Utility-based Message Replication for Intermittently Connected Heterogeneous Networks

Thrasyvoulos Spyropoulos, Thierry Turletti, Katia Obraczka
INRIA, Sophia-Antipolis
{Thrasyvoulos.Spyropoulos,Thierry.Turletti,Katia.Obraczka}@sophia.inria.fr

***Abstract*— Communication networks (wired or wireless) have traditionally been assumed to be connected at least most of the time. However, emerging applications such as emergency response, special operations, smart environments, VANETs, etc. coupled with node heterogeneity and volatile links (e.g. due to wireless propagation phenomena and node mobility) will likely change the typical conditions under which networks operate. In fact, in such scenarios, networks may be mostly disconnected, i.e., most of the time, end-to-end paths connecting every node pair do not exist. To cope with frequent, long-lived disconnections, *opportunistic routing* techniques have been proposed in which, at every hop, a node decides whether it should either forward and/or store-and-carry a message. As a result, a number of message replicas may be created and routed independently ("spraying"). Most opportunistic routing schemes to-date perform *greedy* replication handing over a copy of a message to the first nodes encountered. Yet, in a network with heterogeneous nodes, where some nodes may be much "better" relays than others, such greedy schemes waste a lot of message replicas (and thus energy, storage space, etc.) on "useless" relays. For this reason, we propose the idea of *utility-based replication*, where some *fitness* or *utility* function is maintained for all nodes in a distributed fashion, and a small budget of message replicas is allocated according to this utility only to the fittest nodes. We describe a number of variations using different utility functions, and show that an improvement of up to $5-6\times$ in delay can be achieved over greedy algorithms.**

I. Introduction

The traditional view of networks as a connected graph over which end-to-end paths need to be established might not be appropriate for modeling existing and emerging wireless networks. Due to wireless propagation phenomena, node mobility, low power nodes shutting down, etc., connectivity in wireless networks is more often than not intermittent. Under such intermittent connectivity many traditional routing protocols, both table-driven (e.g. based on link-state routing) and reactive ones (e.g. DSR, AODV), would have their performance deteriorate drastically as connectivity becomes increasingly sporadic and short-lived.

To perform routing under "episodic" connectivity, researchers have proposed a variety of *opportunistic* routing schemes [1], [2], [3], where: (i) a message may be *stored* and *carried* by a node for long periods of time, until a communication opportunity arises ("mobility-assisted"), (ii) local forwarding decisions are made independently with the goal that a message will *eventually* be delivered ("opportunistic"), and (iii) multiple copies of the same message may be propagated in parallel in the above manner ("replication").

One of the schemes proposed that has shown promising performance is that of "Controlled Replication" or "Spray and Wait" [4], [5], [3]. This scheme distributes a *small* and *controlled* number of message replicas *to the first few nodes encountered* (we'll be referring to this basic algorithm as "greedy" controlled replication, or simply "greedy replication", hereafter); then, each of the nodes that received a copy in this first phase is only allowed to give it to the destination itself[1]. There are two desirable characteristics connected with this approach: (i) it consumes much fewer resources than epidemic routing [7] and its variations [8], [9] with sometimes little or no penalty on performance [5], [3], and (ii) due to its simplicity, it can be *controlled* to achieve the desired performance trade-off (resources used vs. delivery delay/probability), even in an almost unknown environment, by appropriately choosing the number of copies [3].

In a network where most nodes are highly "mobile" and make decisions independently (e.g. VANETs [10][2]), greedy replication could be a simple and efficient-enough solution; there, each relay carrying a message copy encounters new nodes quickly, and picking just a few (even randomly) could create enough redundancy to ensure one of them will encounter the destination soon. On the other hand, imagine a scenario where the majority of nodes tend to spend most of their time with the same nodes (e.g. employees on the same group or floor, animals in the same herd or family [9]), while only a small number of nodes tend to move often between disconnected parts of the network (e.g. vehicles, nodes with a more "social" daily routine, etc.). In such a scenario, handing over copies *to the first few nodes encountered*, implies that some or all of these copies will end up with nodes that may never encounter the destination. In general, *when nodes in a network are heterogenous in terms of their ability to deliver a message, greedy replication may fail to discover the "better" relays, especially if the latter are much less common than the "not so good" ones.*

To deal with this situation, it has been proposed to allow an initially chosen relay to hand-over its message replica to another node, if it is deemed that the latter one has a higher chance of encountering the destination (according to some agreed "utility" measure [3], [11]). This scheme can discover better and better relays than the set of the initially chosen ones, and using a gradient-based approach can deliver a message to the destination faster. Although it still uses the same number of replicas per message, each replica may be transmitted more than once, and may even loop around due to the dynamic nature of the utility function maintained. What is more, unlike the case of controlled replication, it is significantly more difficult to analytically predict the number of transmissions and expected delay for this scheme [3].

