

Max-Contribution: On Optimal Resource Allocation in Delay Tolerant Networks

Kyunghan Lee[†], Yung Yi[†], Jaeseong Jeong[†], Hyungsuk Won[‡], Injong Rhee[‡] and Song Chong[†]

Abstract—This is by far the first paper considering joint optimization of link scheduling, routing and replication for disruption-tolerant networks (DTNs). The optimization problems for resource allocation in DTNs are typically solved using dynamic programming which requires knowledge of future events such as meeting schedules and durations. This paper defines a new notion of optimality for DTNs, called *snapshot optimality* where nodes are not clairvoyant, i.e., cannot look ahead into future events, and thus decisions are made using only contemporarily available knowledge. Unfortunately, the optimal solution for snapshot optimality still requires solving an NP-hard problem of maximum weight independent set and a global knowledge of who currently owns a copy and what their delivery probabilities are. This paper presents a new efficient approximation algorithm, called *Distributed Max-Contribution (DMC)* that performs greedy scheduling, routing and replication based only on locally and contemporarily available information. Through a simulation study based on real GPS traces tracking over 4000 taxis for about 30 days in a large city, DMC outperforms existing heuristically engineered resource allocation algorithms for DTNs.

I. INTRODUCTION

Every aspect of modern mobile wireless networks is dynamic. As radios are now attached to moving objects which may make planned, spontaneous, or random movements, the mobility of these objects governs the network state and presents diverse and highly time-varying operating conditions. With increasing density and mobility, the operating regimes of the networks exponentially widen and network connectivity may drastically change over time. Any network protocol operating in such regimes must adapt quickly to the changing conditions, from highly dense networks where link scheduling and interference mitigation are important, to sparse networks where opportunities of contacts and their durations are important. A network can be situated at any point in this space-time continuum of the network design space [1] with a varying temporal and spatial scale of changes. We call such networks collectively *space-time disruption-tolerant networks (ST-DTNs)*; while DTNs are traditionally implying sparse, disconnected networks where mobility and carry-and-forward are the only means of communication, ST-DTNs permits traditional senses of MANETs (Mobile Ad-hoc NETWORKS) and DTNs to coexist in the form of disconnected islands in any proportion of time and space.

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2009- C1090-0902-0006) and the Korea Research Council of Fundamental Science and Technology.

[†]: Department of Electrical Engineering, KAIST, South Korea. [‡]: Department of Computer Science, North Carolina State University, USA.

ST-DTNs need to solve performance issues arising from varying time scales of network state changes such as disconnection; channel quality degradation; and information inconsistency caused mostly by node mobility and inherent channel dynamics. Their network protocols must thrive in environments with partial, inconsistent, incorrect and sometimes no information about the network states and adapt to any point in the space-time design space. The information about the state of the networks, called *metadata*, includes routing tables, routing metrics, past history of meeting or contacting nodes, location information, files, and packets/bundles-in-flight. As these protocols can work well even with inconsistent, outdated and incomplete information, these protocols relieve the network of the burden to maintain consistent information; the network can now opt for “best-effort” information sharing - changing its mode of operation to “whenever convenient” from “however possible at all cost.”

Traditional DTN studies for resource allocation have focused on routing, forwarding and replication in sparse networks [2]–[8] whereas studies for resource allocation in traditional MANET have focused on interference, link scheduling and routing in dense networks with no provisioning for disconnected islands. ST-DTNs jointly consider all these notions of resources, and its resource allocation must be adaptive to the availability of specific types of resources in time and space.

In this paper, we study a joint optimization problem of link scheduling, routing and replications for a type of ST-DTNs where resources such as link budgets and opportunities of meetings and their durations are critical resources, but each node may have enough storage and battery power to allow liberal replications and exchange of packets and metadata whenever and where-ever the critical resources are left unused. Such networks are typically driven by vehicles, e.g., taxis and buses, in a large city. The information dissemination networks of taxis in Shanghai in China [9] and BuCheon in South Korea [10] are key examples of such networks.

DTN resource allocation has traditionally considered only routing [7], [11] and/or replications [2], [8] but have not tackled the issues of interference and link scheduling. Therefore, when the network state changes to MANET-like environments with a dense population of nodes, such schemes produce sub-optimal performance or are not even functional. Joint optimization of link scheduling and routing can produce more adaptive ST-DTNs. Dynamic programming has been the main means of solving optimal resource allocation for DTNs (e.g., [12], [13]). Unfortunately, dynamic programming requires nodes to be clairvoyant – assumes knowledge of future events

such as contacts and their durations, therefore precluding on-line solutions. The complexity of dynamic programming and lack of on-line efficient algorithms make such solutions impractical.

In this paper, we present more realistic optimality, called *snapshot optimality*, in ST-DTNs whose solutions perform close to the “clairvoyant optimal solutions” but rely only on contemporarily available knowledge of the networks. Snapshot optimality dictates to maximize the “contribution” of a packet being scheduled at the current instance, to improve the global utility. This notion of optimality was considered in the name of *per-packet marginal utility* in DTNs [14], but not in the context of link scheduling. However, even maximizing contribution alone requires global knowledge of who owns a copy of a packet (as multiple copies are permitted) and their delivery statistics, and also solving an NP-hard problem of maximum weight independent set for link scheduling. Therefore, more practical, approximating solutions that leverage only contemporary and local information are vital.

To construct a theoretically engineered, highly practical approximation solution for snapshot optimality, we apply a form of distributed greedy resource allocation that performs link scheduling, routing and replication decisions based only on contemporarily and locally available metadata. The result is a distributed approximation of maximizing contribution, so called *Distributed Max-Contribution* (DMC) that employs various techniques to improve the global utility using local views and operations. These techniques include metadata fusions and broadcast-based opportunistic routing.

Our simulation studies are based on detailed GPS (Global Positioning System) traces of tracking the movements of over 4000 taxies, each equipped with GPS in Shanghai [9] for about 30 days, by far the largest traces of vehicle-based networks. In the traces, taxies usually meet at intersections and each taxi has 3 to 4 interfering taxies on average with the maximum of 20, forming interference-rich, but frequently-disconnected islands of networks. As taxies move according to the destinations of passengers, there are no pre-defined schedules of taxi movement. However, we found that they have some notion of locality and hotspots which can be exploited to enable effective routing. Our trace-driven simulation study demonstrates that DMC outperforms existing DTN routing protocols that do not consider link scheduling or snapshot optimality.

