On Multicopy Opportunistic Forwarding Protocols in Nondeterministic Delay Tolerant Networks (Supplementary File)

Cong Liu and Jie Wu, IEEE Fellow

October 26, 2011

1 Additional Background

This section reviews the optimal stopping rule problem [1], discusses the protocols, which we will implement and compare with the proposed ones in our evaluation, in greater detail, and presents additional related work on routing protocols in *delay tolerant networks* (DTNs).

1.1 The optimal stopping rule problem (with an example)

Let us briefly review the optimal stopping rule problem [1] with an example. In the stopping rule problem, we may observe a sequence $X_1, X_2, \ldots$ for as long as we wish, where $X_1, X_2, \ldots$ are random variables whose joint distribution is assumed to be known. For each stage $t = 1, 2, \ldots$ after observing $X_1, X_2, \ldots, X_t$, we may stop and receive the known reward y_t, or we may continue and observe X_{t+1}. In the latter case, the bit X_t on day t will not be valid anymore on day $t + 1$. The optimal stopping rule is to stop at some stage t to maximize the expected reward.

A stopping rule problem has a finite horizon if there is a known upper bound T on the number of stages at which one may stop. If stopping is required after observing $X_1, \ldots, X_T$, we say the problem has a horizon of T. In principle, such problems may be solved by the method of *backward induction*. Since we must stop at stage T, we first find the optimal rule at stage $T-1$. Then, knowing the optimal reward at stage $T-1$, we find the optimal rule at stage $T-2$, and so on, back to the initial stage (stage 0). Let $V_t^{(T)}$ $(1 \leq t \leq T)$ represent the maximum expected reward one can obtain, starting from stage t. We define $V_T^{(T)} = y_T$, and then inductively for $t = T-1$, go backwards to $t = 0$,

$$V_t^{(T)} = E(max\left\{y_t, V_{t+1}^{(T)}\right\}).$$

The meaning of the above equation is that, at stage t, we compare the reward for stopping, namely y_t, with the best reward $V_{t+1}^{(T)}$ that we expect to be able to get by continuing and using the optimal rule for stages $t+1$ through T. The optimal reward is therefore the maximum of these two quantities, and it is optimal to stop at the earliest t when $y_t \geq V_{t+1}^{(T)}$.

Here, we use a "house-selling" scenario as a simple example for the finite horizontal optimal stopping rule problem. Suppose we have a house to sell within T days. An offer comes in each day and X_t denotes the monetary amount of the offer received on day t. $X_1, \ldots, X_T$ are independent and identically-distributed (i.i.d.), and are uniform over 0 to M. We may stop at any day t and receive $y_t = X_t$. We don't know the offers before they come in, and we cannot recall a past offer. We need to find a stopping rule that maximizes the expected sales value.

Let us derive the optimal stopping rule using the backward induction method. Since we must sell the house by day T, the expected offering in the last day is $V_T^{(T)} = E(y_T) = E(X_T) = \frac{M}{2}$. Inductively, at day t,

$$V_t^{(T)} = E(max\left\{y_t, V_{t+1}^{(T)}\right\}) = \int_0^M max\left\{x, V_{t+1}^{(T)}\right\} dF(x)$$

$$= \int_{V_{t+1}^{(T)}}^M x d\frac{x}{M} + \int_0^{V_{t+1}^{(T)}} V_{t+1}^{(T)} d\frac{x}{M} = \frac{M^2 + (V_{t+1}^{(T)})^2}{2M},$$

where $F(x) = \frac{x}{M}$ is the cumulative distribution function of y_t, a uniform distribution over 0 to M. We can calculate $V_t^{(T)}$ inductively for $t = T-1$ down to 1. The optimal stopping rule is to sell the house on the first day t when $X_t \geq V_{t+1}^{(T)}$. In other words, the optimal stopping rule uses the expected reward in stage $t+1$ as the decision threshold for stage t.

1.2 Protocols in comparison (in greater detail)

We compare OOF and OOF- against several opportunistic forwarding protocols. While OOF and OOF- have well-defined utilities to maximize in each forwarding (the joint expected delivery probability or the joint expected delay of all copies of each message), other algorithms use either heuristic forwarding rules or blind forwarding.

Epidemic [2]. A node sends a copy of the message to every node it encounters that does not have a copy already, until its copy of the message times out. Given a long enough time-out, the destination, as well as every other node in the network, will eventually have a copy of the message.

Spray-and-wait [3]. This protocol differs from epidemic in that it controls the number of copies of each message in the network. A number L of logical *tickets* are associated with each message. A node i can only copy a message to another node j it encounters if (1) j is the destination ($j = d$) or (2) $j \neq d$ and the message in i owns $L > 1$ tickets. If forwarded, the new copy in j will have $L_j = \lfloor L/2 \rfloor$ tickets, and $L_i = L - L_j$ tickets will remain with the copy in i.