[1]In this sense, it is essentially an extension to the 2-hop scheme proposed by Grossglauser et. al, where at most 1 relay is used [6].

[2]Due to street limits, traffic considerations, etc. drivers do not in general make decisions independently; yet, to the extent that two drivers will decide for example whether to take a turn or not in the next cross-street, this decision will be largely independent.

Summarizing, although allowing multi-hop forwarding of each copy can improve the delivery delay of controlled replication in heterogeneous networks or networks with correlated mobility patterns, it does so at the expense of *guaranteed* resource usage. This can be a rather undesirable feature of the protocol for some applications. For example, assume an application has stringent energy constraints but some flexibility in the delivery ratio or delay of messages (e.g. a low power sensor network where there is inherent redundancy in the data transmitted). Then, it might be significantly more important to ensure that a bounded amount of energy will be spent per message rather than trying to maximize delivery ratio. As a different example, in a self-organized network where a credit-based system is used to incite collaboration between nodes [12] it is important for the user to be aware exactly how many credits a message will consume.

To this end, we propose the idea of *smart replication* or *utility-based replication*. Similar to the basic, greedy replication algorithm, a fixed number of copies is used; these copies are distributed into a chosen set of relays, and no further forwarding is allowed. This ensures that the total number of transmissions (and thus other resources used) is *controlled and known in advance*. However, instead of naively (greedily) handing copies to the first nodes encountered, relays are chosen according to a utility function. This ensures that *the fixed budget of message copies* is allocated only to the "fittest" possible relays, and no copies are wasted on "non-useful" nodes. In short, *our aim is to have control and predictability of resource usage at the highest priority, while at the same time try to achieve the best performance possible given this fixed set of message replicas.*

In the next section, we will describe the basic utility-based replication algorithm, and present some variations of it. Additionally, we give some insight into how utility-based replication can be tuned to achieve the desirable performance tradeoff. Then, in Section III we'll present simulation results comparing different flavors of utility-based replication against greedy replication. Finally, we'll conclude and present some future work directions in Section IV.

II. Utility-based Replication

A. Greedy Replication in Heterogeneous Environments

Controlled replication or "Spraying" is considered to be an efficient method to reduce the large overhead of epidemic-based schemes, without often incurring significant delay penalties [3], [4], [5]. The basic controlled replication algorithm is the following:

Definition 2.1 (Spray and Wait): When a new message is generated at a source node, this node also creates L "forwarding tokens" *for this message*. A forwarding token implies that the node that owns it, can spawn and forward an additional copy of the given message. During the spraying phase messages get forwarded according to the following rules:

- if a node (either the source or a relay), carrying a message copy and $c > 1$ forwarding tokens for this message, encounters a node with no copy of the message[3], it spawns and forwards a copy of that message to the second node; it also hands over $l(c)$ tokens to that node ($l(c) \in [1, c-1]$) and keeps the rest $c - l(c)$ for itself ("Spray" phase);
- when a node has a message copy and $c = 1$ forwarding tokens for this message, then it can only forward this message to the destination itself ("Wait" phase).

There are two flavors of the basic scheme that have been proposed. In the *2-hop* version of the scheme [13], only the source may forward an extra copy (i.e. $l(c) = 1$ in the above description). In the (binary) *tree-based* version [3], [4], $l(c) = \lfloor \frac{c}{2} \rfloor$ for any node with $c > 1$ tokens. The common characteristic of these two algorithms is that they're both *greedy*. By "greedy" here it is meant that *any opportunity to forward one of the available copies to a new node with no copies will always be seized upon*. In other words, the budget of L message replicas will be distributed to the L first nodes encountered.

