Max Contribution: An On-line Approximation of Optimal Resource Allocation in Delay Tolerant Networks

Kyunghan Lee, *Member, IEEE,* Jaeseong Jeong *Member, IEEE,* Yung Yi, *Member, IEEE,*
Hyungsuk Won, *Member, IEEE,* Injong Rhee, *Member, IEEE,* and Song Chong, *Member, IEEE*

Abstract—In this paper, a joint optimization of link scheduling, routing and replication for delay-tolerant networks (DTNs) has been studied. 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 approximation to the optimality for DTNs, called snapshot approximation where nodes are not clairvoyant, i.e., not looking ahead into future events, and thus decisions are made using only contemporarily available knowledges. Unfortunately, the snapshot approximation still requires solving an NP-hard problem of maximum weighted independent set (MWIS) and a global knowledge of who currently owns a copy and what their delivery probabilities are. This paper proposes an algorithm, Max-Contribution (MC) that approximates MWIS problem with a greedy method and its distributed on-line approximation algorithm, Distributed Max-Contribution (DMC) that performs scheduling, routing and replication based only on locally and contemporarily available information. Through extensive simulations based on real GPS traces tracking over 4000 taxies and 500 taxies for about 30 days and 25 days in two different large cities, DMC is verified to perform closely to MC and outperform existing heuristically engineered resource allocation algorithms for DTNs.

Index Terms—resource allocation, routing, scheduling, optimality, delay tolerant network, mobile ad hoc network.

1 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. While DTNs are traditionally regardes as sparse, disconnected networks where mobility and carry-and-forward paradigm has been adopted as the only means of communication, an extended view of DTNs shown in [1] permits traditional senses of MANETs and DTNs to coexist in the form of disconnected islands in any proportion of time and space.

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 replications in

This work was supported in part by the National Science Foundation under Grants CNS-0910868 and CNS-1016216 and the Future Strategic Fund(1.14003.01) of UNIST(Ulsan National Institute of Science and Technology) and Basic Science Research Program and International Research&Development Program through National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT&Future Planning (NRF-2011-0015042, NRF-2013R1A2A2A01067633, 2013K1A3A7A03079321) and the Center for Integrated Smart Sensors funded by the MSIP as Global Frontier Project (CISS-2012M3A6A6054195).

K. Lee is with the School of Electrical and Computer Engineering, UNIST, Ulsan, Korea (e-mail: khlee@unist.ac.kr).

J. Jeong, corresponding author, was with the Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea. He is now with the Automatic Control Department, KTH, Stockholm, Sweden (e-mail: jsjeong.dr@gmail.com).

Y. Yi, S. Chong are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, 305-701 Korea (e-mail: yiyung@kaist.edu,songchong@kaist.edu).

H. Won and I. Rhee are with the Department of Computer Science, North Carolina State University, Raleigh, NC, 27695 USA (e-mail: hwon@ncsu.edu, rhee@ncsu.edu).

A preliminary version of this paper was published in INFOCOM 2010.

sparse networks [2]–[8] whereas traditional MANET studies for resource allocation have focused on interference, link scheduling and routing in dense networks with no provisioning for disconnected islands. 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 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., taxies and buses, in a large city. The information dissemination networks of taxies 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 DTNs. Furthermore, 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 efficient on-line algorithms make such solutions impractical.

In this paper, we present an approximation technique, called *snapshot approximation*, in DTNs and its optimal solution, *OPT* which performs closely to the "clairvoyant optimal solutions" but rely only on contemporarily available knowledge of the networks. *OPT* dictates to maximize the "contribution" of a packet being scheduled at the current instance to improve the global utility. This notion was considered in the name of *per-packet marginal utility* in DTNs [14], but not in the context of joint consideration of packet and link scheduling. However, even maximizing contribution alone requires *(i)* global knowledge of how the links are formed and who owns a copy of a packet (as multiple copies are permitted) and *(ii)* their delivery statistics, and also solving an NP-hard problem of maximum weighted independent set for link scheduling. Therefore, a low-complexity approximating solution that leverages only contemporary and local information are vital. To construct a theoretically engineered, highly practical solution for snapshot approximation, we first apply a greedy resource allocation that performs link scheduling, routing and replication decision based on global information and propose an algorithm *Max-Contribution* (MC). MC reduces complexity for scheduling but still has major difficulty in obtaining global knowledge on the list of nodes holding the same packet and the overall link statistics. We propose an on-line approximation algorithm of MC, so called *Distributed Max-Contribution* (DMC) that replaces the global information with iterative computations of contemporarily available information.

Our simulation studies are based on two detailed GPS (Global Positioning System) traces of tracking the movements of taxies, each equipped with GPS in different cities. A set of traces is with over 4000 taxies moving around Shanghai, China [9] and another set of traces is with over 500 taxies in San Francisco area [34]. 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 approximation.