MaxProp*. A cost is assigned to each node regarding each destination. Each node i keeps track of a probability f_j^i of the next meeting node being j, and disseminates it to every node in the network. The delivery probability from a source to a destination is the total cost $\sum_{(i,j)\in P}(1-f_j^i)$ on their shortest (in terms of total cost) path P calculated by the Bellman-Ford algorithm, where the cost of each forwarding (i,j) is $1 - f_j^i$. We use a variation MaxProp*, which differs in that (1) it incorporates the hop-count-limited forwarding protocol to control forwarding overhead in order to make a fair comparison, and (2) it assumes that each node can carry an infinite number of messages. Note that the first modification will effect the performance negatively.

Delegation [4]. Delegation forwarding may use a wide range of forwarding metrics (qualities). We use the mean inter-meeting time $I_{k,d}$ of node k, with destination d, as the forwarding quality of a node k, and node j has a higher forwarding quality than node i if $I_{j,d} < I_{i,d}$. In delegation, each message copy maintains a forwarding threshold τ, initialized to be the forwarding quality of its source node ($I_{s,d}$). When node i meets node j, i forwards a message to j if the forwarding quality of node j exceeds the message's threshold τ ($I_{j,d} < \tau$). Then, the τs of both copies in i and j are set to $I_{j,d}$. In the case that $I_{j,d} < \tau$, but j already has the message copy, the copy is not forwarded, but the τ of the copy in i will still be set to $I_{j,d}$.

Epidemic and Spray-and-wait do not use any forwarding metric. The performance of Spray-and-wait degrades the fastest as the network size increases. This is because, when forwarded randomly, the chance that some of the L copies of a message end up in the hands of quality forwarders decreases as the network size increases. In terms of cost, Spray-and-wait, OOF, and OOF- maintain a constant cost in terms of the number of forwardings per message, which achieves ultimate scalability. Delegation has an $O(N)$ worst-case cost and an $O(\sqrt{N})$ average-case cost. The cost increase proportional to the network size N may result in a degraded performance in a small network and an excessive cost in a large network.

1.3 Additional related works

The Delay Tolerant Network Research Group (DTNRG) [5] has designed a complete architecture to support various protocols in DTNs. In [6], Jain, Fall, and Patra presented a comprehensive investigation on the DTN routing problem with different levels of prior knowledge about the network. Specifically, Bellman-Ford algorithm (with future connectivity information) or the linear programming approach (with information of future connectivity and traffic demands) is used to obtain an optimal path between a source and a destination. In [7], Merugu, Ammar, and Zegura proposed a DTN routing algorithm that is similar in spirit to Bellman-Ford algorithm in [6]. In [8], Liu and Wu presented hierarchical routing in DTNs with deterministic repetitive mobility to improve scalability.

Epidemic routing [2] is the first flooding-based routing algorithm. Gossip [9] forwards to each encountering node with probability p. Opportunistic routing protocols, such as [10], make forwarding decisions based on the comparison of the nodes in terms of some delivery probability metric. Different delivery probability metrics are defined including encounter frequency [10], time elapsed since last encounter [11, 12, 13, 14, 15], social similarity [16, 17], location similarity [18], time-varying expected delay [19], timely-contact probability [20], and geometric distance [21].

Trace data available for the research community [22] include the UMassDieselNet trace, the NUS student contact trace, and the MIT Reality Mining [20]. In [20], several opportunistic routing algorithms are simulated in large realistic contact traces. A timely-contact probability metric is proposed in this paper, which captures the contact frequency of mobile nodes and is similar to [10] and [12] in spirit.

The optimal stopping rule is also applied in [23] to design the joint optimal opportunistic scheduling and channel state information acquisition strategies to increase the transmission rate.

2 Discussion and Analysis

This section will discuss how OOF and OOF- can work with incomplete routing information and will analyze the complexity and limitations of both.

2.1 Routing with incomplete information

OOF and OOF- can work with incomplete routing information. To calculate P_{i,d,K,T_r} for any K and T_r, each node collects the mean inter-meeting times for every pair of nodes. When the nodal mobility in the network exhibits long-term regularities, the mean inter-meeting times are long-term stable, which can be infrequently updated, and therefore, the amortized overhead of disseminating this routing information is low in the long run. In practice, they can be generated from historical connectivity information (as in the UMassDieselNet trace [12, 24]), or in some situations, from prior knowledge on the contact pattern of the nodes (as in the NUS student contact trace [25]).

The mean inter-meeting times can also be incrementally exchanged among the nodes. When node i meets node j, node i sends to node j its mean inter-meeting times with the other nodes (1-hop routing information), or it can also send the mean inter-meeting times received from the other nodes (k-hop routing information). Alternatively, node i can send only some preferred information to j, e.g., the mean inter-meeting times between frequently meeting nodes.