The tree-based greedy algorithm is shown to be optimal in a homogeneous environment (I.I.D. node mobility) [3], [4]. The intuition behind this it that, if all nodes are *statistically equivalent* [14], there is no benefit in waiting when a forwarding opportunity arises. Nevertheless, if nodes are heterogeneous (e.g. different mobility characteristics, different resources, etc.) greedy distribution of message copies can be easily shown to be sub-optimal. Consider, for example, an intermittently connected network of mobile nodes, where a percentage p of the nodes are not (that) useful in delivering a message to a remote destination (e.g. move very slowly, or encounter few new nodes over time). We'll call these "snail" nodes. If we assume uniform *mixing rates* [14] for both normal and snail nodes, then with greedy replication pL of the copies, on average, would end up with snail nodes[4]. Using the delay equations of [3] or [4], it is not difficult to show that the performance degradation compared to the case where all nodes are "non-snail" ones is approximately given by

$$\frac{\text{Delay(p snail)}}{\text{Delay(normal)}} \approx c_1 + \frac{c_2}{1-p},$$

where c_1, c_2 are constants, such that $c_1 + c_2 = 1$.

This equation implies that if the number of snail nodes is large, then the delay of greedy Spray and Wait quickly increases. The reason for this is that greedy replication cannot identify nodes that are not useful, and mistakenly hands them over some copies. *The larger the percentage of non-useful relays in the network, the larger the negative impact of greediness*. Of course, a more appropriate comparison would be against the delay of the optimal algorithm given p snail nodes. Nevertheless, the above equation already provides enough insight about the expected performance degradation due to heterogeneity.

B. Utility-based Spraying

Based on the previous exposition we can draw the following conclusion: in a heterogeneous environment, where a limited budget of L message copies needs to be distributed to L relays, a mechanism is necessary that will distinguish the "better" relays, and avoid using the least useful ones. Ideally, we would like to find *the L best relays* in the network (given some optimization criterion). However, this problem is not trivial, even in moderately complex scenarios [15], especially given the fact that candidate relays appear (i.e. are encountered) not all-together, but in an *online* fashion. Therefore, here we will turn our attention to

[3]We assume that a message ID vector exchange similar to that performed in epidemic routing occurs [7].

[4]This is not necessarily true; If snail nodes move less frequently around the network than normal ones, it may be the case that they are also encountered by other nodes less frequently; in that case, pL would just be an upper bound.

heuristic methods to decide on the *fitness* or *utility* of a given node as a relay.

Definition 2.2 (Utility-based Spraying): Similarly to the basic spraying algorithm (see Def. 2.1), Utility-based Spraying uses forwarding tokens to grant a node the right to further forward message copies. Additionally,

- each node i maintains a utility function $U_i(j)$ for every other node in the network j. $U_i(j)$ reflects the probability that node i will deliver a message to node j, and it may be based on a number of different parameters (e.g. encounter history, mobility, friendship index with j, etc.)
- if a node i (either the source or a relay) carrying a message copy for a destination d and $c > 1$ forwarding tokens for this message encounters a node j with no copy of the message, it spawns and forwards a copy of that message to the second node according to one of the following rules:
  - **rule 1:** if $U_j(d) > U_{th}$ for some U_{th} threshold value (*absolute utility criterion*);
  - **rule 2:** if $U_j(d) > U_i(d)$ (*relative utility criterion*);

  It also hands $l(c) = \lfloor \frac{c}{2} \rfloor$ forwarding tokens to that node and keeps the rest $\lceil \frac{c}{2} \rceil$ for itself (i.e. tree-based)[5];

We are going to describe different variations of this basic algorithm in terms of the utility function they use. Each algorithm could use either of the forwarding rules above ("absolute utility" or "relative utility") or a combination of them (e.g. "use rule 1 to ensure a minimum utility and then rule 2 among the nodes that qualify"). Rule 1 requires a good threshold parameter U_{th} to be found in every case. Rule 2 is easier to implement, yet it does not guarantee that all "bad nodes" will be avoided (e.g. "very low utility" nodes could still give copies to just "low utility" nodes).

Candidate utility functions could be broadly categorized into *destination-dependent* ("DD") and *destination-independent* ("DI") functions:

Destination-dependent (DD) Utility: One node may be the best relay for one destination (d_1), and another node the best relay for a different destination (d_2). In other words, for DD functions:

$$U_i(d_1) > U_j(d_1) \text{ but } U_i(d_2) < U_j(d_2), d_1 \neq d_2. \quad (1)$$

Examples of DD utility functions could be those based on age-of-last-encounter for a given destination [11], [2], social relation with a given destination [16], etc. Destination-dependent utility functions impose a larger overhead on nodes, as in essence they need to maintain an entry for every other node in the network. To reduce this overhead caching techniques could be used (e.g. storing only the m highest utility values).