II. OPTIMAL RESOURCE ALLOCATION

A. System Model

Network and traffic model. We consider a network consisting of a set \mathcal{N} of n nodes that move and meet intermittently. Two nodes v and w is said to *meet* if v is within the communication range of w , and vice versa. Every node is equipped with an infinite-size queue to store packets. A node v can copy packets from its queue to the node that v meets¹. There is a set \mathcal{F}

of F sessions (flows) that are identified by a pair of source-destination nodes. Associated with each session f is a file consisting of a set \mathcal{G}_f of equal-sized packets. We use the *packet-company* m to refer to the original packet m and its copies together. The source of a session f is responsible for transferring the packets in \mathcal{G}_f to its destination with some QoS constraints.

Resource model. Time is assumed to be slotted, indexed by $t = 0, 1, \dots$. The length of a time-slot is suitably chosen to schedule one packet and nodes are stationary. Then, network resources are represented by a finite set $\mathcal{S}(t) \subset \{0, 1\}^L$ of feasible link schedules, where L is the number of all possible links. A *feasible link schedule*, $S = (S_l \in \{0, 1\} : l = 1, \dots, L)$ is a vector representing a set of schedulable links without interference where $S_l = 1$ if the link l is scheduled, and 0 otherwise. We also use notation $l \in S$ when $S_l = 1$. Denote by $\Pi(t) \subset \mathcal{G}^L$, a set of feasible copy schedules where $\mathcal{G} = \cup_{f \in \mathcal{F}} \mathcal{G}_f$. A *feasible copy schedule* is a vector whose l -th element represents a packet that can be potentially copied if link l is scheduled. Note that a packet m can be copied from $v \in \mathcal{N}$ to $w \in \mathcal{N}$ when v holds m but w does not. Note that in a feasible copy schedule, two different packets belonging to a single packet-company can be scheduled over different links.

Interference and resource allocation. A set $\mathcal{S}(t)$ depends on interference patterns among links. We generally model interference by a $L \times L$ symmetric matrix $I = [I_{ij}]$, where $I_{ij} = 1$ means that links i and j interfere with each other. The matrix I is able to model various wireless systems, ranging from FH-CDMA (one-hop interference) to 802.11 (two-hop interference²). For ease of presentation, we assume that when a link is established by the meeting between two nodes, the link is configured to have a unit capacity, but it can be readily extended to more general cases. Resource allocation at each slot t consists of two parts: (i) *link scheduling* and (ii) *copy scheduling* where a copy schedule $\pi \in \Pi(t)$ and a link schedule $S \in \mathcal{S}(t)$ are selected. Then, the element-wise multiplication of two vectors, $\pi \times S$, represents which packets are served and copied over the links.

Objectives. The primary objectives of resource allocation are delivery ratio maximization or delay minimization. Denote by the random variable, $N_f(t, t_{\text{dl}})$, the total aggregate number of delivered packets in flow f to its destination over an interval $[t, t_{\text{dl}}]$, where t_{dl} is a given deadline (we henceforth omit t_{dl} and just use $N_f(t)$ in all notations unless confusion arises). We also denote by $D_f(t)$ the total aggregate remaining time in flow f from t to the delivery. Then, $N_f(0)$ and $D_f(0)$ correspond to the aggregate delivery ratio (until the deadline) and the total delay of flow f , respectively. The following four objectives are mainly considered.

$$\begin{aligned} \mathbf{R1. Max-Delivery} & \quad \max \sum_{f \in \mathcal{F}} \mathbb{E}[N_f(0)] \\ \mathbf{R2. Fair-Delivery} & \quad \max \min_{f \in \mathcal{F}} \mathbb{E}[N_f(0)] \end{aligned}$$

¹We also use the word ‘packet’ to refer to the copies of the original packet, unless explicitly specified otherwise.

²In the K-hop interference model, two links that are within K -hops interfere with each other.

- D1. Min-Delay** $\min \sum_{f \in \mathcal{F}} \mathbb{E}[D_f(0)]$
D2. Fair-Delay $\min \max_{f \in \mathcal{F}} \mathbb{E}[D_f(0)].$

B. Snapshot Optimality

Hardness of full optimality. Solving the optimization problems in **R1**, **R2**, **D1**, and **D2** via practical, on-line, decentralized algorithms is hard. It can be formulated by a dynamic programming (DP), often requiring a large dimensional search (i.e., curse of dimensionality) and knowledge of the future. There are studies that use DP to develop optimal solutions. However, they have been done in much simpler models and assumptions, e.g., a model without consideration of link scheduling [12], [13]. Our main interest lies in practical, on-line algorithms. To that end, rather than pursuing the “full”-optimality based on DP, we adopt a *temporal approximation* where implementable algorithms may be temporally restricted in terms of available information. In other words, we only look at system states available contemporarily and try to optimize a certain objective naturally interpreted as a *snapshot-optimal approximation* to the original problem. It is possible simply by temporally stretching the original optimization problems over the entire slots, and look at what needs to be optimized just using the information available at time t .

Objective functions. We now elaborate the snapshot-optimal problems for various objectives introduced in the subsection II-A.

(a) Max-delivery. We stretch the objective function over the entire time-interval $[0, t_{dl}]$. Then we have

$$\max_{f \in \mathcal{F}} \sum \mathbb{E}[N_f(0)] = \max_{f \in \mathcal{F}} \mathbb{E} \left[\sum_{f \in \mathcal{F}} \left(N_f(t) + \sum_{i=1}^{t-1} \Delta N_f(i) \right) \right], \quad (1)$$

where $\Delta N_f(t) \triangleq N_f(t-1) - N_f(t)$ corresponds to the number of packets in f delivered over the interval $[t-1, t]$. Recall that $N_f(t)$ is decreasing in t . In Eq. (1), the “max” operation is taken over a set of a sequence of copy and link schedules over the entire time. From (1), what we can do, given the available information at slot t , is to maximize $\mathbb{E}[\sum_{f \in \mathcal{F}} \Delta N_f(t)]$ i.e., maximize the average increase in the total number of delivered packets over $[t-1, t]$ across all sessions.