OOF and OOF- can work with incomplete information, i.e., when the mean inter-meeting times between all of the pairs of nodes are not available to every node. To allow OOF and OOF- to work with incomplete routing information, for each pair of nodes i and j whose mean inter-meeting time is unknown, we simply set their (1) mean inter-meeting time $I_{i,j} = \infty$, (2) time-slot based meeting probability $M_{i,j} = 0$, (3) 1-hop delivery probability $P_{i,j,0,T_r} = 0$ for all time-slots T_r, and (4) the expected delay $D_{i,j,0} = \infty$. When no routing information is available at all, it is easy to see that OOF behaves as spray-and-wait, which spawns copies to the first node seen. To allow OOF (OOF-) to behave as Spray-and-wait, node i will forward a message to node j if both $P_{i,d,K,T_r} = 0$ and $P_{j,d,K-1,T_r-1} = 0$ (if both $D_{i,d,K} = \infty$ and $D_{j,d,K-1} = \infty$).

2.2 Complexity

In this subsection, we will discuss the complexity of our algorithms. The online portion of our algorithms involves a simple table lookup and several comparing operations: its computation complexity is $O(1)$. The offline portion is not as simple as the former, but it only needs to be computed when there is a substantial update in the inter-meeting times between the nodes. To save energy, the offline computation can be performed when the mobile device is charging. It can also be sent to a computer if the mobile device is connected to one.

The offline portion may require a large amount of computation. For OOF, we need to compute P_{i,d,K,T_r} for each routing node i, each possible destination d, each remaining hop-count K, and each residual lifetime T_r, using Algorithm 1. Therefore, if the number of routing nodes is N_1, the number of destinations is N_2, the maximum hop-count is H, and the maximum TTL is T_{max}, the computation complexity is $N_1 \times N_2 \times H \times T_{max} \times O(N_2 log N_2) = O(N_1 N_2^2 log N_2 H T_{max}$, where $O(N_2 log N_2)$ is the computation complexity of Algorithm 1. The storage complexity of OOF is the size of the table needed to store P_{i,d,K,T_r}, which is $O(N_1 N_2 H T_{max})$.

For OOF-, we need to compute $D_{i,d,K}$, for each routing node i, each possible destination d, and each remaining hop-count K, using Algorithm 2. Therefore, the computation complexity is $N_1 \times N_2 \times H \times O(N_2 log N_2) = O(N_1 N_2^2 log N_2 H$, where $O(N_2 log N_2)$ is the computation complexity of Algorithm 2. The storage complexity of OOF is the size of the table needed to store $D_{i,d,K}$, which is $O(N_1 N_2 H)$.

2.3 Limitations

Firstly, our algorithms suffer from the limitations of all routing algorithms in delay tolerant networks that do not rely on infrastructure, which means the amount of delay can be large due to the limited connectivity in the networks. This make it unsuitable for a large amount of time-critical applications.

Secondly, our algorithms assume regularities in the inter-meeting times to accurately predict the probability of delivery and the estimated delay. In networks where there is not much regularity in node mobility, improvement in routing performance becomes less significant. However, as can be found in our simulation results, our algorithms show the best performance improvement as the performances of all of the routing protocols degrade, due to the lack of regularity in mobility.

Lastly, our algorithms, in their current forms, demand relatively large amounts of computation and storage, which limits their application in larger networks. In practical use, different methods might need to attack the computation and storage problem, by (1) using only the most socially active nodes as routing nodes, (2) using a compact table for delivery probabilities, which only stores the largest entries for each destination, or (3) using dynamic time-slots, e.g., we can set some several hours during the night as a single time-slot, when connectivity among the network hardly changes.

3 Additional Evaluations

Additional simulation evaluations are performed to compare the proposed protocols, OOF and OOF-, against other forwarding protocols, using the NUS student contact trace and the UMass-DieselNet trace. In the NUS student contact trace, which is a synthetic trace, we measure the performance of the compared protocols with different variable parameters. In the UMass-DieselNet trace, we evaluate their performance with partial routing information, false routing information, and stale routing information.

3.1 NUS student contact trace

Accurate information about human contact patterns is available in scenarios, such as university campuses. As shown by the National University of Singapore (NUS) student contact trace model [25], when the class schedules and student enrollment for each class on a campus are known, accurate information about contact patterns between students, over large time scales, can be obtained without long-term contact data collection. The schedules of the 4,885 classes and the enrollment of 22,341 students for each of these classes for each week of 77 class hours, are publicly available on [22]. Their contact model is simplified in several ways: (1) two students are in contact with each other only if they are in the same classroom at the same time; (2) sessions start on the hour and end on the hour, which means that hours are the unit of time for the contact duration; (3) only the contacts that take place during the 11 class hours per day are used. Non-class hours are removed. The trace synthesized in this model exhibits characteristics observed in the real world.