Destination-independent (DI) Utility: The "utility" of a given node is independent of any destination; instead, it depends on some special characteristic(s) this node has. This implies that one node may be the best relay for most or all destinations. In other words, for DI functions it holds in general that:

$$U_i(d_1) \geq U_j(d_1) \Rightarrow U_i(d) \geq U_j(d), \text{for most or all } j, d. \quad (2)$$

Examples of nodes which are highly preferable as relays for any destination would be nodes with high and frequent mobility (e.g. vehicles), nodes with many "friends" (e.g. *hubs* in scale-free networks), or nodes with higher resources (e.g. a "message ferry" [1]). DI utility functions have a smaller overhead than DD functions as they require each node to maintain only a single utility value each. Yet, DI functions also imply that the "better" nodes might have to bear a higher forwarding overhead than others. This might result in poorer load-balancing and utilization of the total network capacity, and/or faster battery drainage of a few nodes.

We now turn our attention to specific replication algorithms, some using DD and others DI utility functions. Note that it is possible to define hybrid algorithms also that take into account both the general *fitness* of a node as well as destination-specific information when making a forwarding decision. Due to space limitations, we defer an investigation of such hybrid functions for future work. All algorithms hereafter are variations of the basic utility-based replication scheme described in Def. 2.2.

C. Last-Seen-First (LSF) Spraying

The basic idea is to choose as relays nodes that have seen the destination most recently. Thus, LSF is an example of a destination-dependent utility function.

Definition 2.3 (LSF Spraying): Each node i maintains a timer $\tau_i(j)$ for every other node j in the network, which records the time elapsed since the two nodes last encountered each other as follows: initially set $\tau_i(i) = 0$ and $\tau_i(j) = \infty, \forall i, j$; if i encounters j, set $\tau_i(j) = \tau_j(i) = 0$; otherwise increase each $\tau_i(j)$ at every time unit. Finally $U_i(j) = \frac{1}{1+\tau_i(j)}$.

The basic idea of using age-of-last-encounter timers has been introduced to advance a single message copy towards its destination in a connected mobile network, using gradient-based routing [11]. Rather than starting with a random set of relays and use age-of-last-encounter to find better and better ones as in [11], [3], LSF tries to choose a good set of relay nodes from the beginning, using age of last encounter to identify nodes that often see or recently saw the destination.

D. Most-Mobile-First (MMF) Spraying

This algorithm uses a destination-independent (DI) utility criterion. It assumes that some nodes are more "mobile" than others. This scheme would work well, for example, when the majority of nodes participating in the ad hoc network are pedestrians moving within a local area, while a few nodes are vehicles (e.g. buses, taxis, etc.) that tend to visit larger areas and with higher speeds (and thus have a higher chance of meeting new nodes). For now, we'll assume that each of these nodes carries a *label* that states the type of the node, e.g. "BUS","TAXI", "PEDESTRIAN", "BASE STATION". A similar label assignment is performed in [17] to assign labels based on affiliation. Although, in some scenarios, it wouldn't be too burdensome to manually configure a label (e.g. by setting some software parameter when installing a radio, say, on the top of a bus), in the next section we discuss one way of assigning labels automatically.

Definition 2.4 (MMF Spraying): Assume there are m total node labels, $LABEL_1, LABEL_2, \ldots, LABEL_m$. Each node i is assigned one of the labels, let $LABEL(i)$, and we define the utility of node i as $U_i(j) = U_i = LABEL(i), \forall j$. Finally, labels are put in a preference order ($\succ$):

$$LABEL_1 \succ LABEL_2 \succ \cdots \succ LABEL_m$$

[5]A more generic approach would be to make $l(c)$, the number of tokens forwarded, a function of the relay's utility also: $l_c = f(c, U_j(d))$. This would increase the flexibility but also the complexity of the distribution algorithm. Due to space limitations, we choose not to address this issue here.

This assignment of preference order to labels can be made offline, based on the general mobility statistics and perceived usefulness of different types of nodes. Furthermore, one could also have different preference orders depending one the type (label) of the destination. For example, nodes of $LABEL_j$ may not be good relays in general, but excellent candidates if the destination is also of $LABEL_j$. This would actually bring the scheme somewhere in between DI and DD.