2 Related Works

In DTNs, when a source node has a packet to send to a destination node, generally there are several options for delivery which is often referred to as DTN routing. The first option is holding the packet until the source node meets the destination node. The second option is repeatedly forwarding the packet to a node (i.e., relay node) which is more likely to deliver the packet better than the previous holder until the current packet holder meets the destination node. The third option is repeatedly replicating the packet to a node (i.e., relay node) whenever the node is determined to be helpful in the delivery and all the nodes holding the packet carry it until one of them meets the destination node. Obviously third option has better chance to deliver a packet to a destination but makes the network congested. Most of the DTN research papers have focused on designing an efficient routing algorithm that makes a good balance between the number of copies in the network and the delivery performance where the efficiency comes from the judicious selection of relay nodes and the appropriate number of copies of a packet in a network.

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 (e.g., Epidemic routing [2], Prioritized epidemic routing [3], DataMule [15], DREAM [4], PRoPHET [5], Knowledge-based forwarding [6], Last encounter-time based forwarding [7], Spray and wait [8]) are heuristically developed and engineered under this assumption. Among many protocols, Delegation forwarding (DF) [16] in which a node is allowed to copy a packet only to a node whose delivery probability is higher than the maximum delivery probability that the packet has ever seen since it is first copied from a source node, is demonstrated to have a good balance between the performance and the cost.

Under the assumption, there have been a few theoretical work [12], [13] for maximizing the delivery ratio as well as minimizing transmission cost by controlling the number of copies. Their solutions use dynamic programming and provide simple threshold policies. Despite dynamic programming is useful in finding the best strategy in average sense, if the solution is derived from imperfect knowledge on future events (e.g., meeting process, time varying list of neighbor nodes and the list of packets queued in neighbor nodes), its performance cannot be guaranteed in practice.

Recently, [14], [17] started to consider more practical scenarios where link bandwidth could be not enough to transmit all the packets during a contact period. They modelled routing in a DTN as a resource allocation problem and provided a heuristic packet scheduling algorithm which decides an order to transmit the packets. In specific, RAPID [14] explicitly pointed out that there can be a situation where all packets queued in a node cannot be transmitted to a node in contact during the contact period and suggested a packet ordering metric relying on per-packet marginal utility (i.e., estimated increment in utility after a transmission) where utility is defined as average packet delivery delay or packet delivery ratio. In RAPID, packets with the highest marginal utility are sequentially transmitted until the contact is disconnected by nodes' mobilities. On a similar framework, [18] added an optimal drop policy for a limited buffer which drops packets in consideration of the per-packet marginal utility. However far, only few work [19] has jointly considered link and copy scheduling under limited transmission opportunities in DTNs.

We can also broadly classify DTNs into two major types: mobility-controllable and uncontrollable DTNs. Most of the aforementioned work are in the category of mobility-uncontrollable DTNs. Human-carried networks, also referred to as pocket-switched networks [20], [21], are the most popular mobility-uncontrollable DTNs. They focus mainly on the study of social relationship [22], [23] and context such as spatio-temporal regularity in inter-contact patterns which directly influence the performance of DTNs. In the vehicle-carried networks [24]–[27], another mobility-uncontrollable networks, mobility patterns of taxies/buses and their simulators including vehicular traffic models are studied. In controllable DTNs, often referred to as message-ferry networks, [28]–[33] mainly worked on designing movement paths of controllable mobile objects (i.e., ferries) to maximize the delivery performance of a network or to save the total energy consumed in a network. A controllable but stationary node (i.e., throwbox) was also introduced in [34] to extend the capacity of DTNs.

In this paper, our focus is on mobility-uncontrollable DTNs.

3 Optimal Resource Allocation

3.1 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 meets[1]. There is a set $\mathcal{F}$ of F sessions (flows) that are identified by a pair source-destination nodes. Associated with each session f, 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 $t = 0, 1, \ldots$. 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*, $S = (S_l \in \{0,1\} : l = 1, \ldots, L)$ 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 interference[2]). 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.

[1] We also use the word 'packet' to refer to the copies of the original packet, unless explicitly specified otherwise.

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

3.2 Objectives and Challenges

General 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). Similarly, 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.

R1. Max-Delivery	$\max \sum_{f\in\mathcal{F}} \mathbb{E}[N_f(0)]$
R2. Fair-Delivery	$\max \min_{f\in\mathcal{F}} \mathbb{E}[N_f(0)]$
D1. Min-Delay	$\min \sum_{f\in\mathcal{F}} \mathbb{E}[D_f(0)]$
D2. Fair-Delay	$\min \max_{f\in\mathcal{F}} \mathbb{E}[D_f(0)]$.

Optimization problem. Based on the above objectives R1, R2, D1, D2, the optimal sequence of copy and link schedules over time, $\{(\pi_t^*, S_t^*)\}_{t=0}^{t_{dl}}$ should be found under the constraint that $\pi_t^* \in \Pi(t), S_t^* \in \mathcal{S}(t), \forall t$, where $\Pi(t)$ and $\mathcal{S}(t)$ are the feasible sets of copy and link schedules at time t, respectively. Since the solutions are broken down into a sequence of scheduling steps over time, it can be formulated by a dynamic programming (DP).