We generate networks for our experiments by selecting some of the students instead of using all of the students in the network for the following two reasons: (1) the generated networks

Table 1: Settings for NUS student trace.

parameter name	default	range
number of students (N)	400	100∼500
number of messages	$3N^2$	
attendance rate (P_{attend})	0.8	0.1∼0.9
connectivity factor (C)	0.4	0.1∼0.9
message time-to-live (TTL)	7 days	1∼7 days
tickets in spray-and-wait (L)	10	
initial hop-count (K)	3	
length of time-slot (U)	1 hour	
simulation time	7 days	

allow us to perform experiments with networks that have different characteristics, including network size and degree of connectivities; and (2) the storage requirement, which is $O(N^2)$ in a network of N students, and the corresponding computational overhead in backward induction are overwhelming when $N = 22,341$. We select a number of N ($100 \leq N \leq 500$) students in different simulations. We define a *connectivity factor* C to determine the degree of connectivity of the nodes in the network. Specifically, $C = |S_1|/(|S_1| + |S_2|)$, where for each student s, two sets of students, S_1 and S_2, are defined such that s is similar to the students in S_1 and is dissimilar to those in S_2. Here, the similarity between two students is defined in terms of the number of common class sections they enrolled in. Please refer to [19] for details on student selection.

We generate non-deterministic traces by taking absentees into consideration. Each student that attends a class has an attendance probability, P_{attend}. The settings in our simulation are shown in Table 1. In the beginning of each simulation, every node sends three messages to every other node, and the total number of messages is $3N^2$, where N is the number of students. In these traces, we assume unlimited messages can be forwarded in each contact whose duration is one hour.

3.1.1 Simulation results

The delivery rates of the protocols are compared in Figures 1(a), 1(b), 2(a), and 2(b) with different attendance rates, connectivity factors, message time-to-lives, and numbers of nodes (students). We do not include Epidemic in the NUS traces because there are too many nodes in this trace, which results in too many copies with Epidemic. If the result of Epidemic is shown, the number of forwardings of other routing protocols all look close to the x-axis and are hard to compare. The results show that, in default settings, OOF and OOF- deliver around 40% more than MaxProp* and Delegation, and 150% more than Spray-and-wait. The delivery rates of OOF and OOF- are very close.

While OOF and OOF- maintain high delivery rates, they also have a low cost in terms of number of forwardings, which is shown in Figures 1(c), 1(d), 2(c), and 2(d). The forwarding cost of OOF is on average 15% less than Spray-and-wait, 8% more than MaxProp*, and 45% more than Delegation. As shown in Figure 1(d), the difference between OOF and OOF- shows that OOF is able to refrain from forwarding when forwarding opportunities are in abundance. It shows that OOF can have a smaller cost than OOF- under the same delivery rate. On the other hand, OOF- does not have a sense of message residual lifetime. Note that, although Delegation has a smaller cost than OOF in these simulation results, the cost of the former is not bounded in general situations.

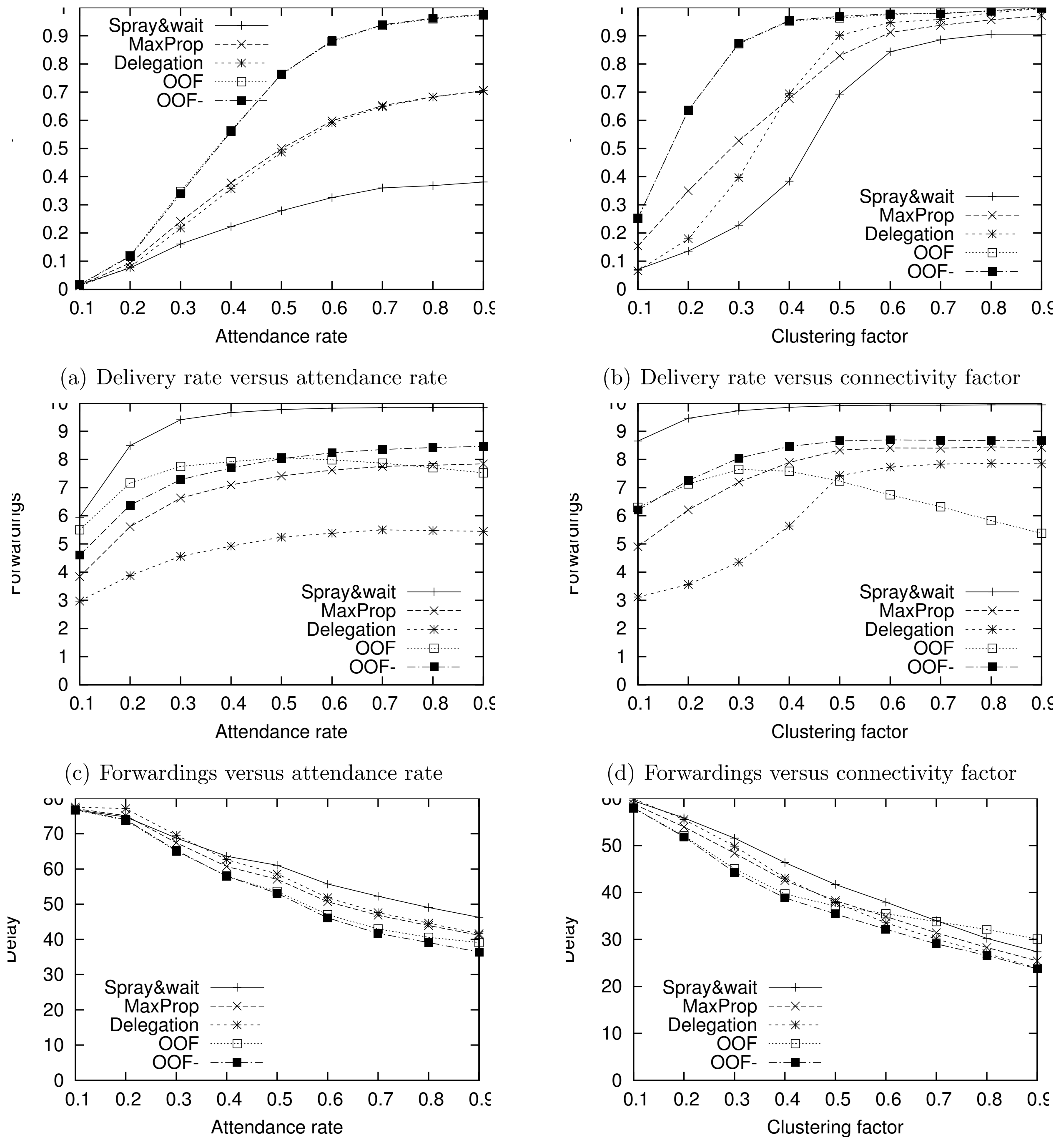
OOF and OOF- also have the lowest delay, which is (as shown in Figures 1(e), 1(f), 2(e), and 2(f)) on average 10% lower than MaxProp* and Delegation, and 20% lower than Spray-and-wait. OOF- has a slightly smaller delay than OOF because its sends a slightly higher amount of messages than OOF.