E. Most-Social-First (MSF) Spraying

In the previous scheme we assume that, somehow, information about the mobility characteristics of a node might be readily available. However, in many scenarios the same wireless device might be carried by a pedestrian, left at the office desk, or lie inside a vehicle at times. Hence, we need a mechanism that could estimate the "degree of mobility" *online*. What is more, some nodes might encounter more nodes than average not due to mobility, but just because they visit some "hub" locations (e.g. cafeteria) more often, or have more *social links* than the average node. This implies that a more appropriate metric of the utility of a node might be its *sociability* rather than its *mobility*.

Definition 2.5 (Sociability): Let us look at a node i during a time interval $t_n = [(n-1)T, nT]$, where T is the duration of the interval. Let us further define the indicator function $I_{ij}(t)$, which shows whether nodes i and j are neighbors at time t (e.g. have performed an "association" with Bluetooth). Finally, let $N_i(n)$ denote the set of nodes j such that

$$N_i(n) = \{j \neq i : \exists t \in [(n-1)T, nT] \text{ for which } I_{ij}(t) = 1\}.$$

Then, we define the sociability $S_i(n)$ of node i during the interval t_n as $S_i(n) = \frac{\|N_i(n)\|}{T}$.

In general, the sociability value of a node will be a function of the time interval during which it is measured. If a node's statistical behavior varies over time, then its sociability index might also change between time intervals. This implies that it might be more appropriate for a node to maintain a *running average* of its *perceived* sociability index, rather than just looking at the previous interval.

Definition 2.6 (MSF Spraying): Each node i maintains a running average for the "sociability index" $\hat{S}_i$ as follows: for a given time interval $t_n = [(n-1)T, nT]$ (T = sliding window duration) it counts the number of unique node IDs encountered, $N_i(n)$[6]. Then, at the end of window n it updates $\hat{S}_i$ as

$$\hat{S}_i = (1-\alpha)\hat{S}_i + \alpha\frac{N_i(n)}{T},$$

and proceeds to the next interval t_{n+1}. α is the weight given to the current measurement in the sliding window mechanism. Finally, the utility of node i is given by $U_i(j) = U_i = \hat{S}_i, \forall j$. This is another example of a DI utility function.

F. Choosing the Protocol Parameters

Sliding Window for Sociability Estimation: Setting the sliding window parameters in MSF correctly (window duration T, and α) is not trivial in the general case. Whether or not the running sociability estimate $\hat{S}_i$ will be a useful predictor of future interactions depends on these parameters, the amount of "structure" in the interaction and mobility patterns of the nodes involved, and the desired "horizon" for the prediction. Although it is beyond the scope of this paper to deal with the general case, there are some situations that are easier to handle. For example, if a node's behavior is homogenous over time

$$E[S_i(t_n)] = E[S_i(T)], \forall n \text{ (time-homogeneous)}.$$

In the time-homogeneous case, it is not important during which time window ones looks at a node's behavior. A node with a highly social past behavior, will also be a node with a very social future behavior. An example of this could be if all nodes move according to the Random Waypoint model, but different nodes move with different average speeds and/or pause times. A node i with higher speed and shorter pauses than another node j, will meet *on average* more new nodes in the same amount of time than j, *at any time scale*[7].

Real-life mobility, unlike the above example, usually varies over time (e.g. people tend to have different mobility/interaction patterns during different times of the day). Yet, even in these cases, time homogeneity may also emerge given a large enough time horizon. It has been frequently observed that real-life mobility exhibits time-periodicity [19]. Thus, if the duration of such a period is not prohibitive compared to the desired delivery delay, setting the sliding window to this value could produce a quite reliable predictor for different nodes' sociability. On the other hand, if finer granularity is needed (i.e. targeted delays are much smaller than this period), a shorter sliding window should be used, along with a smaller α to filter out situations where, for example, a node might occasionally meet a large number of nodes, but then few nodes for long stretches of time. Finally, maintaining higher moments of a node's past sociability might also help in identifying such pathological situations, and allowing a node to make more informed decisions.

Number of Copies: Another important protocol parameter is the number of copies L to be used. This number has been calculated for Greedy Spraying in a homogeneous network, in order to achieve a desired *transmission-vs-delay* tradeoff [3]. The same procedure could be modified for the case of MMF spraying. The main idea is that, if there are M total nodes and MMF uses only p percent of them based on their labels, then this is equivalent to a network of pM nodes (some care is needed though, regarding the effect of the other $(1-p)M$ nodes in the minimum/optimal delay [3]).

