An Optimal Joint Scheduling and Drop Policy for Delay Tolerant Networks

Amir Krifa, Chadi Barakat, Thrasyvoulos Spyropoulos

An Optimal Joint Scheduling and Drop Policy for Delay Tolerant Networks

Amir Krifa*†, Chadi Barakat†, Thrasyvoulos Spyropoulos†‡
†Project-Team Planète, INRIA Sophia-Antipolis, France
*National School of Computer Sciences (ENSI), Tunisia
‡Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
Emails: {Amir.Krifa, Chadi.Barakat}@sophia.inria.fr,spyropoulos@tik.ee.ethz.ch

***Abstract*—Delay Tolerant Networks are wireless networks where disconnections may occur frequently due to propagation phenomena, node mobility, and power outages. In order to achieve data delivery in such challenging networking environments, researchers have proposed the use of *store-carry-and-forward* protocols: there, a node may store a message in its buffer and carry it along for long periods of time, until an appropriate forwarding opportunity arises. Multiple message replicas are often propagated to increase delivery probability. This combination of long-term storage and replication imposes a high storage and bandwidth overhead. Thus, efficient scheduling and drop policies are necessary to: (i) decide on the order by which messages should be replicated when contact durations are limited, and (ii) which messages should be discarded when nodes' buffers operate close to their capacity.**

In this paper, we propose an efficient joint scheduling and drop policy that can optimize different performance metrics, such as the average delivery ratio and the average delivery delay. Using the theory of encounter-based message dissemination, we first propose an optimal policy based on global knowledge about the network. Then, we introduce a distributed algorithm that uses statistical learning to approximate the global knowledge required by the optimal policy, in practice. Using simulations based on a synthetic mobility model and a real mobility trace, we show that our policy based on statistical learning successfully approximates the performance of the optimal policy in all considered scenarios. At the same time, both our optimal policy and its distributed variant outperform existing resource allocation schemes for DTNs, such as the RAPID protocol [1], both in terms of average delivery ratio and delivery delay.

I. Introduction

The traditional view of a network 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 periodically shutting down and waking up, etc, connectivity in many wireless networks is, more often than not, intermittent. Despite this limited or episodic connectivity, many emerging wireless applications could still be supported. Some examples are the low-cost Internet provision in remote or developing communities [2], [3], vehicular networks (VANETs) for dissemination of location-dependent information (e.g. local ads, traffic reports, parking information, etc) [4], underwater networks [5], etc.

To enable services to operate even under these challenging conditions, researchers have proposed a new networking paradigm, often referred to as Delay Tolerant Networking (DTN [6]), based on the *Store-carry-and-forward* routing principle [2]. One of the most popular DTN routing protocols, Epidemic routing [7], as well as many of its variants, replicates messages during transfer opportunities ("contacts") searching multiple paths towards a destination, in parallel. However, the naive flooding of Epidemic routing wastes resources and can severely degrade performance. Other protocols attempt to limit replication or otherwise clear useless messages in various ways, for example: (i) using past meeting information [8]; (ii) removing useless messages using acknowledgments of delivered data [9]; and (iii) bounding the number of replicas of a message [10].

Despite a large amount of effort invested in the design of efficient routing algorithms for DTNs, there has not been a similar focus on drop and scheduling policies. Yet, the combination of long-term storage and the, often expensive, message replication performed by many DTN routing protocols [7], [11] imposes a high bandwidth and storage overhead on wireless nodes [12]. Moreover, the data units disseminated in this context, called *bundles*, are self-contained, application-level data units, which can often be large [6]. It is evident that, in this context, node buffers will very likely run out of capacity. For the same reasons, when mobility results in short contacts between nodes, available bandwidth could be insufficient to communicate all intended messages. Consequently, efficient drop policies are necessary to decide which message(s) should be discarded when a node's buffer is full, together with efficient scheduling policies to decide which messages should be chosen when bandwidth is limited, *regardless of the specific routing algorithm used*.