3.2 UMassDieselNet trace

In the main paper, we have shown the simulation method, simulation settings, as well as the simulation results and the discussion of routing in the UMassDieselNet trace with full routing information. Here, we will show how the compared protocols perform when routing information is not perfect in three ways: (1) routing information is incomplete; (2) routing information is false; (3) routing information is stale. In the following, we will explain how we produce these simulations and present their simulation results, respectively.

3.2.1 Incomplete routing information

We evaluate the routing performance of the protocols with incomplete routing information in the UMassDieselNet trace in the default setting, as shown in Table 1 of the main paper. We simulate incomplete routing information by setting a percentage of inter-meeting times between nodes as unknown. When the inter-meeting time between i and j is unknown, we set $I_{i,j} = \infty$,



(a) Delivery rate versus attendance rate

(b) Delivery rate versus connectivity factor

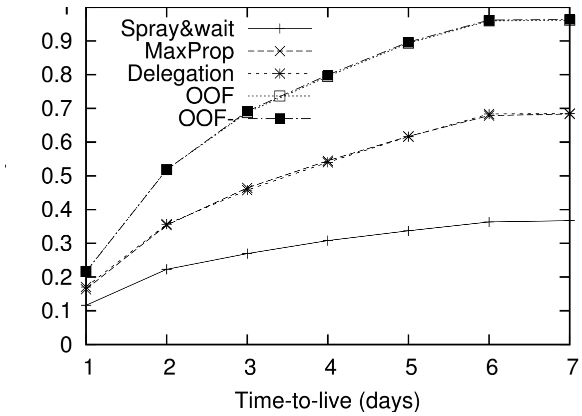
(c) Forwardings versus attendance rate

(d) Forwardings versus connectivity factor

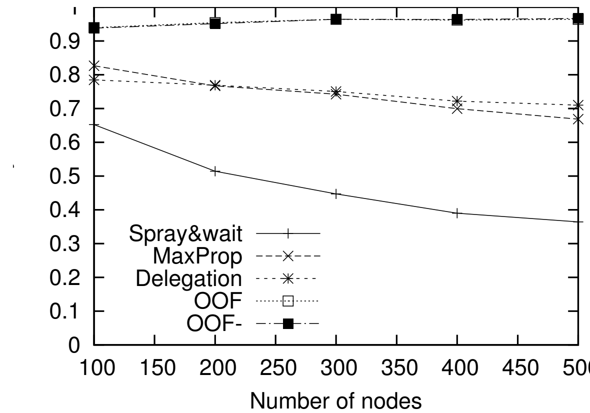
(e) Delay versus attendance rate

(f) Delay versus connectivity factor

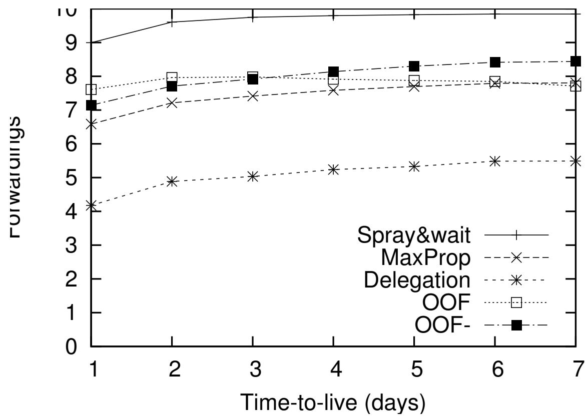
Figure 1: Delivery rate, delay, and number of forwardings versus attendance rate and connectivity factor in the NUS trace.