Hardness of full optimality. The above DP problem requires a large dimensional search (i.e., curse of dimensionality) and knowledge of the future (i.e., in order to decide (π_0^*, S_0^*), we have to know $\Pi(t)$ and $\mathcal{S}(t)$ for $t > 0$). Due to such requirements, solving this problem via practical, on-line, decentralized algorithms is non-trivial. There are studies that use DP to develop optimal solutions. However, those 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 proposing a practical on-line algorithm. To that end, rather than pursuing the "full"-optimality based on DP, we adopt a *temporal greedy algorithm* 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 approximation* to the original problem. It is possible simply by temporally stretching the original optimization problems over the entire time slots, and investigating what needs to be optimized just using the information available at time t. Throughout this paper, we use the notion of '*snapshot approximation*' to represent temporally greedy decomposition of the original optimization problems.

Snapshot objectives. We now elaborate the snapshot approximated problems for various objectives introduced in the subsection 3.1.

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

$$\max \sum_{f\in\mathcal{F}} \mathbb{E}[N_f(0)] = \max \mathbb{E}\Big[\sum_{f\in\mathcal{F}} \Big(N_f(t) + \sum_{i=1}^{t} \Delta N_f(i)\Big)\Big] \quad (1)$$

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 $[t-1, t]$. Note 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 $\mathbb{E}[\sum_f \Delta N_f(t)]$ 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

$$\max \min_{f\in\mathcal{F}} \mathbb{E}[N_f(0)] = \max \min_{f\in\mathcal{F}} \Big(\mathbb{E}[N_f(t)] + \sum_{i=1}^{t} \mathbb{E}[\Delta N_f(i)]\Big) \quad (2)$$

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 $f^\star$ such that $f^\star = f^\star(t) = \arg\min_{f\in\mathcal{F}} \mathbb{E}[N_f(t)]$, and allocate resource to maximize $\Delta N_{f^\star}(t)$.

(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.

$$\min \sum_{f\in\mathcal{F}} \mathbb{E}[D_f(0)] = \min \mathbb{E}\Big[\sum_{f\in\mathcal{F}} \inf_s \{D_f(s) = 0\}\Big]$$
$$= \min \mathbb{E}\Big[\sum_{f\in\mathcal{F}} \inf_s \{D_f(0) = \sum_{i=1}^{s} \Delta D_f(i)\}\Big]. \quad (3)$$

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 $\max \sum_{f\in\mathcal{F}} \mathbb{E}[\Delta D_f(t)]$.

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

$$\min \max_{f\in F} \mathbb{E}[D_f(0)] = \min \max_{f\in\mathcal{F}} \mathbb{E}\Big[\inf_s \{D_f(s) = 0\}\Big]$$
$$= \min \max_{f\in\mathcal{F}} \mathbb{E}\Big[\inf_s \{D_f(0) = \sum_{i=1}^{s} \Delta D_f(i)\}\Big]. \quad (4)$$

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 $\Delta D_{f^\star}(t)$ where $f^\star \triangleq f^\star(t) = \arg\max_{f\in\mathcal{F}} \mathbb{E}[D_f(t)]$.

4 Snapshot Approximation

Towards practical, distributed algorithms, we take a multi-step systematic approach. First, we develop an algorithm, called OPT, that is provably temporal-greedy 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 5.2.

4.1 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 ($v_m = p_m$). For fair-delivery (**R2**), the value of an m is basically defined as 0 except the value of the packet-company associated with the flow of the lowest expected delivery ratio. The value of that packet-company is defined as the delivery probability. On the other hand, for min-delay (**D1**), the value of an m is defined as the expected delivery delay. Similarly to fair-delivery, for fair-delay (**D2**), the value of an m is basically defined as 0 except the value of the packet-company associated with the flow of the longest expected delivery delay. The value of that packet-company is defined as the expected delivery delay.

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. For all objectives, 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.

4.2 OPT: Solving Snapshot Approximation

We now describe the generic algorithm, OPT, that is optimal for the four snapshot 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 $(\pi^\star, S^\star)$, which is the optimal solution of

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

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 optimal for the snapshot 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 4.1 (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 $\Delta p_m(t) = p_m(t) - p_m(t-1)$. Then, remarking that $\Delta N_f(t) = \sum_{m\in\mathcal{G}_f} I_m(t)$, we get

$$\begin{aligned}
&\max_{\pi,S}\sum_f \mathbb{E}[\Delta N_f(t)]\\
&= \max_{\pi,S}\sum_f \mathbb{E}\Big[\sum_{m\in\mathcal{G}_f} I_m(t)\Big] \;=\; \max_{\pi,S}\sum_{m\in\cup_f\mathcal{G}_f} p_m(t)\\
&= \max_{\pi,S}\sum_{m\in\cup_f\mathcal{G}_f}\Big(p_m(t) - p_m(t-1) + p_m(t-1)\Big)\\
&= \max_{\pi,S}\Big(\sum_{m\in\mathcal{G}(\pi,S)}\Delta p_m(t) + \sum_{m\in\cup_f\mathcal{G}_f\setminus\mathcal{G}(\pi,S)}\Delta p_m(t)\Big)\\
&\quad + \sum_{m\in\cup_f\mathcal{G}_f} p_m(t-1) \qquad\qquad (6)\\
&= \max_{\pi,S}\sum_{m\in\mathcal{G}(\pi,S)}\Delta p_m(t) + K_1(t) + K_2(t-1), \qquad (7)
\end{aligned}$$