(b) Fair-delivery. Similarly to the above, we get

$$\max_{f \in \mathcal{F}} \min_{f \in \mathcal{F}} \mathbb{E}[N_f(0)] = \max_{f \in \mathcal{F}} \min_{f \in \mathcal{F}} \left(\mathbb{E}[N_f(t)] + \sum_{i=1}^{t-1} \mathbb{E}[\Delta N_f(i)] \right). \quad (2)$$

In contrast to max-delivery, we give higher priority to the flows with the less average number of delivered packets. Again, since only $(\mathbb{E}[N_f(t)], f \in \mathcal{F})$ is available to resource allocation at slot t , we first choose a session f^* such that $f^* = f^*(t) = \arg \min_{f \in \mathcal{F}} \mathbb{E}[N_f(t)]$, and allocate resource to maximize $\Delta N_{f^*}(t)$.

(c) Min-delay. The structure of minimizing delay is similar to maximizing that of the delivery ratio. Similarly to $\Delta N_f(t)$, we define $\Delta D_f(t) \triangleq D_f(t-1) - D_f(t)$ to be a marginal decrease in delay of flow f over interval $[t-1, t]$. Note that this delay

decrease is possible by copying the packet in question to other nodes.

$$\begin{aligned} \min_{f \in \mathcal{F}} \sum \mathbb{E}[D_f(0)] &= \min_{f \in \mathcal{F}} \mathbb{E} \left[\sum_{f \in \mathcal{F}} \inf_s \{D_f(s) = 0\} \right] \\ &= \min_{f \in \mathcal{F}} \mathbb{E} \left[\sum_{f \in \mathcal{F}} \inf_s \{D_f(0) = \sum_{i=1}^s \Delta D_f(i)\} \right]. \quad (3) \end{aligned}$$

At slot t , the first step to approximate the above using the snapshot information, is to maximize $\Delta D_f(t)$. Recall that $\Delta D_f(t)$ is random in terms of random mobility. It means that the maximization of $\Delta D_f(t)$ is feasible (in the sample-path sense) only if the full information about mobility (even including future) were given to nodes, which is impossible due to limited knowledge of mobility in the future. Thus, an alternative approach is to take the expectation of $\Delta D_f(t)$, i.e., $\mathbb{E}[\Delta D_f(t)]$, which we maximize at the snapshot. Thus, our snapshot optimization problem is $\max_{f \in \mathcal{F}} \mathbb{E}[\Delta D_f(t)]$.

(d) Fair-delay. Similarly to fair-delivery, we have:

$$\begin{aligned} \min_{f \in \mathcal{F}} \max_{f \in \mathcal{F}} \mathbb{E}[D_f(0)] &= \min_{f \in \mathcal{F}} \max_{f \in \mathcal{F}} \mathbb{E} \left[\inf_s \{D_f(s) = 0\} \right] \\ &= \min_{f \in \mathcal{F}} \max_{f \in \mathcal{F}} \mathbb{E} \left[\inf_s \{D_f(0) = \sum_{i=1}^s \Delta D_f(i)\} \right]. \quad (4) \end{aligned}$$

However, the issues of approximating sample-paths with the expectation exist, which we handle similarly to min-delay. Thus, our snapshot objective is to maximize $\Delta D_{f^*}(t)$ where $f^* \triangleq f^*(t) = \arg \max_{f \in \mathcal{F}} \mathbb{E}[D_f(t)]$.

For space saving and ease of presentation, we henceforth focus mainly on objective **R1**. Discussions of other three objectives will be handled in the technical report [15].

III. SNAPSHOT OPTIMALITY: HARDNESS AND MAX-CONTRIBUTION

Towards practical, distributed algorithms, we take a multi-step systematic approach. First, we develop an algorithm, called OPT, that is provably snapshot optimal. We will show that OPT requires centralized, intractable computations and the global knowledge of network state. Next, we develop a centralized approximation heuristic to OPT, called, *Max-Contribution* (MC) which provides an insight to the development of a distributed approximation to OPT, called *Distributed Max-Contribution* (DMC) presented in Section IV.

A. Value and Contribution

We first introduce a notion of *value*. Associated with each packet-company m is a *value* v_m . A packet value quantifies a per-packet metric defined according to the target objective. For a given objective, the value of a packet-company at a time slot is time-varying over slots and depends on the mobility patterns of the nodes holding the copies of the packet at that slot. For max-delivery (**R1**), the value of a packet-company m is defined as the delivery probability of any packet in m to be delivered to its destination. Since all packets in the same company share the same value, we interchangeably use the value of a packet and the value of packet company that

the packet belongs to. As a measure of the improvement in the value incurred by packet forwarding and replication, we introduce the notion of *contribution* of a packet m , Δv_m to be the increased amount of v_m when m is forwarded and copied in the network. Note that when multiple packets in a packet-company are copied at the same time in the network, the contribution is the sum of all the contributions that each copy makes.

B. OPT: A Snapshot Optimal Algorithm

We now describe the generic algorithm, OPT, that is snapshot-optimal for the four objectives, when value v_m is suitably defined. The key idea of OPT is to make link/copy scheduling decisions (over slots) that maximize the expectation of the total increase in the packet values over the entire network.

OPT

At each slot t , copy packets according to (π^*, S^*) , which is the optimal solution of

$$\max_{\pi \in \Pi(t), S \in \mathcal{S}(t)} \sum_{m \in \mathcal{G}(\pi, S)} \Delta v_m(t), \quad (5)$$

where $\mathcal{G}(\pi, S)$ is the set of all packet-companies scheduled by a pair of copy and link schedule (π, S) .

Note that $\mathcal{G}(\pi, S)$ is a set. Thus, even in the case when the packets in the same company m are scheduled over different links, only the company index m is in $\mathcal{G}(\pi, S)$. As an example, we now explain that OPT with $v_m = p_m$ is snapshot-optimal for the max-delivery objective, **R1**, where p_m is the probability that at least one packet in the packet-company m is delivered to the destination. Recall that the snapshot objective for **R1** is to $\max_{\pi, S} \sum_f \mathbb{E}[\Delta N_f(t)]$.