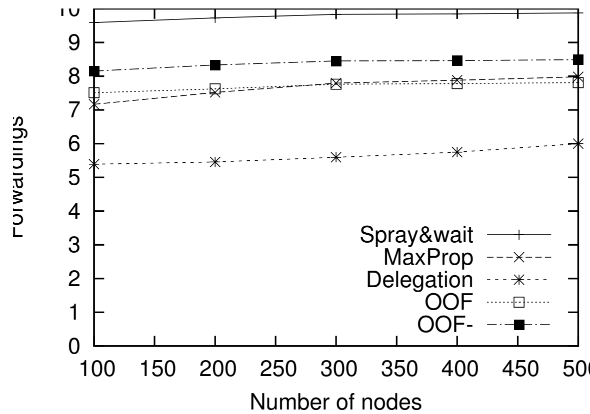
(a) Delivery rate versus time-to-live (days)



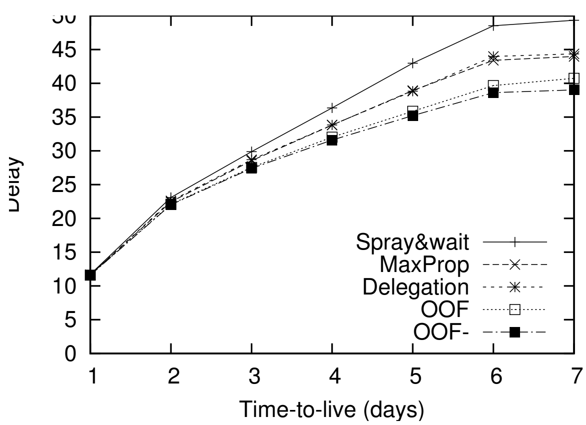
(b) Delivery rate versus number of nodes



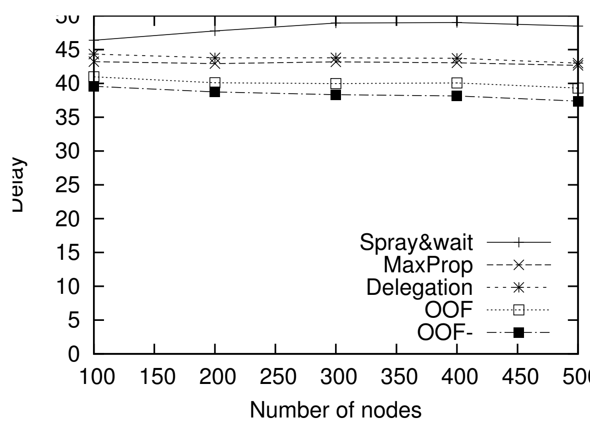
(c) Forwardings versus time-to-live (days)



(d) Forwardings versus number of nodes

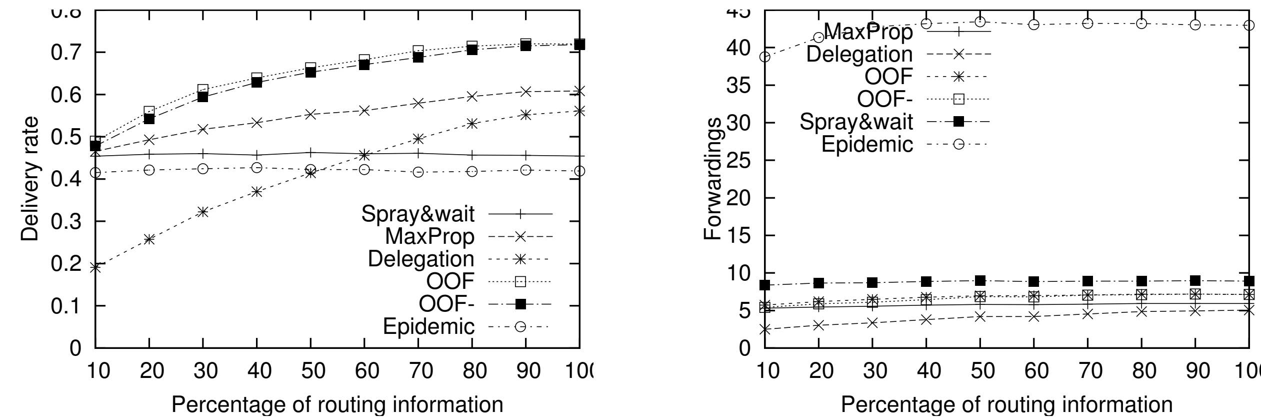


(e) Delay versus time-to-live (days)



(f) Delay versus number of nodes

Figure 2: Delivery rate, delay, and number of forwardings versus message time-to-live and number of nodes in the NUS trace.



(a) Delivery rate versus percentage of routing information

(b) Forwardings versus percentage of routing information

Figure 3: Delivery rate and delay versus percentage of routing information in the UMassDieselNet trace.