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

Example 4.2 (D1. Min-Delay):

$$\begin{aligned}
\max_{\pi,S}\sum_f \mathbb{E}[\Delta D_f(t)] &= \max_{\pi,S}\sum_{m\in\cup_f\mathcal{G}_f}\mathbb{E}[\Delta D_f(t)]\\
= \max_{\pi,S}\Big(\sum_{m\in\mathcal{G}(\pi,S)}\mathbb{E}[\Delta D_f(t)] &+ \sum_{m\in\cup_f\mathcal{G}_f\setminus\mathcal{G}(\pi,S)}\mathbb{E}[\Delta D_f(t)]\Big)\\
&= \max_{\pi,S}\sum_{m\in\mathcal{G}(\pi,S)}\mathbb{E}[\Delta D_f(t)] + K(t). \qquad (8)
\end{aligned}$$

Again, $K(t)$ which does not depend on (π, S) is a constant at time t. Thus, minimizing delay in a snapshot is finding a set of the copy and link schedule which maximizes the sum of improvement in expected delay. Note that the improvement in the expected delay at time t of a packet-company m is evaluated as $\mathbb{E}[\Delta D_f(t)] = \int_{s=0}^{\infty} sp_m(t-1+s)ds - \int_{s=0}^{\infty} sp_m(t+s)ds$.
We can show the similar examples for the objectives **R2** and **D2** which can be trivially extended from the **R1** and **D1** respectively.

OPT 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.*[3] 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 [35] for details) that can be reduced to the NP-hard MWIS (Maximum Weighted Independent Set) problem

5 Distributed Max-Contribution

5.1 Max-Contribution

Complex coupling between copy and link scheduling happens when multiple copies of the same packet are scheduled over different links simultaneously. In our approximation, *Max-Contribution*, OPT is solved with the set $\Pi'(t)$ of copy schedules, where

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

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 5.2.

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 $m_l^\star$ 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 $|\cup_f \mathcal{G}_f|/|(S(t))|$ gets larger. Thus, MC is near-optimal when the offered load in the network is high compared to the number of schedulable links.