Example 3.1 (R1. Max-Delivery): First, denote by $I_m(t)$ is an indicator random variable recording whether at least one packet in company m is delivered over $[t-1, t]$ or not. Let $\Delta p_m(t) = p_m(t-1) - p_m(t)$. Then, remarking that $\Delta N_f(t) = \sum_{m \in \mathcal{G}_f} I_m(t)$, we get

$$\begin{aligned} & \max_{\pi, S} \sum_f \mathbb{E}[\Delta N_f(t)] \\ &= \max_{\pi, S} \sum_f \mathbb{E} \left[\sum_{m \in \mathcal{G}_f} I_m(t) \right] = \max_{\pi, S} \sum_{m \in \cup_f \mathcal{G}_f} p_m(t) \\ &= \max_{\pi, S} \sum_{m \in \cup_f \mathcal{G}_f} \left(p_m(t) - p_m(t-1) + p_m(t-1) \right) \\ &= \max_{\pi, S} \left(\sum_{m \in \mathcal{G}(\pi, S)} \Delta p_m(t) + \sum_{m \in \cup_f \mathcal{G}_f \setminus \mathcal{G}(\pi, S)} \Delta p_m(t) \right) \\ &+ \sum_{m \in \cup_f \mathcal{G}_f} p_m(t-1) \quad (6) \end{aligned}$$

$$= \max_{\pi, S} \sum_{m \in \mathcal{G}(\pi, S)} \Delta p_m(t) + K_1(t) + K_2(t-1), \quad (7)$$

where in (6) we divide the packet-companies into ones that are scheduled and not by (π, S) . $K_1(t)$ and $K_2(t-1)$ correspond

to the second and third term in (6). For a fixed t , $K_1(t)$ is a constant as the packet-companies that are not scheduled do not depend on (π, S) . $K_2(t-1)$ is also a constant at time t . Finally, from $\Delta v_m(t) = \Delta p_m(t)$ by definition, the result follows.

The OPT algorithm is impractical for the following reasons:

- 1) *Coupling between copy and link scheduling.* v_m jointly depends on both copy and link schedules. For **R1**, when two different packets in the *same* packet-company m are scheduled over different links, the contribution of m should jointly consider the two copies because its delivery probability p_m is determined by any copy in m .
- 2) *Global knowledge of values.* All nodes holding any packet in a packet-company m need to have the same value v_m , which is hard to achieve in a distributed environment. A vanilla method is to flood the value change event, requiring heavy message passing, thereby wasting resources.
- 3) *Computational intractability.* The OPT algorithm requires the exhaustive search to find a solution in the large-scale search space. Formally, the problem can generally be formulated by an integer programming with an exponential size of search space. In fact, for a fixed π , the inner maximization of Eq. (5) over all feasible schedules is a variant of an NP-hard wireless scheduling problem (see [16] for details) that can be reduced to the NP-hard MWIS (Maximum Weight Independent Set) problem³.

C. Link/Copy Scheduling Decomposition: Max-Contribution

Complex coupling between copy and link scheduling happens when multiple copies of the same packet are scheduled over different links simultaneously. To develop a practical approximation algorithm, we propose Max-Contribution that decouples link and copy scheduling. In Max-Contribution, OPT is solved with the set $\Pi'(t)$ of copy schedules, where

$$\Pi'(t) = \{ \pi \in \Pi(t) \mid \pi_i \neq \pi_j, \forall i, j \}.$$

Since $\Pi'(t) \subset \Pi(t)$ for all t , it is clear that the contribution computed from OPT is no less than that from MC. We transform the original optimization problem into one over a *reduced* constraint set. Then, as we discussed, the optimal algorithm becomes much simpler, which we in turn use to develop practical, on-line, distributed algorithms later in Section IV.

From the use of $\Pi'(t)$ instead of $\Pi(t)$, the contributions do not depend on the entire schedule, but only on the corresponding link l (more precisely, its receiver node, $rx(l)$), because only node, say v , changes the contribution of a packet that it holds. This approximation enables us to decompose copy scheduling from link scheduling, and first solve the outer-maximization by, for each link l , selecting the packet-company m_l^* that has the maximum contribution. For clarity, we now use a notation Δv_m^l to refer to the contribution of a packet in packet company m when it is copied over link l . Note that $|\Pi'(t)|$ gets closer to $|\Pi(t)|$ as $|\cup_f \mathcal{G}_f|/|\mathcal{S}(t)|$ gets larger. Thus, MC is near-optimal when the offered load in the network is high compared to the number of schedulable links.

³Due to space limitations, we omit the proof.

Max-Contribution

At each slot t ,

Step 1. Contribution computation.

Each node computes the contributions of the packets (or copies) in its buffer over its connected links.

Step 2. Copy scheduling.

On each link $l \in \mathcal{S}(t)$, set the weight $W_l(t)$ of the link l to be $\max_m \Delta v_m^l(t)$, and let

$$m_l^* = \arg \max_m \Delta v_m^l(t)$$

Step 3. Link scheduling.

Select the schedule $S^*(t)$ that satisfies

$$S^*(t) = \arg \max_{S \in \mathcal{S}(t)} \sum_{l \in S} W_l(t), \quad (8)$$

Step 4. Packet copying.

Replicate the packet (or the copy) m_l^* over the link l , for all $l \in S^*(t)$.

Unfortunately, Max-Contribution is still expensive to implement even with decoupling between link and copy scheduling. The need to have global knowledge of v_m remains, and the link scheduling problem maximizing the sum weights of links is NP-hard, which, again, can be reduced to the MWIS problem⁴.

IV. DISTRIBUTED MAX-CONTRIBUTION AND EXTENSION

A. Distributed Max-Contribution (DMC)

Copy scheduling. The main difficulty of MC is its requirement that all nodes holding a copy of a packet company m have to have the same value of v_m . DMC approximates this process through a technique called *fusion* which is used to maintain the set of nodes that currently own a copy of a packet m . Each node i keeps track of a set of other nodes, $\mathcal{N}_{m,i}$, that have a copy of each packet m it currently holds. $\mathcal{N}_{m,i}$ is called a *node set* of i for m . Along with a node set for m , node i maintains the delivery probability of each member in the set. It is initially empty and adds another node j when node i forwards a copy of m to j . After the forwarding happens, node j sets $\mathcal{N}_{m,j} = \mathcal{N}_{m,i}$. When node i meets a node k with the same copy m , then nodes i and k synchronize their node sets for m by taking union of $\mathcal{N}_{m,i}$ and $\mathcal{N}_{m,k}$. Whenever $\mathcal{N}_{m,i}$ is updated either by forwarding the copy or by meeting another node with the same copy, node i recomputes v_m . If the global performance objective is **R1**, v_m is equal to the probability, p_m that any node holding any copy of m meets the destination of m and delivers m . v_m is recomputed in the following manner. Denote the value of packet m at node i by $v_{m,i}$ and the delivery probability (i.e., meeting probability) of i with the destination of m by $q_{m,i}$. Then

$$v_{m,i}(t) = p_{m,i}(t) = 1 - \prod_{k \in \mathcal{N}_{m,i}} (1 - q_{m,k}(t)). \quad (9)$$

⁴Under one-hop interference model, the link scheduling problem is reduced to Maximum Weight Matching (MWM) whose complexity is $O(L^3)$.