In this paper, we try to solve this problem in its foundation. We develop a theoretical framework based on Epidemic message dissemination [9], [13], [14] that takes into account all information that are relevant for encounter-based (or store-carry-and-forward) message delivery. Based on this theory, we first propose an optimal joint scheduling and drop policy, GBSD (Global khnowledge Based Scheduling and Drop) that can maximize the average delivery ratio or minimize the average delivery delay. GBSD uses global information about the network to derive a per-message utility for a given routing metric, and thus is difficult to implement in practice. In order to amend this, we propose a second policy, HBSD (History Based Scheduling and Drop), employing a distributed (local)

algorithm that uses statistical learning in order to estimate information about the global status of the network that can be used later to calculate message utility. To our best knowledge, the recently proposed RAPID protocol [15] is the only effort aiming at scheduling (and to a lesser extend message drop) using such a theoretical framework, but is sub-optimal in a number of respects, as we explain later. Simulations results based on both synthetic mobility as well as traces, show that our statistical learning policy HBSD outperforms existing schemes, achieving close-to-optimal performance in all considered scenarios.

The rest of this paper is organized as follow. Section II describes the current state-of-the art in terms of buffer management and scheduling in DTNs. In Section III, we establish theoretically a "reference", optimal joint scheduling and drop policy that uses global knowledge about the network. Then, we present in Section IV a learning process that enables us to approximate the global network state required by the reference policy. Section V describes the experimental setup and the results of our performance evaluation. Finally, we summarize our conclusions and discuss future work in Section VI.

II. State of the art

Several solutions have been proposed to handle routing in DTNs. Yet, an important issue that has been largely disregarded by the DTN community is the impact of buffer management and scheduling policies on the performance of the system. In [16], Zhang et al. present an analysis of buffer-constrained *Epidemic* routing, and evaluate some simple drop policies like drop-front and drop-tail. The authors conclude that drop-front, and a variant of it giving priority to source messages, outperform drop-tail in the DTN context. A somewhat more extensive set of combinations of *heuristic* buffer management policies and routing protocols for DTNs is evaluated in [17], confirming the performance of drop-front. However, all these policies are simple and/or heuristic that neither aim at optimality in the DTN context nor do they address scheduling. In a different work [18], we address the problem of optimal drop policy only using a similar analytical framework, and have compared it extensively against the policies described in [16] and [17]. Due to space limitations, the comparison between various drop policies is not repeated here. We note only that the queue drop component of our scheme outperforms all policies discussed there, and thus focus here on the *joint scheduling and drop problem*, for which we believe the RAPID protocol [1] represents the state-of-the-art.

RAPID is the first protocol to explicitly assume both bandwidth and (to a lesser extent) buffer constraints exist, and to handle the DTN routing problem as an optimal resource allocation problem, given some assumption regarding node mobility. As such, it is the most related to our own proposal, and we will compare directly against it. Despite the elegance of the approach, and performance benefits demonstrated compared to well-known routing protocols, RAPID suffers from two main drawbacks as a scheduling and buffer management scheme: (i) its policy is based on non-optimal message utilities (more on this in Section III); (ii) in order to derive these utilities, RAPID requires the flooding of information about all the replicas of a given message in the queues of all nodes in the network; further, the propagated information may be stale (a problem that the authors also note) due to change in the number of replicas, change in delivery delays, or if the message is delivered but acknowledgements have not yet propagated. In this paper, we propose a policy that fixes both (i) and (ii), and outperforms both RAPID and existing policies.

GBSD (Node x, Node y):

Scheduling (Transfer opportunity with limited bandwidth) :

1. **Replicates messages to Y in decreasing order of their utility.**
2. **End transfer when out of radio range or all messages replicated.**

Drop (Y is receiving a new message while his buffer is congested):

1. **Drop the message with the smallest utility among all buffered messages including the new received one (except source messages).**
2. **If all buffered messages and the new received one are source messages, drop the one having the smallest utility.**

Fig. 1. GBSD's scheme.

III. Optimal Joint Scheduling and Drop Policy

In this section, we assume both limited storage and bandwidth. We first make a few assumptions regarding some generic mobility characteristics of the nodes, and then embark on finding theoretically the optimal policy, based on these characteristics. This policy, GBSD (Global Knowledge based Scheduling and Drop), uses global knowledge about the state of each message in the network (number of replicas). Hence, it is difficult to be implemented, in practice, and will serve as a point of reference.

A. Problem description (assumptions)

In the context of DTNs, message transmissions occur only when nodes encounter each other. Thus, *the time elapsed between node meetings is the basic delay component*. This also implies that changes in the nodes' buffers also occur only during node encounters. This inter-encounter time between nodes depends on the value of a particular property of the mobility model assumed, namely the *meeting* time [19], [18][1].