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
        $m_l^\star \quad = \quad \arg\max_m \Delta v_m^l(t)$
Step 3. Link scheduling.
    Select the schedule $S^\star(t)$ that satisfies
        $S^\star(t) = \arg\max_{S \in \mathcal{S}(t)} \sum_{l \in S} W_l(t),$    (9)
Step 4. Packet copying.
    Replicate the packet (or the copy) $m_l^\star$ over the link $l$, for all $l \in S^\star(t)$.
```

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 problem[4].

5.2 Distributed Max-Contribution (DMC)

Copy scheduling. The main difficulty of MC is that all nodes holding a copy of a packet company m should have the same value of v_m. DMC addresses such a challenge through an on-line approximation 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

[3] Throughout this paper, when we mention global knowledge, we mean global knowledge of values.

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

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

$$v_{m,i}(t) = p_{m,i}(t) = 1 - \prod_{k \in \mathcal{N}_{m,i}} (1 - q_{m,k}(t)). \tag{10}$$

For making 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 $m^{\star}_{i,j}$ where m is scheduled for copy for a link between nodes i and j. We call $m^{\star}_{i,j}(t)$ the *candidate copy* of node i at time t.

Link scheduling. The scheduling algorithm that solves Eq. (9), 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 [35] the references therein for the detailed algorithm description). 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. Note that this type of greedy algorithm is known to achieve constant fraction approximation to the optimal max weight scheduling [36] and also shown to be implementable in a CSMA fashion [37].

5.3 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. But this has a tendency of improving the performance.

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 that 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).

DF and DMC: Comparison. As briefly discussed in Section 2, *Delegation Forwarding* (DF) [16] is known to efficiently save cost while maintaining reasonable delivery ratio. DMC can be tunable so that it approximates DF by setting a threshold value of DMC that equalizes asymptotic number of copies in DMC with that in DF as follows (whose formal derivation is presented in Appendix in a supplement material):

$$T = \theta(p_{m,i}) = 1 - a(1 - p_{m,i})^{C}, \tag{11}$$

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 $C \geq 0$. Recall that link scheduling under limited transmission opportunities is not considered in DF. Our simulation comparing the costs of DMC-threshold (i.e., DMC with a threshold) to DF shown in Section 6 confirms that the cost of DMC-threshold similar to that of DF.

6 Performance Evaluation

6.1 Node Delivery Probability from Real Traces

To evaluate performance, we use the GPS traces of taxies in two cities: Shanghai [9] and San Francisco [38]. First, the traces in Shanghai are collected from over 4000 taxies, which is by far the largest vehicular GPS traces publicly available. The location information of each taxi is recorded every 40 seconds within an area of $102km^2$ for 28 days (4 weeks). Second, the traces in San Francisco are measured from 536 taxies recorded every 30 seconds for 25 days. We consider a DTN application where many infostations are randomly scattered around the city in a uniform manner and using a mobile network of taxies 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 taxies. 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 scenario.