To make a copy schedule at time t , DMC performs the following operations. When a node i with a packet m meets other nodes, they first exchange the IDs of packets whose copy they currently hold and then perform fusion by synchronizing their node sets and corresponding value information (i.e., delivery probabilities) and recomputing packet values. After this process, a node performs copy scheduling. For each packet m , node i computes the marginal increase of packet value of m when i is copied to each neighbor j . If j is already holding m , then the marginal increase is zero. If it is not, then the marginal increase is the difference between the current value of m and the new value of m if m is copied to j (i.e., recomputed value after adding j to $\mathcal{N}_{m,i}$). Node i picks the packet with the biggest marginal value increase for scheduling. Denote such a packet by $m_{i,j}^*$ where m is scheduled for copy for a link between nodes i and j . We call $m_{i,j}^*(t)$ the *candidate copy* of node i at time t .

Link scheduling. The scheduling algorithm that solves Eq. (8), referred to as Max-Weight scheduling, has been extensively studied to provide provable throughput guarantee. Recent efforts on distributed scheduling can provide us an array of candidate, low-cost algorithm to Max-Weight. Examples include greedy, locally-greedy, random pick-and-compare (see [16] and the references therein for details). Such algorithms provide (partial) throughput performance guarantee, where throughput is defined by the achieved stability region. We can also adopt one of them in our framework as a distributed heuristic. For our simulation, we use a locally greedy algorithm which schedules, at each time t , the transmission of a packet whose marginal value increase is biggest among all candidate copies of nodes that are in an interference region at time t .

B. Extension

Exploiting physical broadcast. We have so far considered only unicast transmissions. Physical transmission in wireless networks is broadcast. We can improve the performance of DMC by exploiting overhearing through broadcast. When a node i transmits a copy m to node j , then another neighboring node k can overhear m . Then we allow node k to carry the packet and performs DMC with that. In this case, node i does not know whether node k has received that packet or not (as no acknowledgement is sent). Thus, node i does not update its packet value for the reception of m by k . However, this has a tendency of improving the performance for a given objective.

Cost and efficiency: Tradeoff. In networks, the number of packets is an important concern especially when transmitting a packet can be costly in terms of energy consumption and storage. In such cases, DMC can be adapted to keep the the number of copies in the network in check. One way to accomplish it is to set a threshold T such that a node does not schedule a packet to a node whose delivery probability is less than T (meaning that the node is not qualified for efficient delivery).

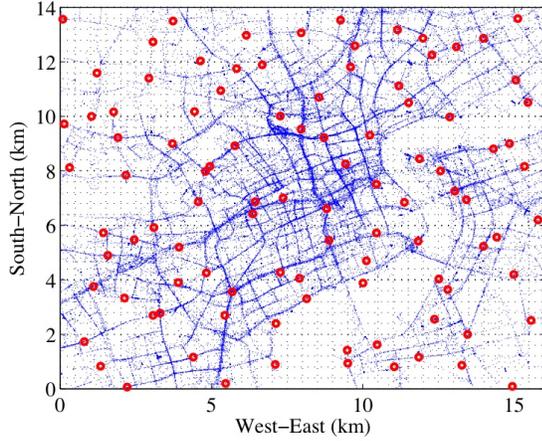


Fig. 1. Snapshot distribution of taxis in Shanghai. Each dot indicates the location of a taxi during one hour at 11/28/2006. Circles indicate the candidate locations of sources and destinations.

Delegation Forwarding (DF) [17] is known to efficiently save cost while maintaining reasonable delivery ratio (note that scheduling is not considered in DF). DMC is versatile enough to approximate DF. We can set the threshold value of DMC that equalizes the asymptotic number of copies in DMC with that in DF as follows.

$$T = \theta(p_{m,i}) = 1 - a(1 - p_{m,i})^C, \quad (10)$$

where

$$C = \frac{\log(2/a) - \sqrt{(\log(2/a))^2 - 4 \log(1 - p_{m,i}) \log 2}}{2 \log(1 - p_{m,i})},$$

and a is any positive constant satisfying $C \geq 0$.

Our simulation comparing the costs of DMC-with-threshold to DF shows that the cost is indeed the same although DMC shows better delivery ratio. The formal derivation of the threshold function is presented in Appendix.

V. PERFORMANCE EVALUATION

A. Node Delivery Probability from Shanghai Trace

For performance evaluation, we use GPS traces of over 4000 Shanghai taxis [9], by far, the largest vehicular GPS traces publicly available. The location information of each taxi is recorded at every 40 seconds within an area of $102km^2$ for 28 days (4 weeks). We consider a DTN application where many infostations are randomly scattered around the city in a uniform manner and using a mobile network of taxis equipped with WiFi, data from one infostation (i.e., source) is moved to another infostation (i.e., destination). The infostations do not have an access to infrastructure and they simply upload data in units of packets to passing-by taxis. These infostations are like public bulletin boards or street advertisement boards. Daily updated content from one location is delivered to a set of destination infostations for display. In this paper, we consider only unicast scenarios and defer multicast scenarios to future work.

People do not move randomly. Any mobile networks whose constituent members are humans or vehicles driven by them

cannot be described as random movement and there exists some regularity or periodicity in their mobility [18], [19]. From the taxi traces, we also find some regularity (1) in the patterns of locations each taxi visits daily and (2) in the patterns of meetings among taxis. Further, we find that taxis exhibit some biases in choosing locations they visit and thus other taxis they meet and that different taxis tend to have different biases. These regularities are essential in extracting information required to run DMC. To illustrate this, we plot the CCDF (complementary CDF) of the inter-contact times (ICT) and inter-visit times (IVT) of taxis in the traces, in Figure 2. The distributions are best fitted with exponential distributions. This is quite different from the human mobility pattern which shows truncated power-law inter-contact time distributions. Figure 3 plots the individual intensity values (λ^{IVT} and λ^{ICT}) of IVT and ICT exponential distributions of randomly chosen 100 taxis. For better readability of the graph, we plotted 100 taxis instead of whole taxis. IVT is plotted for 100 destination locations. From the plots, we find that different taxis show different biases in the locations they visit and in the set of taxis they meet daily.

