[26] A. A. Hasson, D. R. Fletcher, and D. A. Pentland. A road to universal broadband connectivity. In *Proc. of Development by Design*, Dec. 2002.

[27] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. MOBICOM*, Seattle, 1999.

[28] P. Hui and J. Crowcroft. Bubble rap: Forwarding in small world dtns in ever decreasing circles. Technical Report UCAM-CL-TR-684, Univ. of Cambridge, Computer Laboratory, May 2007.

[29] M. Ibrahim, A. Al Hanbali, and P. Nain. Delay and resource analysis in manets in presence of throwboxes. *Performance Evaluation*, 64(9-12):933–947, Oct. 2007.

[30] S. Jain, M. Demmer, R. Patra, and K. Fall. Using redundancy to cope with failures in a delay tolerant network. In *Proc. of ACM Sigcomm*, Philadelphia, PA, USA, Aug. 2005.

[31] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant networking. In *Proc. of ACM Sigcomm*, Portland, OR, USA, Aug. 2004.

[32] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy-efficient data collection in sensor networks. In *IEEE WiOpt*, 2004.

[33] T. Jin, K. Yunjung, and M. G. Yi. Efficient flooding in ad hoc networks: a comparative performance study, 2003.

[34] E. P. Jones, L. Li, and P. A. Ward. Practical routing in delay-tolerant networks. In *Proc. of ACM Sigcomm Workshop on Delay Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.

[35] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *Proc. of ASPLOS-X*, San Jose, CA, Oct. 2002.

[36] A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in manets. In *Proc. of MSWiM*, pages 54–60, New York, NY, USA, 2002. ACM Press.

[37] A. Khelil, P. J. Marrón, C. Becker, and K. Rothermel. Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. *Ad Hoc Netw.*, 5(5):531–546, 2007.

[38] S. Khuller and S. Guha. Approximation algorithms for connected dominating sets. In *European Symposium on Algorithms*, pages 179–193, 1996.

[39] S. Krishnamurthy, X. Chen, and M. Faloutsos. Distance adaptive (dad) broadcasting for ad hoc networks. In *Proc. of MILCOM*, Oct. 2002.

[40] T. O.-N. Larsen, T. H. Clausen, and L. Viennot. Investigating data broadcast performance in mobile adhoc networks. In *Proc. of WPMC*, Oct. 2002.

[41] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for DTNs. In *Proc. of IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[42] H. Li and J. Wu. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of DiaLM*, 1999.

[43] X. Lin and I. Stojmenovic. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks, 2001.

[44] A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. In *Proc. of ACM MobiHoc (poster session)*, Maryland, USA, June 2003.

[45] X. Lu and W. Peng. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proc. of Mobihoc*, 2000.

[46] X. Lu and W. Peng. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Science and Technology*, 2002.

[47] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Geometry based heuristics for unit disk graphs, 1994.

[48] A. G. Miklas, K. K. Gollu, S. Saroiu, K. P. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *Proc. of UBICOMP*, Innsbruck, Austria, Sept 2007.

[49] M. Mitzenmacher. Digital fountains: A survey and look forward. In *Proc. of IEEE Information Theory Workshop*, Texas, USA, Oct. 2004.

[50] P. Nain, D. Towsley, B. Liu, and Z. Liu. Properties of random direction models. In *Proc. of IEEE Infocom*, Miami, FL, Mar. 2005.

[51] M. J. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. *IEEE Transactions on Information Theory*, 51(6), June 2005.

[52] G. Neglia and X. Zhang. Optimal delay-power tradeoff

in sparse delay tolerant networks: a preliminary study. In *Proc. of CHANTS*, Pisa, Italy, Sep. 2006.

[53] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. of MobiCom*, pages 151–162, New York, NY, USA, 1999. ACM.

[54] J. Ott and D. Kutscher. Drive-thru internet: IEEE 802.11b for automobile users. In *Proc. of IEEE INFOCOM*, Hong Kong, Mar. 2004.

[55] N. Sarafijanovic-Djukic, M. Piorkowski, and M. Grossglauser. Island hopping: Efficient mobility-assisted forwarding in partitioned networks. In *IEEE SECON 2006*, Reston, VA, Sept. 2006.

[56] V. Simon, L. Bacsï¿½rdi, S. Szabï¿½, and D. Miorandi. Bionets: A new vision of opportunistic networks. In *Proc. of IEEE WRECOM*, 2007.

[57] T. Small and Z. J. Haas. The shared wireless infostation model: A new ad hoc networking paradigm. In *Proc. of ACM MobiHoc*, Annapolis, MD, USA, June 2003.

[58] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *Proc. of ACM Sigcomm Workshop on Delay Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.

[59] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proc. of ACM Sigcomm Workshop on Delay Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.

[60] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[61] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[62] R. J. Vamsi, K. Paruchuria, and A. Durresib. Optimized flooding protocol for ad hoc networks.

[63] E. Varga, T. Csvorics, L. Bacsï¿½rdi, M. Berces, V. Simon, and S. Szabï¿½. Novel information dissemination solutions in biologically inspired networks. In *Proc. of ConTEL*, Zagreb Croatia, Jun. 2007.

[64] L. Viennot, A. Qayyum, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical report, INRIA, 2000.

[65] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. In *Proc. of ACM Sigcomm Workshop on Delay-Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.

[66] J. Widmer and J.-Y. Le Boudec. Network coding for efficient communication in extreme networks. In *Proc. of ACM Sigcomm Workshop on Delay-Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.

[67] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proc. of MOBIHOC*, pages 194–205, 2002.

[68] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. *Int. J. Found. Comput. Sci.*, 14(2):201–221, 2003.

[69] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *Proc. of Networking*, pages 827–839, Coimbra, Portugal, May 2006.

[70] W. Zhao and M. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *Proc. of the IEEE Workshop on Futrure Trends in Distributed Computing Systems*, Puerto Rico, May 2003.

[71] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. of ACM Mobihoc*, Tokyo, Japan, May 2004.

[72] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proc. of IEEE INFOCOM*, Miami, Florida, USA, Mar. 2005.