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 [21], [39]. 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 taxies. Further, we find that taxies exhibit some biases in choosing locations they visit and thus other taxies they meet and that different taxies tend to have different biases. The bias in regularity makes different taxies to have different contributions in our framework. To illustrate this, we plot the CCDF (complementary cumulative density function) of the inter-contact times (ICT) and inter-visit times (IVT) of taxies in Shanghai traces in Fig. 1. The aggregate ICT and IVT distributions are best fitted with exponential distributions as depicted Figs. 1 (a) and (b) of semi-log scale. From Figs. 1 (c) and (d), the same patterns are also verified for individual pairs This is quite different from the human mobility pattern which shows power-law inter-contact time distributions. The exponential tail of vehicular ICT is also shown in [40]. Figs. 2 (a) and (b) demonstrate the pairwise intensity values (λ^{IVT} and λ^{ICT}) of IVT exponential distributions of the pairs between 100 destination locations and 100 taxies and ICT distributions among randomly chosen 100 taxies Figs. 2 (c) and (d) [5] show PDFs of all aggregated λ^{IVT} and λ^{ICT} values compared to PDFs of the values obtained from an individual taxi. From the plots, we find that different taxies show different biases in the locations they visit and in the set of taxies 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,

$$q_{m,i}(t) = \max\{q^1_{m,i}(t), q^2_{m,i}(t), q^3_{m,i}(t), ...\} \tag{12}$$

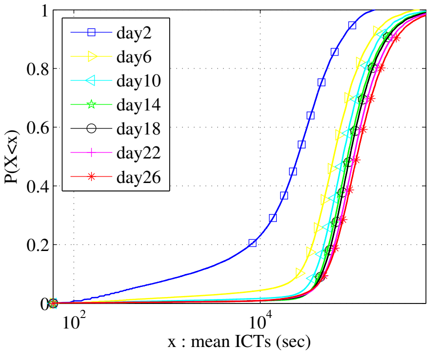


Fig. 3. CDFs of average ICT values of all taxi pairs, which are accumulated over different amount of days. We observe that λ for all taxi pairs almost become constants after aggregating ICT samples for more than two weeks. This regularity enables us to compute $q_{m,i}(t)$ online.

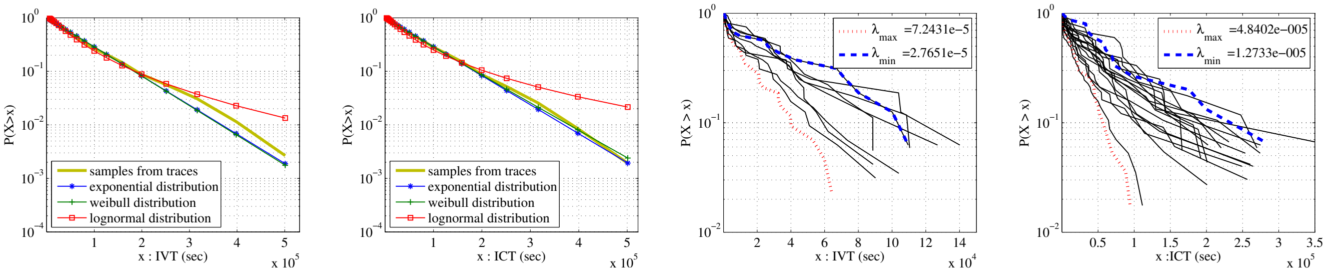
where $q^k_{m,i}$ denotes the delivery probability through k hops. For example, 1-hop probability, $q^1_{m,i}(t)$ is the probability that node i directly meets the destination location, $d(m)$ during the interval $[t, t_{\text{dl}}]$. For 2-hops or more, we find the path with the maximum delivery probability 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).

$$\begin{aligned} & q^k_{m,i}(t) = \\ & \max_{\{n_2,\ldots,n_k\} \in \mathcal{N}^{k-1}} \Bigg\{ \int_{t_{n_{k-1},n_k}}^{t_d - t} \cdots \int_0^{t_d - t} \mathbb{P}[T_{n_1,n_2} = t_{n_1,n_2}] \\ & \prod_{j=2}^{k-1} \mathbb{P}[T_{n_j,n_{j+1}} = t_{n_j,n_{j+1}} - t_{n_{j-1},n_j}] \\ & \mathbb{P}[T_{n_k,n_{k+1}} \le (t_d - t) - t_{n_{k-1},n_k}] dt_{n_1,n_2} \cdots dt_{n_{k-1},n_k} \Bigg\} \end{aligned} \tag{13}$$

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).

Feasibility of online computation of link statistics. Whenever a node contacts other nodes, they first share their metadata listing packet-companies they have and their delivery probabilities to all destinations. Based on this information, nodes calculates their contributions to determine who is the best candidate to get replicated and when the replication can be performed. Therefore, the required computation on the fly is just the simple contribution calculation, given node delivery probabilities, $q_{m,i}(t)$. Similarly to previous work [14], [23], we compute the probabilities using ICT and IVT distributions as shown in Eq (12) and (13). It is true that the equations can be too heavy to be computed on the fly if a node i computes $q_{m,i}(t)$ from scratch whenever it is requested. However, once the ICT and IVT distributions become stabilized to have almost constant λ values for all contact pairs, we can apply online approximation of link statistics (i.e., $q_{m,i}(t)$) by distributing the computation of Eq (12) and (13) over time similarly to that of link state routing algorithms finding a shortest path. To enable this, a node remembers only the best route information (i.e., the sequence of nodes resulting in the maximum node delivery probability) to each of possible destinations

[5] We excluded pairs meeting rarely (i.e., pairs with λ^{IVT} or λ^{ICT} values smaller than 0.00003 or 0.00005 respectively).



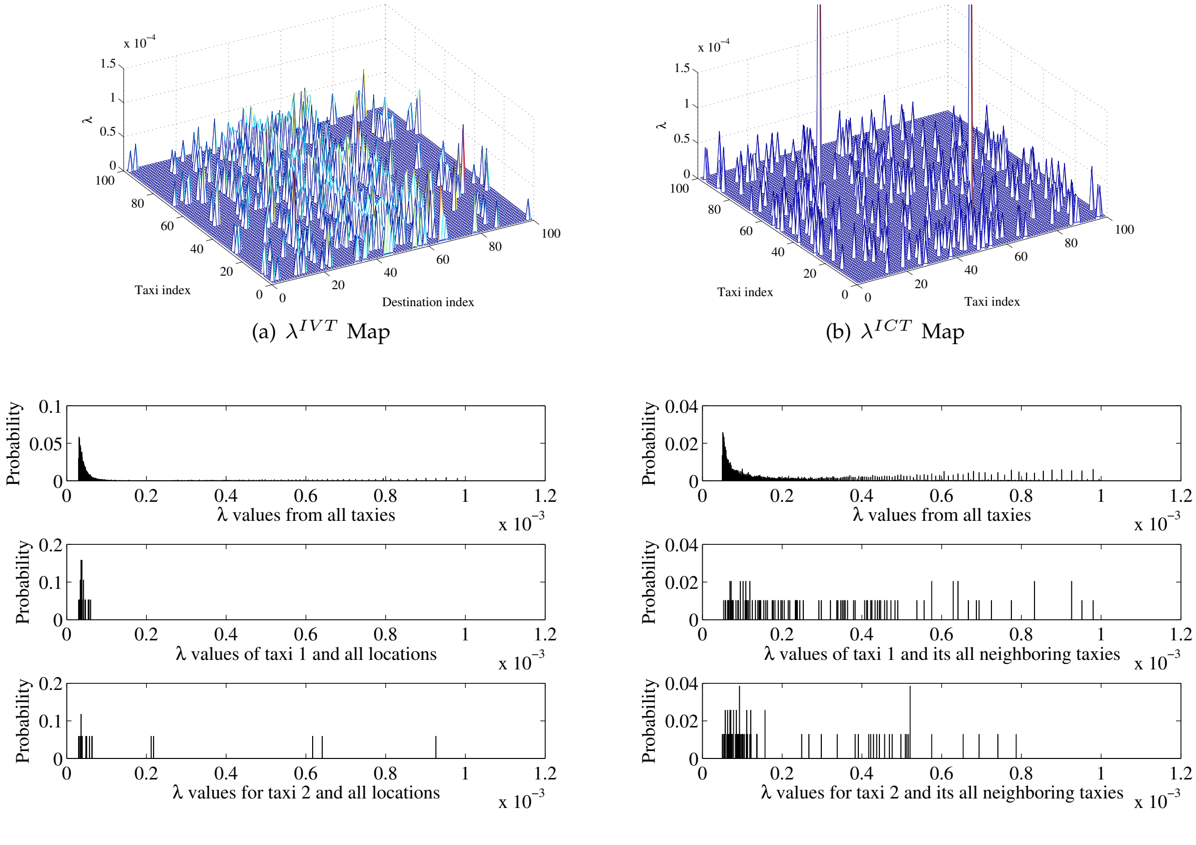
(a) aggregate IVT

(b) aggregate ICT

(c) Individual IVT

(d) Individual ICT

Fig. 1. (a),(b) CCDFs of aggregate IVT distribution and aggregate inter-contact time ICT distributions of all taxies to all candidate locations and to all other taxies. They are tested with exponential, Weibull and log normal distributions by maximum likelihood estimation (MLE) and exponential distribution showed the best fit. (c),(d) CCDFs of IVT distributions and ICT distributions of a taxi to locations and to other taxies. The maximum and minimum intensity of the best fitting exponential distributions are given as λ_{min} and λ_{max}, respectively.