In the case of MSF Spraying, a different approach could be taken, as described by Lemma 2.1.

Lemma 2.1: Assume there are M total nodes in a network moving independently of each other. Let further each node use the MSF algorithm with a threshold sociability value $S_{th} = \frac{N_{th}}{T}$ (rule 1), according to Def. 2.5 and 2.6. If each message has a TTL value of T_{max}, and the desired delivery probability for each message is R_{min}, then the number of copies per message L_{min} and sociability threshold must satisfy

$$1 - \left(1 - \frac{N_{th}}{T_{max}}\right)^{L_{min}} \geq R_{min}. \quad (3)$$

The above lemma is easy to derive by setting the sliding window duration T equal to the TTL value (T_{max}). Then, the probability that a chosen relay will not find the destination before the TTL

[6] If T is not too large, the overhead of maintaining a list of unique node IDs and look-up time when a new node is found is not significant.

[7] We assume here that no encounters are missed. In reality, if a node moves very fast, it might not discover some contact opportunities which are short-lived, due to the workings of the neighbor discovery mechanism [18].

expires is $1 - S_{th}$, and $(1 - S_{th})^{L_{min}}$ assuming the L_{min} relays encounter other nodes in the network independently. Although this might not be realistic in practice, it could serve as a useful rule of thumb to either find a useful threshold value given L or the number of copies L_{min} given S_{th}.

In general, the problem of choosing the number of copies *and* the copy carriers optimally in a heterogeneous network is difficult. In [15] the authors formulate one version of the problem using dynamic programming and find the optimal (but centralized) solution only for some special cases. Another interesting effort to address such issues from the point of view of reliability borrows some ideas from modern portfolio theory [20]. Due to space limitations, we intend to address this issue further in future work.

III. Simulation Results

We have used a custom discrete event-driven simulator to evaluate and compare the performance of the different Utility-based Spraying algorithms against the basic (greedy) replication scheme (i.e. tree-based Spray and Wait [3], [4]). A simplified version of the slotted CSMA (Carrier-Sense Multiple Access) MAC protocol has been implemented (further details about the simulator can be found in [3]).

We assume that 100 nodes move according to the "Community-based Mobility Model" [21], which is motivated by real mobility traces like [18]. In the Community-based model, each node has its own small community (network size $= 500 \times 500$, community size $= 50 \times 50$) inside which it moves preferentially for the majority of time (e.g. the user's department building on a campus). Every now and then it leaves its community (with probability $1 - p_l$), roams around the network for sometime (e.g. going to the cafeteria, library), and then decides to return to its community (with probability $1 - p_r$). Finally, to capture node heterogeneity, each node may have different mobility characteristics (p_l, p_r) in addition to different communities (details about the model in [21]).

A. LSF Spraying

The first scenario considered (Scenario 1) consists of 4 *different types of nodes* (e.g. to capture a hybrid metropolitan network): (*local nodes*) 40% of the nodes move locally most of the time ($p_l \in [0.85, 0.95]$) but may occasionally roam in the whole network ($p_r \in [0.1, 0.2]$); (*community nodes*) 40% of the nodes move only inside their own community ($p_l = 1, p_r = 0$); (*roaming nodes*) 10% of the nodes roam quite often outside their community ($p_l \in [0.7, 0.8], p_r \in [0.3, 0.5]$); (*base stations*) 10% of the nodes are static and uniformly distributed in the network, corresponding for example to base stations or static repeaters.

In Fig. 1 we depict the improvement in the delivery delay of greedy Spray and Wait ("SW"), when message replicas are handed over using LSF Spraying only to nodes that have a last-encounter timer value for the destination that is higher than some threshold value U_{th} (i.e. rule 1 is used – threshold values for this scenario were $500 - 1000$ time units). As can be seen there, taking into account the age of last encounter when spraying the L copies can indeed improve performance. What is also interesting to note here is that, unlike the case where multi-hop forwarding is allowed (e.g. [2], [3]), not having transitivity of utility [2] is preferable in this case. This is reasonable. If no further forwarding is allowed (except during the initial tree-based replication phase [4]), we are only interested in relays that have a good chance of delivering a message *themselves*.