To formulate the optimal policy problem, we do not make any specific assumption about the used mobility model. Our only requirement is that *the meeting time of the mobility model is exponentially distributed or has at least an exponential tail, with parameter* $\lambda = \frac{1}{E[U]}$, where $E[X]$ denotes the expectation of a random variable X. It has been shown that many popular mobility models like Random Walk [19], Random Waypoint and Random Direction [14], [13] have such a property. Moreover, it has recently been argued that inter-meeting times observed in many traces may also exhibit an exponential tail [20].

[1]If some of the nodes in the network are static, then one needs to add the *hitting* time between a mobile node and a static node, instead. For simplicity, we assume here that all nodes are mobile and we refer only to meeting times thereafter. Our theory can be easily modified to account for static nodes.

TABLE I
NOTATION

Variable	Description
L	Number of nodes in the network
K(t)	Number of distinct messages in the network at time t
TTL_i	Initial Time To Live for message i
R_i	Remaining Time To Live for message i
$T_i = TTL_i - R_i$	Elapsed Time for message i. It measures the time since this message was generated by its source
$n_i(T_i)$	Number of copies of message i in the network after elapsed time T_i
$m_i(T_i)$	Number of nodes (excluding source) that have *seen* message i since its creation until elapsed time T_i
λ	Meeting *rate* between two nodes under the given mobility model; $\lambda = \frac{1}{E[U]}$ where $E[U]$ is the average meeting time

Given the above problem setting and a routing metric, our policy GBSD derives a per-message utility. This utility captures the *marginal value* of a given message copy for the overall routing process, and with respect to the chosen optimization metric. As described in Figure 1, GBSD has two core components: (i) Scheduling—determines which messages to replicate at a limited transfer opportunity given their utilities, and (ii) Drop—decides which messages to drop when a node exhausts all available storage. We derive here such a utility for two popular metrics: maximizing the average delivery ratio, and minimizing the average delivery delay.

In Table I, we summarize the various quantities and notations we use throughout the paper.

B. *Maximizing the average delivery ratio*

To maximize the average delivery ratio, the per-message utility used by GBSD is defined by the following theorem:

Theorem III.1. *Delivery-Ratio: Let us assume there are K messages in the network with elapsed time T_i for the message i at the moment when the drop or replication decision by a node is to be taken. For each message $i \in [1, K]$, let $m_i(T_i)$ and $n_i(T_i)$ be the number of nodes that have "seen" the message since it's creation[2] (excluding the source), and those who have a copy of it at this instant $(n_i(T_i) \leqslant m_i(T_i)+1)$, respectively. To maximize the average delivery ratio of all messages, a DTN node should apply the GBSD policy using the following utility for each message i:*

$$(1 - \frac{m_i(T_i)}{L-1})\lambda R_i \exp(-\lambda n_i(T_i)R_i) \quad (1)$$

Proof: We know that the meeting time between nodes is exponentially distributed with parameter λ. The probability that a copy of a message i will not be delivered by a node is then given by the probability that the next meeting time with the destination is greater than the remaining time R_i. This is equal to $\exp(-\lambda R_i)$.

Knowing that message i has $n_i(T_i)$ copies in the network, and assuming that the message has not yet been delivered, we can derive the probability that the message itself will not be delivered (i.e. none of the n_i copies gets delivered):

$$\prod_{i=1}^{n_i(T_i)} \exp(-\lambda R_i) = \exp(-\lambda n_i(T_i) R_i).$$

Here, we have not taken into account that more copies may be created in the future through new node encounters (and thus this policy is to some extent "greedy"). Predicting future encounters and the effect of further replicas complicates the problem significantly. Nevertheless, the same assumption is performed for all messages equally and thus can justify the relative comparison between the delivery probabilities for different messages. Unlike RAPID [1], we take into consideration what has happened in the network since the message generation. Given that all nodes including the destination have the same chance to see the message, the probability that a message i has been already delivered is equal to:

$$P\{\text{message } i \text{ already delivered}\} = \quad m_i(T_i)/(L-1).$$

So, if we take at instant t a snapshot of the network, the global delivery ratio for the whole network will be:

$$DR = \sum_{i=1}^{K(t)} \left[(1 - \frac{m_i(T_i)}{L-1}) * (1 - \exp(-\lambda n_i(T_i) R_i)) + \frac{m_i(T_i)}{L-1} \right]$$