(a) λ^{IVT} Map

(b) λ^{ICT} Map

(c) PDF of λ_{IVT}

(d) PDF of λ_{ICT}

Fig. 2. (a),(b) We plot the individual intensity values (λ^{IVT} and λ^{ICT}) of IVT and ICT exponential distributions from 100 taxies. IVT is plotted for 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. Different taxies show different biases in the locations they visit and in the set of taxies they meet. (c) We plot PDFs of λ_{IVT} from all taxi and 100 location pairs and location pairs of an individual taxi. This verifies that visit rates are unevenly distributed over taxies. (d) We perform the same for λ_{ICT}.

and compares the route with the new route established through a node in contact for the same destination. When the new route is determined to have better performance, the node updates the route to remember with new one. In Fig. 3, we showed that CDF of average ICT values for all taxi pairs in Shanghai traces after aggregating samples for different amount of days. As expected by the regularity of mobility [41], [42], the CDF becomes stabilized after around two weeks resulting in constant λ values. We also verified that distributed computation of $q_{m,i}(t)$ updated by the aforementioned method makes the probability converged to that computed in offline through Fig. 4.

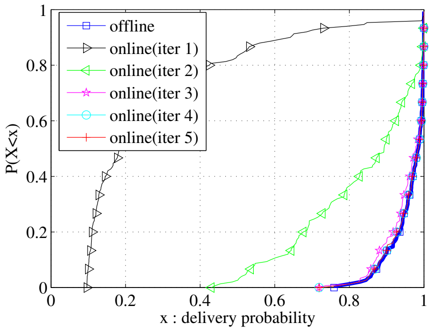


Fig. 4. The CDF of node delivery probabilities for 150 nodes to a destination computed by our online update method is compared to that from the exhaustive offline computation method. After several iterations of updating the best route, the node delivery probabilities of 150 nodes converged to the results from offline computation.

6.2 Setup, Metric and Tested Algorithms

We implemented a resource allocation simulator for a DTN using MATLAB. Among over 4000 taxies in Shanghai traces, we selected relatively reliable 1486 taxies that show less than 30% of unreliable GPS coordinates in a day [6] before performing interpolation. For San Francisco traces, we selected 536 taxies as they all show reliable GPS coordinates. 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 1. *MC-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 4.

6.3 Metadata overhead

In this subsection, we analyze the amount of metadata for sharing the information N_m. In order to track $N_{m,i}$ in a distributed way, when node i and j meet each other, they exchange $N_{m,i}$ and $N_{m,j}$ for a packet m and all other packets they hold. Thus, the size of meta data from node i simply becomes $O(nM)$, where n is the number of nodes and M is the number of packet companies.

The impact of the overhead in the scale of $O(nM)$ in a practical scenario can be estimated as follows. In case of Shanghai trace where the number of nodes is 1486 and the total number of packet companies is 16000, according to the evaluation result in Fig.7, the number of copy events of DMC made until the deadline is about 800,000. The number tells that the average number of packets contained in each node is about 540 and the average number of nodes having a certain packet is about 50. If we encode the IDs of nodes and packets into bits, 11 bits and 14 bits will be occupied, respectively. Therefore, the average size of a metadata to be exchanged in each encounter can be estimated as $14bits \times 540 \times 11bits \times 50 = 520kByte$. In the same way, San Francisco scenario where N is 536 and M is 3200 gives the average size of a metadata as $125kB$. In the case of making a resource allocation decision at every 30 seconds with transmission rate of about 1Mbyte/sec, the metadata of $520kB$ and $125kB$ would consume 1.7% and 0.4% of the given air time, respectively. A faster link speed will contribute to a smaller portion of air time. Based on this observation, we proceed our simulation studies with the overhead from the metadata exchange excluded.