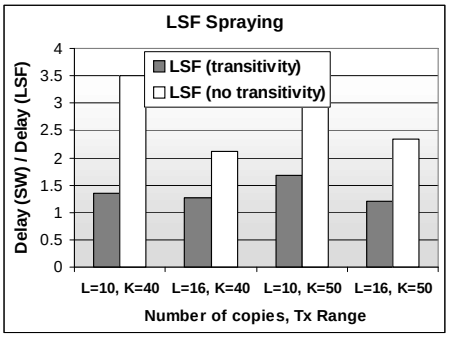

Fig. 1. Scenario 1: performance improvement of LSF Spraying over greedy Spray and Wait (SW); K =node's transmission range, L = number of copies.

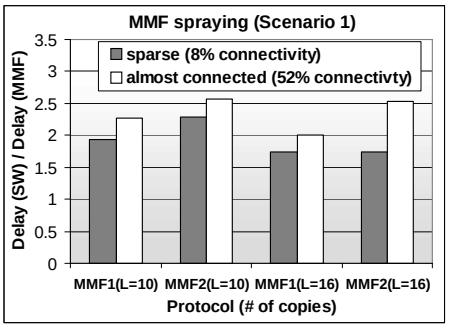

Fig. 2. Scenario 1: performance improvement of MMF spraying over greedy Spray and Wait (SW), for two different connectivity scenarios (sparse and almost connected); L is the number of copies.

B. MMF Spraying

Here, we turn our attention to the MMF Spraying scheme. In Fig. 2 we compare the delivery delay of MMF Spraying and Greedy Spraying, in the same scenario as Sec. III-A (Scenario 1). We compare two levels of knowledge for the MMF scheme: (i) messages are given only to nodes of type "local" or type "roaming" but never to "static" or "community" (denoted as MMF1), and (ii) now a relay must also have $\{p_l < 0.9$ AND $p_r > 0.15\}$ in addition to label "local" or "roaming" (MMF2). We show plots for both a very sparse network ($< 8\%$ of nodes are connected) and a denser one ("flakynet" – around 50% of nodes are connected). As can be seen by these plots, by choosing only nodes that actually have a chance to deliver a message we can get up to $2.5-3\times$ reduction in delay in this scenario. Greedy spraying wastes many copies by handing them over to community and static nodes.

Intuitively, the larger the amount of node heterogeneity the higher the improvement is expected to be by using "smarter" copy distribution algorithms. To confirm this, in this second scenario we assume only two types of nodes: (i) $p\%$ are "roaming" nodes that are useful as message relays and (ii) $(1-p)\%$ are "community" nodes that are useless as message carriers, unless the destination lies inside their community (small probability). In Fig. 3 we compare the performance of greedy Spray and Wait and MMF Spraying, which gives copies only to roaming nodes, as the percentage p of roaming nodes decreases.

From this figure, it is evident that the smaller the percentage of useful ("roaming") nodes the higher the delay improvement by using smarter spraying. This is because greedy forwarding makes an increasing number of errors, wasting more and more

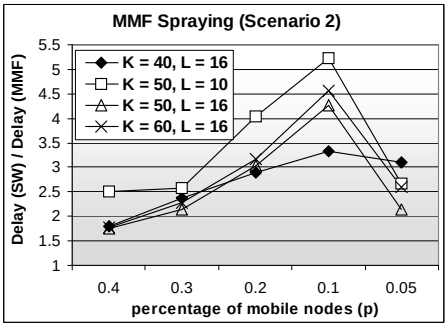

Fig. 3. Scenario 2: performance improvement over greedy Spray and Wait by MMF spraying, as a function of the percentage p of "useful" relay nodes; $K =$ nodes' transmission range.

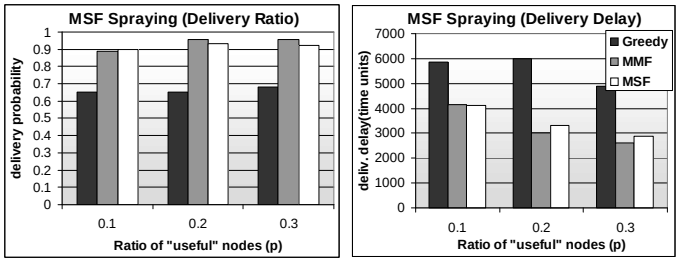

Fig. 4. Performance comparison of MSF, MMF, and greedy Spray and Wait, as a function of the percentage p of "useful" relay nodes.

copies ($(1-p)L$). On the other hand, we see that the achievable improvement peaks at a given value of p and starts becoming smaller again. At this point the number of useful relays is so small (smaller than the number of copies) that the smarter Spraying scheme takes a very long time to find them.