In case of congestion or limited transfert opportunity, a DTN node should take respectively a drop or a replication decision, that leads to the best gain in the global delivery ratio DR. To find the local optimal decision, we differentiate DR with respect to $n_i(T_i)$:

$$\Delta(DR) = \sum_{i=1}^{K(t)} \left[(1 - \frac{m_i(T_i)}{L-1}) \lambda R_i \exp(-\lambda n_i(T_i) R_i) * \triangle n_i(T_i) \right]$$

Our aim is to maximize $\Delta(DR)$. We know that: $\Delta n_i(T_i) = -1$ if we drop an already existing message i from the buffer, $\Delta n_i(T_i) = 0$ if we don't drop an already existing message i from the buffer and $\Delta n_i(T_i) = +1$ if we keep and store the new received message i or replicate and forward an already buffered message i to another node. Based on that, the per-message utility that should be used by GBSD to maximize the average delivery ratio is Eq.(1).

■

C. *Minimizing the average delivery delay*

We now turn our attention to minimizing the expected delivery delay over all messages in the network. The following Theorem derives the optimal per-message utility, for the same setting and assumptions as Theorem III.1.

Theorem III.2. *To minimize the average delivery delay of all messages, a DTN node should apply the GBSD policy using the following utility for each message i:*

$$\frac{1}{n_i(T_i)^2 \lambda}(1 - \frac{m_i(T_i)}{L-1}) \quad (2)$$

Proof: Let us denote the delivery delay for message i with random variable X_i. This delay is set to zero if the message

[2]We say that a node *A* has "seen" a message i, when A had received a copy of message i sometime in the past, regardless of whether it still has the copy or if it has already removed it from the buffer.

has been already delivered. Then, the total expected delivery delay (D) is given by,

$$D = \sum_{i=1}^{K(t)} \left[\frac{m_i(T_i)}{L-1} * 0 + (1 - \frac{m_i(T_i)}{L-1}) * E[X_i | X_i > T_i] \right].$$

We know that the time until the first copy of the message i reaches the destination follows an exponential distribution with mean $1/(n_i(T_i)\lambda)$. It follows that,

$$E[X_i | X_i > T_i] = T_i + \frac{1}{n_i(T_i)\lambda}.$$

Then, as for the delivery ratio, we differentiate D with respect to $n_i(T_i)$ and find Eq.(2). ■

Thus, the per-message utility with respect to delivery delay is different than the one for the delivery ratio. This implies (naturally) that both metrics cannot be optimized concurrently.

IV. Using learning to approximate global knowledge in practice

In order to optimize a specific routing metric using GBSD, we need global information about the network and the "spread" of messages. In particular, for each message present in a node's buffer, we need to know the values of $m_i(T_i)$ and $n_i(T_i)$, which are respectively the number of nodes that have seen the message and those that have a copy of it. Unfortunately, this is not feasible in practice due to intermittent network connectivity and the long time it takes to flood buffer status information across DTN nodes. Note that RAPID [1] assumes, for simplicity, that the global view obtained by flooding (or a secondary, "instantaneous" channel) is sufficient to achieve significant performance gains over existing DTN routing protocols. Our experiments prove that the impact of the flooding delay is non negligible and that a much better gain can be realized if we find estimators for the metrics involved in the calculation of message utilities, namely m and n.

To this end, we have developed and implemented a learning process that allows a DTN node to gather knowledge about the global network history by making in-band exchanges with other nodes. As in the case of RAPID [1], each node maintains a list of encountered nodes and the state of each message carried by them, which could be 0 if the message was in the node's buffer or 1 if the message was seen but deleted due to congestion. Each node maintains the time of the last list update and only sends the list if it has been updated since the last exchange. Using this information exchanged among nodes, all DTN nodes start to have after a while the same history on global network information. Yet, unlike RAPID's approach that uses the actual, explicit values of $m_i(T)$ and $n_i(T)$ *for a specific message* i at an elapsed time T, *we look at what happens, on average, for all messages after an elapsed time* T. In other words, the $m_i(T)$ and $n_i(T)$ values for message i at elapsed time T are estimated using measurements of m and n for the same elapsed time T *but measured for (and averaged over) all other older messages*. These estimators are then used in the evaluation of the per-message utility, and can approximate the actual global message state considerably more successfully, and with less overhead.