These characteristics of the Shanghai taxi network allow us to extract scheduling metadata. In particular, from the exponential distributions we fitted to each individual taxi's IVT and ICT, we can derive the *node delivery probability*, $q_{m,i}$ of a node i to the destination location, $d(m)$ of a packet m which implies the maximum potential delivery probability. More precisely,

$$q_{m,i}(t) = \max\{q_{m,i}^1(t), q_{m,i}^2(t), q_{m,i}^3(t), \dots\}, \quad (11)$$

where $q_{m,i}^k$ denotes the delivery probability through k hops. For example, 1-hop probability, $q_{m,i}^1(t)$ is the probability that node i directly meets the destination location, $d(m)$ during the interval $[t, t_{dl}]$. For 2-hops or more, we find the path (sequence of nodes) with the maximum delivery probability by comparing all combinations of the intermediate nodes. Thus, the k -hop delivery probability is defined as follows (note that n_k denotes the k -th hop node and we replace $n_1 = i, n_{k+1} = d(m)$ for the ease of expression).

$$q_{m,i}^k(t) = \max_{\{n_2, \dots, n_k\} \in \mathcal{N}^{k-1}} \left\{ \int_{t_{n_{k-1}, n_k}}^{t_d - t} \dots \int_0^{t_d - t} \mathbb{P}[T_{n_1, n_2} = t_{n_1, n_2}] \prod_{j=2}^{k-1} \mathbb{P}[T_{n_j, n_{j+1}} = t_{n_j, n_{j+1}} - t_{n_{j-1}, n_j}] \mathbb{P}[T_{n_k, n_{k+1}} \leq (t_d - t) - t_{n_{k-1}, n_k}] dt_{n_1, n_2} \dots dt_{n_{k-1}, n_k} \right\} \quad (12)$$

where \mathcal{N}^k and $T_{n_j, n_{j+1}}$ denote the k -combinations of node sequences from the node set \mathcal{N} excluding the node i itself and a random variable indicating the inter-contact time or the inter-visit time between the j -th node and the $(j+1)$ -th node (or location).

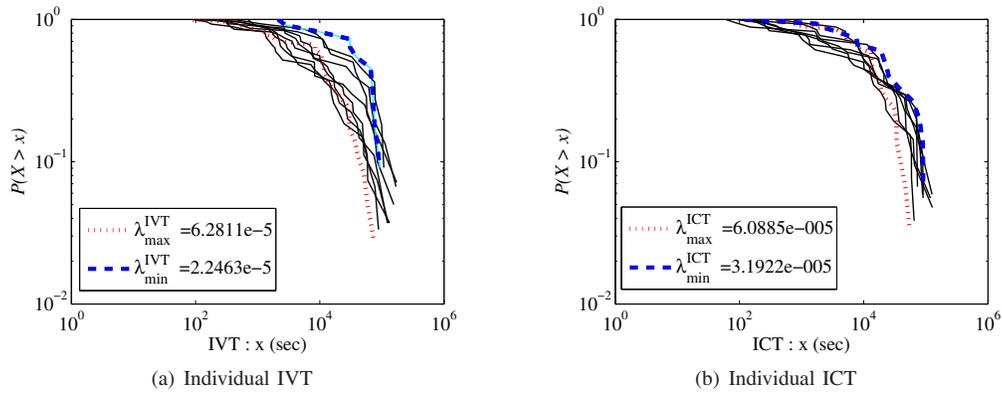


Fig. 2. Inter-visit time (IVT) distribution and inter-contact time (ICT) distribution of a taxi to locations and other taxis. They are fitted by maximum likelihood estimation (MLE) to exponential distributions. The maximum and minimum intensity of the best fitting exponential distributions are λ_{min} and λ_{max} , respectively.

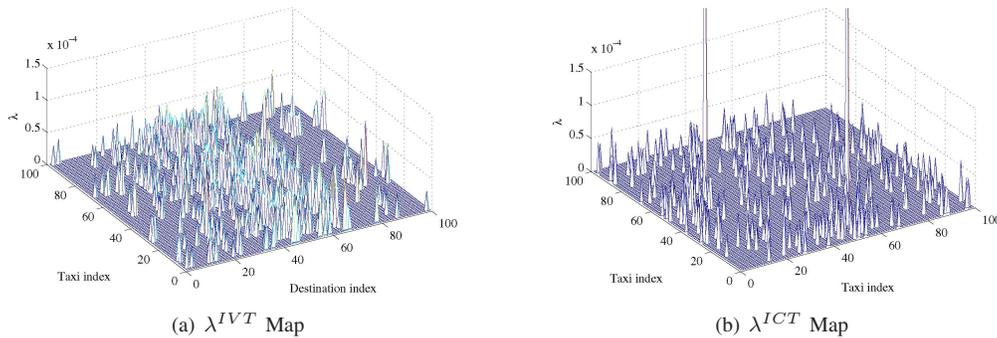


Fig. 3. We plot the individual intensity values (λ^{IVT} and λ^{ICT}) of IVT and ICT exponential distributions of randomly chosen 100 taxis for better readability. IVT is plotted for the 100 destination locations. A high intensity value of a particular location by a particular taxi implies that a taxi has a high rate of visit to a particular location. Likewise, a high intensity value of a taxi with respect to another taxi meets they tend to meet very often. From the plots, we find that different taxis show different biases in the locations they visit and in the set of taxis they meet daily.

B. Setup, Metric and Tested Algorithms

We implemented a resource allocation simulator for a DTN using MATLAB. Among over 4000 taxis, we selected relatively reliable 1486 taxis that show less than 30% of unclear GPS coordinates in a day (included in the traces) before performing interpolation. By default, we use the communication range of WiFi, 300 meters. Also, we selected 100 candidate locations (uniformly distributed) and 32 random pairs of S-D (source, destination) in the 100 candidate locations for our simulation. We also vary the number of packets per S-D pair to see the performance for different traffic loads. We set the deadline (i.e., t_{dl}) to be 24 hours. We make resource allocation decisions every 30 seconds. We also tested other intervals, and observed similar trends. We repeated ten simulations; each time, we vary S-D pairs randomly with different seeds.

We present the results for the max-delivery objective. Two performance metrics are considered: (i) *delivery ratio* and (ii) *efficiency*. Delivery ratio is the ratio of the total delivered packets (counting only original packets) within a designated time deadline to the total number of packets that sources initially have. Efficiency is the delivery ratio per unit cost where cost is simply the total number of transmissions by transmitters.