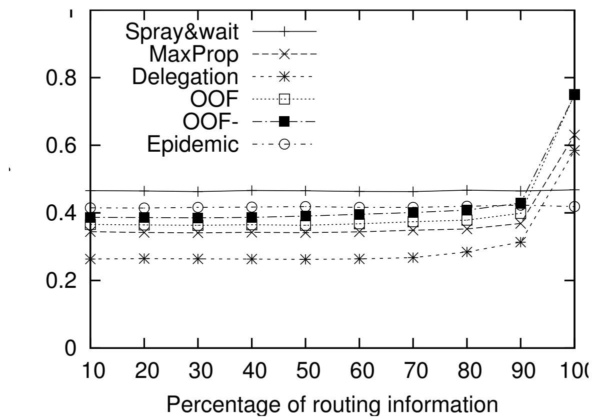
$M_{i,j} = 0$, $P_{i,j,0,T_r} = 0$. and $D_{i,j,0} = \infty$. The percentage of routing information ranges from 10% to 90%, in increments of 10%. For example, when the percentage of routing information is 10%, we randomly set 90% of the mean inter-meeting times to ∞.

As shown in Figure 3(a), as the percentage of routing information lowers, OOF and OOF-behave as Spray-and-wait. The figure shows that the routing performance of OOF degrades very slowly as routing information decreases. With only 50% of routing information, the delivery rate of OOF is 80% of that with full routing information.

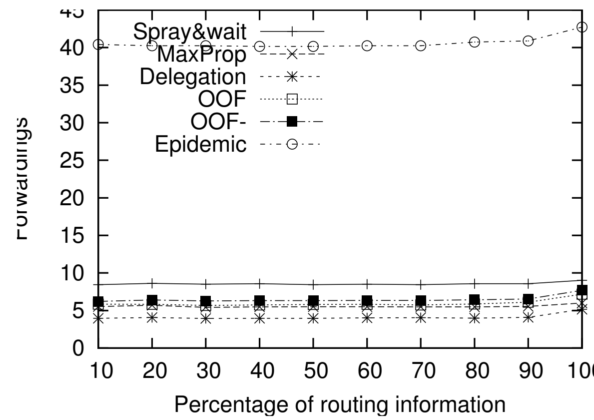
Figure 3(b) shows that the number of forwardings increases gradually as the percentage of routing information increases in all of the algorithms, including Epidemic. This is because we only calculate the number of forwardings for the messages that are delivered by all of the routing algorithms. As the percentage of routing information decreases, the algorithms using forwarding metrics are more reluctant to forward messages. As a result, as the percentage of routing information decreases, only the messages, which require less forwardings to be delivered, get delivered by all of the algorithms.

3.2.2 False routing information

We investigate how false routing information can affect our algorithms and the compared algorithms. We produce false information by swapping a percentage p of the inter-meeting times



(a) Delivery rate versus percentage of routing information



(b) Forwardings versus percentage of routing information

Figure 4: Delivery rate and delay versus percentage of true routing information in the UMass-DieselNet trace.

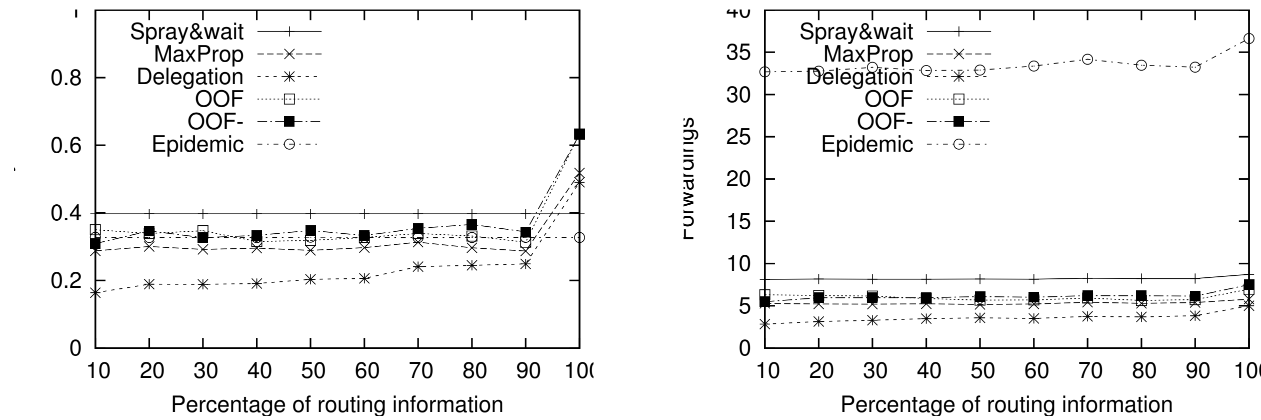
of each node, with p ranging from 0% to 90%. Specifically, for each node i and each of its inter-meeting time $I_{i,j}$, the value of $I_{i,j}$ and another randomly selected inter-meeting times $I_{i,k}$ ($j \neq k$) are swapped with probability p.