Let's denote by $\hat{n}(T)$ and $\hat{m}(T)$ the estimators for $n_i(T)$ and $m_i(T)$ of message i. For the purpose of the analysis, we suppose that the variables $m_i(T)$ and $n_i(T)$ at elapsed time T are instances of the random variables $N(T)$ and $M(T)$. We develop our estimators $\hat{n}(T)$ and $\hat{m}(T)$ so that when plugged into the GBSD's delivery ratio and delay per-message utility calculated in Section III, we get two new per-message utilities that can be used by a DTN node without any need for global information about messages. This results in a new scheduling and drop policy, called HBSD (History Based Scheduling and Drop), a deployable variant of GBSD that uses the same algorithm described in Figure 1, yet with per-message utility values calculated using estimates of m and n.

A. *Calculating estimators* $\hat{n}(T)$ *and* $\hat{m}(T)$ *for the average delivery ratio metric*

When the global information is unavailable, one can calculate the average delivery ratio of a message over all possible values for $M(T)$ and $N(T)$, and then try to minimize it. We want our estimators to be unbiased, that is, the average delivery ratio to be the same as that obtained when using our estimators of m and n instead of their real values. Differently speaking, we don't want our estimation of the global information to affect the value of the average message delivery ratio. In the framework of the GBSD, this can be written as:

$$E[(1 - \frac{M(T)}{L-1}) * (1 - \exp(-\lambda N(T) R_i)) + \frac{M(T)}{L-1}] =$$
$$(1 - \frac{\hat{m}(T)}{L-1}) * (1 - \exp(-\lambda \hat{n}(T) R_i)) + \frac{\hat{m}(T)}{L-1}$$

By plugging in the per-message utility in Eq.(1) any values of $\hat{n}(T)$ and $\hat{m}(T)$ that verify this equality, one can make sure that the obtained policy minimizes the average delivery ratio. This is exactly our purpose. Suppose now that the best estimator for $\hat{m}(T)$ is its average, i.e., $\hat{m}(T) = \bar{m}(T) = E[M(T)]$ (we refer the reader to [18], for a more detailed justification of our choice). Then, we extract $\hat{n}(T)$ from the above equality and we replace in Eq.(1) to obtain the following per-message utility:

$$\lambda R_i E[(1 - \frac{M(T)}{L-1}) \exp(-\lambda R_i N(T))]$$

Unlike Eq.(1), this new per-message utility is a function of past history of messages and so can be calculated locally. It maximizes the average message delivery ratio calculated over a large number of messages. Except when the number of messages is not large for the law of large numbers to work, our history based policy should give the same result as that of using the real global network information. This will be illustrated later by our simulation results.

B. *Calculating estimators* $\hat{n}(T)$ *and* $\hat{m}(T)$ *for the average delivery delay metric*

Similar to the case of delivery ratio, we calculate the estimators $\hat{n}(T)$ and $\hat{m}(T)$ in such a way that the average delay

is not affected by the estimation. This gives the following per-message utility specific to HBSD,

$$\frac{E[\frac{L-1-M(T)}{N(T)}]^2}{\lambda(L-1)(L-1-\bar{m}\ (T))}$$

Unlike GBSD's per-message delivery delay utility, this new utility is only a function of the locally available history of old messages and is independent of the actual global network state.

V. Performance Evaluation

A. Experimental setup

To evaluate our new scheme, we have added an implementation of the DTN architecture to the Network Simulator NS-2. This implementation includes (i) the Epidemic routing protocol with *FIFO* and *drop-tail* for scheduling and message drop in case of congestion, respectively, (ii) the RAPID routing protocol based on flooding (i.e. no side-channel) as described, to our best understanding, in [1], (iii) a new version of the Epidemic routing protocol enhanced with our optimal joint scheduling and drop policy (GBSD), and another version using our statistical learning distributed algorithm (HBSD). The VACCINE mechanism described in [16] is used with all solutions to "clean up" the network after message delivery.

In our simulations, each node uses a wireless communication channel 802.11b of range 100 meters and of bandwidth capacity of 1Mbits/s. Our simulations are based on two mobility patterns, a synthetic one based on the Random Waypoint model, and a real-world mobility trace that tracks San Francisco's Yellow Cab taxis [21]. Many cab companies outfit their cabs with *GPS* to aid in rapidly dispatching cabs to their costumers. The Cabspotting system [21] talks to the Yellow Cab server and stores the data in a database. We used an API provided by the Cabspotting system to extract the taxi mobility trace then we converted it into a format readable by the NS-2 simulator. Note that this trace describes taxis' positions according to the *GPS* cylindrical coordinates (*Longitude*, *Latitude*). In order to use it as input to the NS-2 simulator, we had to implement a tool based on the Mercator [22] cylindrical map projection that permits to convert cylindrical coordinates into plane coordinates.