We evaluate seven algorithms summarized in Table I. *MC-*

TABLE I
 TESTED ALGORITHMS (* CORRESPONDS TO THE ITEMS THAT WE ADDED FOR FAIR COMPARISON)

| Algorithm | Link scheduling | Copy scheduling | Value update |
|---------------|-----------------|-----------------------------|--------------|
| Random | random | random | × |
| DF [17] | *greedy | *difference. | delegation |
| RAPID [14] | *random | contribution | global |
| MC-Global | greedy | contribution | global |
| DMC | greedy | contribution | fusion |
| DMC-Threshold | greedy | contribution with threshold | fusion |
| DMC-Broadcast | greedy | contribution with broadcast | fusion |

Global uses the global view of packet values, but solves link scheduling using local greedy link scheduling of DMC. This is because solving the MWIS problem for link scheduling at the scale of our network is too time consuming. Some protocols do not have in their design the specifications for link/copy scheduling and value updates. Thus, for fair comparison, we additionally implemented the absent features. For example, link scheduling has not been considered in DF and RAPID in their papers. In random scheduling and forwarding, links and packets are randomly selected out of the connected links and packets that exist in either of two nodes that meets. In DF, link scheduling requires prioritizing the packets to copy, for

which we apply the differences of packet delivery probabilities (that are originally used in DF for reducing cost based on thresholds). We used “delegation” originally proposed in DF for value updates, i.e., when a packet m is copied from v to w , the delivery probability of w for m is also copied to v . We intentionally use random (e.g., CSMA) for link scheduling at RAPID to quantify the impact of the joint copy and link scheduling. DMC-threshold and DMC-Broadcast use the features of thresholding and broadcast described in Section III.

C. Simulation Results

Fig. 4 shows the delivery ratio and efficiency of scheduling algorithms against the offered load (the number of packets). The delivery ratio decreases as the offered load (the number of input packets) increases. MC-Global, DMC and DMC-Broadcast show better deliver ratios than any other protocols. DMC shows almost as good delivery ratio as MC-Global. This indicates that the localized information update, *Fusion*, can efficiently replace the expensive global knowledge update used in MC and also in RAPID. The main performance difference between DMC and RAPID is about 10% to 15% under high load and arises from use of more intelligent link scheduling for DMC. We believe this effect will be more resounding when the network density increases. DMC-Broadcast shows the best delivery ratio in all offered loads. Under 1500 packets, the offered load is much higher than the capacity of the network leaving many packets to miss the deadline. All protocols suffer their performance. However, DMC-Broadcast still outperforms by about 20-30% over DMC and 45% over RAPID. Clearly opportunistic copying using broadcast improves the performance substantially. DMC-Threshold always does better than DF in efficiency which is known to achieve good balance of the delivery ratio and the cost. We confirm that the cost of DMC-Threshold and DF is very similar, which is why DMC-Threshold shows better efficiency. Among all tested algorithms, Random shows the worst performance in all cases. It was expected as it does not exploit the characteristics of IVT and ICT as shown in Figs 2 and 3.

Figs 4 (a) and (b) are obtained when the radio range is 300 meter. We now modify the radio range from 300 meters to 500 meters to test the performance of various protocols under high density environments. We are interested in studying the effect of more intelligent (but practical) link scheduling on the performance. Fig. 4 (c) compares the delivery ratios of DMC and RAPID as these two protocols are essentially different in two points: (1) DMC uses more lightweight metadata dissemination called *Fusion* than RAPID which uses flooding and (2) DMC uses greedy link scheduling while RAPID uses random link scheduling. In general, the effect of the first point is minimal because the performance of DMC and MC-Global is not much different. Therefore, the performance difference between them most likely comes from the effect of the second point. As the density of the network increases, the interference becomes larger. Thus, we can observe from the figure that the performance gap between the two protocols get bigger by about two times. The performance of DMC improves with the

increased radio range because of higher chance of meeting other nodes.

VI. RELATED WORKS

A common assumption in DTN research is that nodes are sparsely distributed and packet delivery is instantaneous. This assumption greatly simplifies the problem. Many popular DTN routing algorithms are heuristically developed under this assumption (e.g., Epidemic routing [2], Prioritized epidemic routing [3], DREAM [4], PRoPHET [5], Knowledge-based forwarding [6], Last encounter-Time based forwarding [7], Spray and wait [8]). Under the same assumption, there have been some theoretical work [12], [13] for maximizing the delivery ratio as well as minimizing energy consumption by controlling the number of copies. Their solutions use dynamic programming. However, dynamic programming has limitations in producing an on-line solution because of its complexity and also the requirement to know the future events.

Some recent work [14], [20] considers the case where link bandwidth could be limited at the time of contacts and apply some concept of packet scheduling (i.e., copy schedules). They model the DTN as a resource allocation problem and provide a heuristic packet scheduling based on per-packet marginal utility where utility is defined as the average delay or the ratio of packets which missed a deadline. Along the similar line of research, [21] adds an optimal drop policy for a limited buffer which drops packets in consideration of the per-packet marginal utility. By far, no other work has considered the joint constraints of interference and link and copy schedules.

VII. CONCLUSION

The main contributions of this paper are three-folds. First, we consider resource allocation for jointly optimizing link scheduling, routing and replication. This framework allows the developed solutions to be adaptive to various conditions of networks whether they are dense with high interference or sparse with high rates of disconnections. Second, optimal resource allocation for jointly optimizing link schedule and replication-based routing is a hard problem in ST-DTNs because of dynamic links and various control knobs of improvement for forwarding and replications. Many existing techniques try to focus on one or two knobs for improved performance by applying intuition-driven heuristics. In this paper, we systematically approach the problem; we theoretically solve the optimal solution for snapshot optimality which restricts nodes to use only contemporarily available knowledge, and then approximate various components of the optimal solution to reduce its complexity without much loss in the performance. Our theory-driven approach clearly shows how we derive our heuristic solutions and provides some confidence over the expected performance. Finally, we demonstrate how our solutions can be applied to solving real world problems, such as information dissemination over a network of over 1000 taxis, each equipped with a WiFi radio, which is by far the biggest DTN network being simulated using real traces. From the traces, we extract statistical properties of taxi movements