As shown in Figure 4(a), as the percentage of true routing information lowers, the delivery rates of all of the algorithms using contact history decrease very quickly: the average delivery rate drops to around 40% with a 10% loss of the true routing information. While all routing algorithms are very vulnerable to false routing information, the order of their performance is preserved as the percentage of true information decreases. Figure 4(b) shows that the number of forwardings is hardly affected by the false routing information.

3.2.3 Stale routing information

We investigate how stale routing information can affect our algorithms and the compared algorithms. In these simulations, staleness is defined as a percentage of the latest information being removed. The percentage of information ranges from 10% to 100%, where 10% means that 90% of the latest information is removed, and the algorithm uses the oldest 10% of the routing information. Simulations are run on the latest 10% trace.

As shown in Figure 5(a), as the percentage of true routing information lowers, the delivery rates of all of the algorithms using contact history decrease very quickly: the average delivery



(a) Percentage of routing information versus Delivery rate

(b) Percentage of routing information versus Forwardings

Figure 5: Delivery rate and delay versus percentage of latest routing information in the UMass-DieselNet trace.

rate drops to around 50% with a 10% loss of the latest routing information. While all routing algorithms are very vulnerable to stale routing information, the order of their performance is preserved as the percentage of true information decreases. Figure 5(b) shows that the number of forwardings is hardly affected by the stale routing information.

3.3 Summary of evaluation

Evaluation results show that, compared with other algorithms, OOF and OOF- have higher delivery rates and smaller delays under a bounded number of forwardings per message. For example, in the NUS student contact traces, the delivery rate of OOF and OOF- are very close, which is around 40% greater than MaxProp* and Delegation, and 15% greater than Spray-and-wait. On the other hand, the number of forwardings of OOF and OOF- is equal or moderately larger than the other protocols within the bound of number of forwardings. We conclude that the proposed routing algorithms perform well when the routing nodes have regular mobility, preferred locations, or preferred contacts, and perform moderately otherwise.

Simulation results with incomplete routing information, using the UMassDieselNet traces, show that the routing performance of OOF degrades very slowly as routing information decreases. With only 50% of the routing information, the delivery rate of OOF is 80% of that with full routing information. On the other hand, simulation results with false routing information and stale routing information show that the proposed routing protocols, as well as the other compared routing protocols, are very vulnerable to incorrect routing information. While their routing performance drops, the proposed routing protocols keep better than those others using historical routing information.

References

[1] Optimal Stopping and Applications. http://www.math.ucla.edu/∼tom/Stopping/Contents.html.

[2] A. Vahdat and D. Becker. Epidemic Routing for Partially-connected Ad Hoc Networks. In *Technical Report, Duke University*, 2002.

[3] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proc. of ACM WDTN*, 2005.

[4] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation Forwarding. In *Proc. of ACM MobiHoc*, 2008.

[5] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay Tolerant Networking Architecture. In *Internet draft: draft-irrf-dtnrg-arch.txt, DTN Research Group*, 2006.

[6] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. of ACM SIGCOMM*, 2004.

[7] S. Merugu, M. Ammar, and E. Zegura. Routing in Space and Time in Network with Predictable Mobility. In *Technical report: GIT-CC-04-07, College of Computing, Georgia Tech*, 2004.

[8] C. Liu and J. Wu. Scalable Routing in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2007.

[9] J. Haas, J. Y. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. In *Proc. of IEEE INFOCOM*, 2002.

[10] A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *Lecture Notes in Computer Science*, 3126:239–254, August 2004.

[11] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Proc. of IEEE PerCom*, 2007.

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

[13] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM SIGCOMM*, 2007.

[14] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *Proc. of ACM MobiHoc*, 2003.

[15] M. Grossglauser and M. Vetterli. Locating Nodes with Ease: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion. In *Proc. of IEEE INFOCOM*, 2003.

[16] E. Daly and M. Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *Proc. of ACM MobiHoc*, 2007.

[17] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2008.

[18] J. Leguay, T. Friedman, and V. Conan. DTN Routing in a Mobility Pattern Space. In *Proc. of ACM WDTN*, 2005.

[19] C. Liu and J. Wu. Routing in a Cyclic MobiSpace. In *Proc. of ACM MobiHoc*, 2008.

[20] L. Song and D. F. Kotz. Evaluating Opportunistic Routing Protocols with Large Realistic Contact Traces. In *Proc. of ACM CHANTS*, 2007.

[21] U. Acer, S. Kalyanaraman, and A. Abouzeid. Weak State Routing for Large Scale Dynamic Networks. In *Proc. of ACM MobiCom*, 2007.

[22] CRAWDAD data set. Downloaded from http://crawdad.cs.dartmouth.edu/.

[23] P. Chaporkar, A. Proutiere, and H. Asnani. Scheduling with Limited Information in Wireless Systems. In *Proc. of ACM MobiHoc*, 2009.

[24] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. of ACM MobiCom*, 2007.

[25] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and Implications of Student Contact Patterns Derived from Campus Schedules. In *Proc. of ACM MobiCom*, 2006.