To each source node, we have associated a CBR (Constant Bit Rate) application, which chooses randomly from [0, TTL] the time to start generating messages of 85KB for a randomly chosen destination. Other message sizes were also considered but for lack of space we only show results for this value. Unless otherwise stated, we associate to each node a buffer with a capacity of 10 messages.

We compare the performance of the various routing protocols using the following two metrics: the average delivery ratio and average delivery delay of messages in the case of infinite TTL. Note, that the evaluation of the HBSD policy requires to wait until the different nodes collect enough history to be able to calculate their estimators, and thus include an initial "warm-up" period before starting to account for HBSD.

As a final note, we have chosen to compare all policies in the context of Epidemic routing, which uses up the largest amount of resources. However, we believe that similar conclusions could be drawn if the various policies were applied in other routing protocols, as well, operated in a regime of limited bandwidth or buffer space. We defer this investigation for future work.

B. Performance evaluation for delivery ratio

First, we compare the delivery ratio of all protocols for the two scenarios shown in Table II. Figures 2 and Figures 3 show the delivery ratio for the Taxi trace for the case of both limited bandwidth and buffer, and the case of limited bandwidth and unlimited buffer, respectively. The number of sources is changed to cover different congestion levels.

TABLE II
Simulation parameters

Mobility pattern:	RWP	Taxi Trace
Simulation's Duration(s):	5000	36000
Simulation' Area (m^2):	1500*1500	-
Number of Nodes:	40	40
Average Speed (Km/H):	6	-
TTL(s):	750	7200
CBR Interval(s):	200	2100

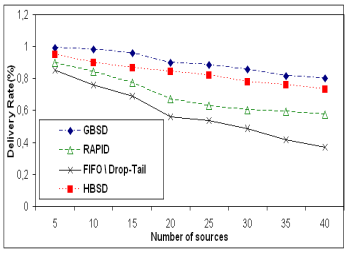

Fig. 2. Average delivery ratio for limited buffer and limited bandwidth.

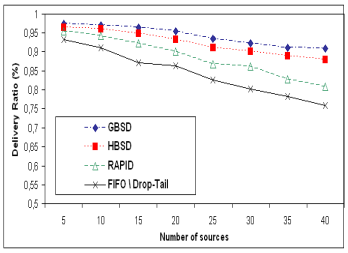

Fig. 3. Average delivery ratio for unlimited buffer and limited bandwidth.

TABLE III
Simulation results for the Random Waypoint mobility model in the case of unlimited buffer and limited bandwidth

Mobility pattern:	GBSD	HBSD	RAPID	FIFO\drop-tail
Delivery Ratio (%):	0,83	0,77	0,65	0,54
Delivery Delay (s):	519,75	532	682	775

TABLE IV
Simulation results for the Random Waypoint mobility model in the case of both limited buffer and bandwidth

Mobility pattern:	GBSD	HBSD	RAPID	FIFO\drop-tail
Delivery Ratio (%):	0,55	0,50	0,36	0,23
Delivery Delay (s):	1469,5	1507,47	1690,7	1970,45

From this plots, it can be seen that: the GBSD policy plugged into Epidemic routing gives the best performance for all numbers of sources. When congestion-level decreases, so does the difference between GBSD and other protocols, as

expected. Moreover, the HBSD policy also outperforms existing protocols (RAPID and Epidemic based on FIFO/drop-tail) and performs very close to the optimal GBSD. For example, for 40 sources, and in the case of limited bandwidth and buffer, HBSD's delivery ratio is 15% higher than RAPID and only 6% worse than GBSD.

Similar conclusions can be also drawn for the case of Random Waypoint mobility and 40 sources. Results for this case are summarized in Table IV and Table III.