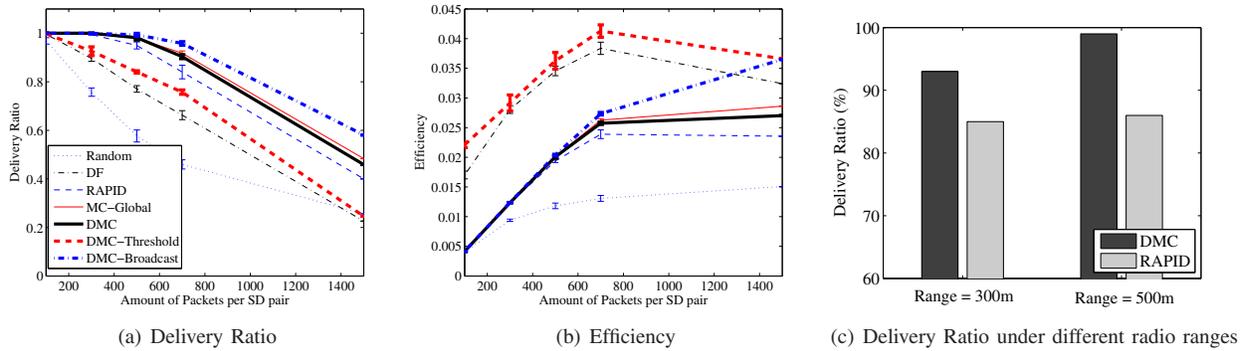


Fig. 4. (a) and (b) The Delivery ratio and Efficiency of algorithms listed in Table I against the offered load to 32 S-D pairs. Each value shows 95% confidence interval. We do not show cost as it is implied in the efficiency. (c) The delivery ratio of RAPID and DMC under different radio ranges. The input load is set to 700 packets.

and apply them to formulate parameter values to the input of our algorithms. This work clearly demonstrates how our solutions would perform in real network settings.

APPENDIX

A. Derivation of Eq. (10)

We borrow the technique from [17]. The main interest is to study how fast the threshold increases, resulting in the asymptotic number of copies in the network. Assuming that the delivery probability of nodes is uniform over $[0, 1]$ for simplicity, whenever a node v (holding the packet m) meets another node whose delivery probability is above $G(t) \triangleq 1 - \theta(p_{m,v}(t))$, the node v copies packet m , and then fuse its probability to update $p_{m,v}(t+1)$. The quantity that we are interested in is $\mathbb{E}[G(t+1)|G(t)]$ that states the increasing speed of the threshold.

$$\begin{aligned} \mathbb{E}[G(t+1)|G(t)] &= 1 - \mathbb{E}[\theta(p_{m,v}(t+1)) | p_{m,v}(t)] \\ &= 1 - \theta\left(\frac{1 + \min(p_{m,v}(t+1))}{2}\right). \end{aligned} \quad (13)$$

Then, in DF it sets the threshold $\theta(p_{m,v}(t)) = p_{m,v}(t)$ as shown in [17] and gives,

$$\mathbb{E}[G(t+1) | G(t)] = (1 - p_{m,v}(t))/2. \quad (14)$$

Whereas, in DMC, as the local value update follows $p_{m,v}(t+1) = 1 - (1 - p_{m,v}(t))(1 - q_{m,w}(t))$ when v copies m to w with the threshold $\theta(p_{m,v}(t))$, by applying this to Eq. (13), it is easy to show that:

$$\mathbb{E}[G(t+1)|G(t)] = 1 - \theta\left(\frac{2 - (1 - p_{m,v}(t))(1 - \theta(p_{m,v}(t)))}{2}\right). \quad (15)$$

Then, it suffices to find a function $\theta(\cdot)$, so that Eq. (15) is equal to Eq. (14) of DF. We can easily check that $\theta(\cdot)$ in Eq. (10) can satisfy it.

REFERENCES

- [1] S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in networks with predictable mobility," in *Technical report: GIT-CC-04-07. College of Computing, Georgia Institute of Technology*, 2004.
- [2] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," in *Technical Report, CS-200006, Duke University*, 2000.
- [3] R. Ramanathan, R. Hansen, P. Basu, R. R. Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of ACM MobiSys workshop on Mobile Opportunistic Networks (MobiOpp)*, 2007.
- [4] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (dream)," in *Proceedings of ACM MobiCom*, 1998.
- [5] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.
- [6] J. Lebrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of IEEE Vehicular Technology Conference*, 2004.
- [7] 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.
- [8] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proceedings of ACM SIGCOMM*, 2005.
- [9] S. J. U. Traffic Information Grid Team, Grid Computing Center, "Shanghai taxi trace data," <http://wirelesslab.sjtu.edu.cn/>.
- [10] B. City, "Bucheon traffic information center," <http://www.bcits.go.kr/>.
- [11] J. Leguay, T. Friedman, and V. Conan, "Dtn routing in a mobility pattern space," in *Proceedings of ACM Chants Workshop*, 2005.
- [12] E. Altman, G. Neglia, F. D. Pellegrini, and D. Miorandi, "Decentralized stochastic control of delay tolerant networks," in *Proceedings of IEEE INFOCOM*, 2009.
- [13] C. Liu and J. Wu, "An optimal probabilistic forwarding protocol in delay tolerant networks," in *Proceedings of ACM MobiHoc*, 2009.
- [14] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proceedings of ACM SIGCOMM*, 2007.
- [15] K. Lee, Y. Yi, I. Rhee, J. Jeong, H. Won, and S. Chong, "Max-contribution: On optimal resource allocation in delay tolerant networks," in *Technical report, KAIST*, 2009, available at <http://lanada.kaist.ac.kr/pubs/dmc.pdf>.
- [16] Y. Yi, A. Proutiere, and M. Chiang, "Complexity in wireless scheduling: Impact and tradeoffs," in *Proceedings of ACM MobiHoc*, 2008.
- [17] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proceedings of ACM MobiHoc*, 2008.
- [18] M. Kim and D. Kotz, "Periodic properties of user mobility and access-point popularity," *Journal of Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 465–479, August 2007.
- [19] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong, "Slaw: A new human mobility model," in *Proceedings of IEEE INFOCOM*, 2009.
- [20] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networking," in *Proceedings of IEEE INFOCOM*, 2006.
- [21] A. Krifa, C. Barakat, and T. Spyropoulos, "An optimal joint scheduling and drop policy for delay tolerant networks," in *Proceedings of the WoWMoM Workshop on Autonomic and Opportunistic Communications*, 2008.