C. MSF Spraying

In this last scenario we'll examine the more practical scheme, MSF Spraying. Unlike the MMF scheme which uses preprocessed information (LABELs) to identify globaly useful relays, MSF tries to estimate this utility online by observing the type and number of encounters the node gets involved into. Fig. 4 compares the delivery delay for a scenario similar to that of Fig. 3, where only a percentage p of nodes are useful as message relays. It compares the performance of MSF to Greedy Spraying, as well as MMF spraying (which knows beforehand which are the useful nodes)[8]. As can be seen there, both MSF and MMF improve the performance of Greedy spraying. What is more, MSF can approximate the performance of MMF without the fore-knowledge that the latter requires. (As a final note, although we do not have space to depict plots for the behavior of the running sociability estimates, we have observed that it manages to identify the "good" nodes after only a few windows.)

IV. Conclusion

In this work, we have looked into the issue of efficient mobility-assisted or store-carry-and-forward routing. Specifically, we have explored the issue of allocating a fixed budget of message replicas to a number of candidate relays, according to a utility function that captures how probable it is for that relay to encounter the destination in the near future. We have presented a number of different heuristics for the distribution of message copies, and have shown that *choosing* the relays carefully rather than picking the first few relays encountered [3], [4], [5], can significantly improve performance in heterogeneous scenarios.

More sophisticated or hybrid heuristics could be envisioned to improve performance further, and the issue of *optimal* allocation of message replicas to relays in heterogeneous environments is open. Yet, we believe that this work clearly shows that utility-based replication is a desirable and often necessary routing primitive, when the number of total transmissions needs to be controlled (e.g. energy-constrained environments, credit/price-based systems for self-organized networks, etc.). Alternatively, using utility-based replication rather than greedy one during the initial distribution phase of message replicas could have an improvement on the performance of schemes that allow multi-hop forwarding also [3]. We intend to look further into this last issue in future work.

[8] We have used $L = 10$ copies per message for all schemes, Tx range $K = 30$, the sliding window for MSF is 1000 time units, and $\alpha = 0.8$; the sociability threshold was chosen manually trying different values.

References

[1] E. P. Jones and P. A. Ward, "Routing strategies for delay-tolerant networks," *Submitted to ACM Computer Communication Review (CCR)*.

[2] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *to appear in Transactions on Networking*, 2007.

[3] ——, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *to appear in Transactions on Networking*, 2007.

[4] T. Small and Z. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN)*, 2005.

[5] G. Neglia and X. Zhang, "Optimal delay-power tradeoff in sparse delay tolerant networks: A preliminary study," in *Proceedings of ACM SIGCOMM workshop on Challenged Networks (CHANTS'06)*, 2006.

[6] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, 2002.

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

[8] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, no. 3, 2003.

[9] 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 *Proceedings of ACM ASPLOS*, 2002.

[10] J. Blum, A. Eskandarian, and L. Hoffman, "Challenges of intervehicle ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*.

[11] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *Proceedings of ACM MobiHoc*, 2003.

[12] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *IEEE INFOCOM*, 2003.

[13] R.Groenevelt, G. Koole, and P. Nain, "Message delay in manet (extended abstract)," in *Proc. ACM Sigmetrics*, 2005.

[14] D. Aldous and J. Fill, "Reversible markov chains and random walks on graphs. (monograph in preparation.)," *http://stat-www.berkeley.edu/users/aldous/RWG/book.html*.

[15] A. Jindal and K. Psounis, "Optimizing multi-copy routing schemes for resource constrained intermittently connected mobile networks," in *to appear in IEEE Asilomar Conference on Signals, Systems and Computers*, 2006.

[16] M. Musolesi and C. Mascolo, "A community based mobility model for ad hoc network research," in *Proceedings of ACM REALMAN*, 2006.

[17] P. Hui and J. Crowcroft, "How small labels create big improvements," in *IEEE ICMAN, in conjunction with PerCom*, 2007.

[18] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proceedings of IEEE INFOCOM*, 2006.

[19] W. Hsu and A. Helmy, "Impact: Investigation of mobile-user patterns across university campuses using wlan trace analysis," University of Southern California, Tech. Rep. 2005.

[20] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proceedings of ACM SIGCOMM*, 2005.

[21] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of ACM/IEEE MOBIHOC*, 2006.