C. *Performance evaluation for delivery delay*

To evaluate the average delivery delay metric, we keep the same simulation duration and message generation rate as those used for the delivery ratio. Based on the Taxi trace, Figures 4 and 5 depict the average delivery delay for the the case of both limited buffer and bandwidth, and the case of unlimied buffer but limited bandwidth, respectively. As in the case of delivery ratio, GBSD gives the best performance for all the considered scenarios. Moreover, the HBSD policy outperforms the two routing protocols (Epidemic based on FIFO/drop-tail, and RAPID) and performs close to GBSD. Specifically, for 40 sources and both limited buffer and bandwidth, HBSD's average delivery delay is 17% better than RAPID and only 7% worse than GBSD. For the case of unlimited buffer and limited bandwidth, HBSD performs 13% better than RAPID and 8% worse than GBSD. Table III and IV show that similar conclusions can be drawn for the delay under Random Waypoint also, with a gain up to 28% compared to RAPID.

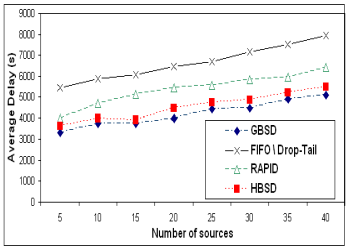

Fig. 4. Average delivery delay for both limited buffer and bandwidth.

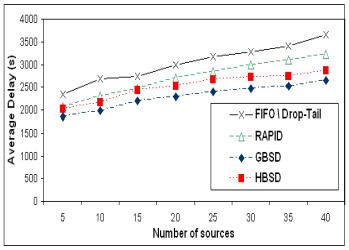

Fig. 5. Average delivery delay for unlimited buffer and limited bandwidth.

VI. Conclusion and Future Work

In this work, we investigated both the problems of scheduling and buffer management in delay tolerant networks. First, we proposed an optimal joint scheduling and buffer management policy based on global knowledge about the network state. Then, we introduced a distributed algorithm that uses statistical learning to approximate the required global knowledge of the optimal algorithm. Using simulations based on a synthetic mobility model (Random Waypoint), and a real mobility trace, the San Francisco taxi trace, we showed that our policy based on statistical learning successfully approximates the performance of the optimal algorithm in all considered scenarios. Finally, both policies (GBSD and HBSD) plugged into the Epidemic routing protocol outperform existing routing protocols with respect to delivery ratio and delivery delay, in all considered scenarios.

Note that in this work, we considered that all messages have the same size. It would be interesting to define policies that take into account different message sizes. For example, in case of congestion, the end-to-end delay versus message delivery trade-off could be influenced by the choice of dropping several small messages or one large message that occupies the entire node's buffer. The consideration of other routing protocols than Epidemic is also an interesting direction to explore.

Acknowledgments

We thank Michal Piorkowski and Wei-Jen Hsu for pointing us out to the San Francisco's taxi cab mobility traces.

References

[1] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *Proc. ACM SIGCOMM*, August 2007.
[2] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of ACM SIGCOMM*, Aug. 2004.
[3] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *IEEE Computer*, 2004.
[4] P. Basu and T. Little, "Networked parking spaces: architecture and applications," in *IEEE Vehicular Technology Conference (VTC)*, 2002.
[5] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2006.
[6] "Delay tolerant networking research group," *http://www.dtnrg.org*.
[7] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-200006, 2000.
[8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *Proc. IEEE INFOCOM*, 2006.
[9] Z. J. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks." *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 27–40, 2006.
[10] T. Spyropoulos, K. Psounis, and C. Raghavendra, "An efficient routing scheme for intermittently connected mobile networks," *ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN-05)*, 2005.
[11] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, no. 3, 2003.
[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *to appear in Transactions on Networking*, Feb. 2008.
[13] R.Groenevelt, G. Koole, and P. Nain, "Message delay in manet (extended abstract)," in *Proc. ACM Sigmetrics*, 2005.
[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of ACM/IEEE MOBIHOC*, 2006.
[15] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM*, 2007.
[16] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," in *Proceedings of IFIP Networking*, 2006.
[17] A. Lindgren and K. S. Phanse, "Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks," in *Proc. of IEEE COMSWARE*, January 2006.
[18] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," INRIA, Tech. Rep. inria-00196306 (submitted to SECON'08), 2008.
[19] 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*.
[20] M. V. Thomas Karagiannis, Jean-Yves Le Boudec, "Power law and exponential decay of inter contact times between mobile devices," in *Proc. of ACM/IEEE MobiCom*, 2007.
[21] "Cabspotting project," *http://cabspotting.org/*.
[22] "The mercator projection," *http://en.wikipedia.org/wiki/Mercator_projection/*.