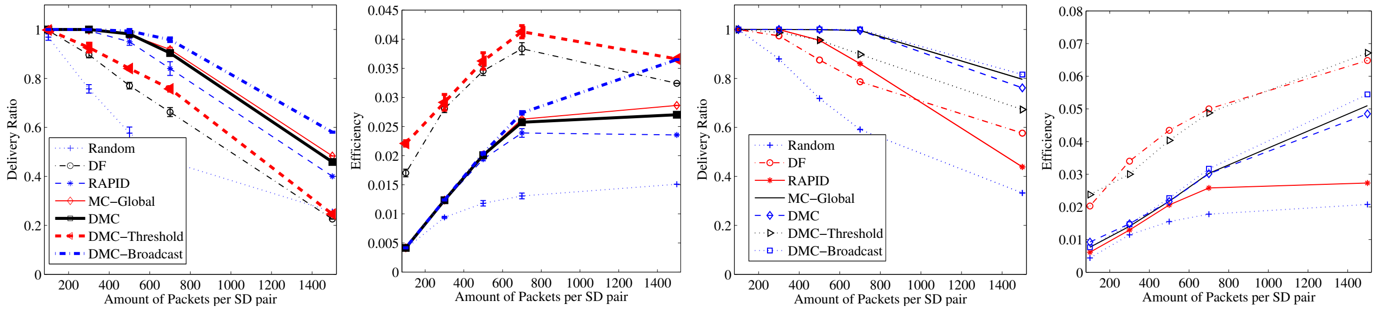
6.4 Simulation Results

For max-delivery objective **R1**, Fig. 5(a) and (b) show the delivery ratio and efficiency of scheduling algorithms against the offered load (i.e., the amount of packets injected to each of S-D pair) in Shanghai taxi traces. The delivery ratio decreases as the offered load (the

[6] Reliability information, DOP (dilution of precision) is included in the traces

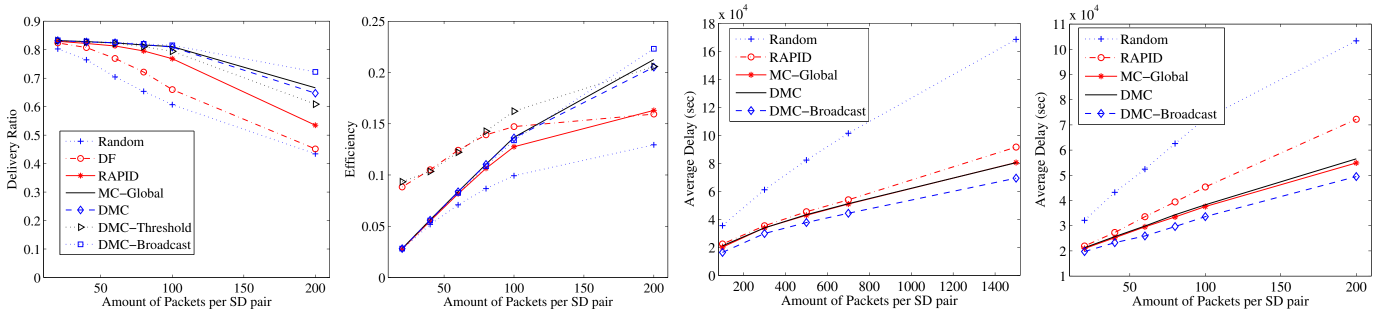
TABLE 1
Tested Algorithms (⋆ corresponds to the items that we added for fair comparison)

Algorithm	Link scheduling	Copy scheduling	Value update	Required knowledge	Limitation
Random	random	random	×	×	heuristic
DF [16]	⋆greedy	⋆difference.	delegation	local metrics	heuristic
RAPID [14]	⋆random	contribution	global	global metrics	global knowledge sharing
MC-Global	greedy	contribution	global	global metrics	global knowledge sharing
DMC	greedy	contribution	fusion	local metric	local approximation
DMC-Threshold	greedy	contribution with threshold	fusion	local metrics	local approximation
DMC-Broadcast	greedy	contribution with broadcast	fusion	local metrics	local approximation



(a) Delivery Ratio (b) Efficiency (c) Delivery Ratio (d) Efficiency

Fig. 5. [Shanghai traces, R1] (a) and (b) Delivery ratio and efficiency of algorithms listed in Table 1 for varying offered load to 32 S-D pairs with a radio range of 300 meters. Each value shows 95% confidence interval. We do not show cost as it is implied in the efficiency. (c) and (d) Delivery ratio and Efficiency under a different radio range, 500 meters.



(a) Delivery Ratio (b) Efficiency (c) Delay, Shanghai (d) Delay, San Francisco

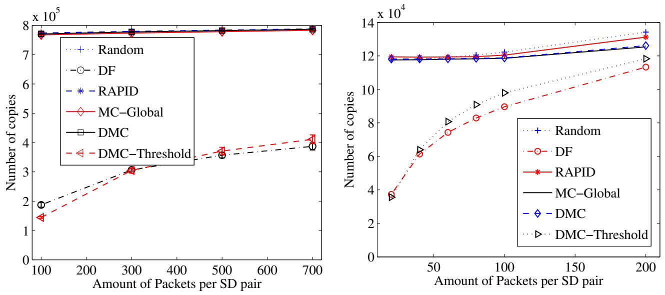
Fig. 6. [San Francisco traces, R1] (a) and (b) Delivery ratio and efficiency of algorithms listed in Table 1 for varying offered load to 200 S-D pairs. (c) [Shanghai traces, D1] Delay of algorithms for varying offered load to 32 S-D pairs in Shanghai. (d) [San Francisco traces, D1] Delay of algorithms for varying offered load to 200 S-D pairs in San Francisco.

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. While other algorithms copy a packet to a single relay node, DMC-broadcast additionally copies a packet to all others nodes in its neighborhood when replicating a packet to the selected relay node. Hence, even though DMC-broadcast lacks global knowledge, more copies generated in the network by DMC-broadcast allows it perform better than MC-Global. 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 shown in Fig. 1 and Fig. 2. We test the performance in a denser environment to see the impact of joint forwarding and link scheduling. We vary the radio range from 300 meters (see Figs. 5(a) and (b)) to 500 meters (see Figs. 5(c) and (d)). We observe that the gap between DMC and RAPID increases as the network density increases. They show difference in the following two points: 1) DMC uses more lightweight metadata dissemination called Fusion than RAPID which uses flooding, and 2) DMC relies on greedy link scheduling while RAPID uses random link scheduling. In general, the effect of 1) is minimal because the DMC and MC-Global performs similarly. The performance gap between the two is likely to come from 2). Denser network leads to increasing interference in transmissions. Thus, we observe that the performance gap reaches about two times. The performance of DMC improves with the increasing radio range due to a higher chance of meeting other nodes.

To reconfirm the performance of our proposed algorithms, we repeat the same simulations for the San Francisco taxi traces with a radio range of 300 meters. For **R1** objective, Figs. 6(a) and (b) show the delivery ratio and efficiency of scheduling algorithms for various offered loads, where we use taxies as sources and destinations instead of infostations. The performance of algorithms show similar order with that observed in Shanghai traces. DMC-broadcast outperforms others and DMC works as good as MC-Global. The performance gap between DMC and RAPID is about 20% under high offered load. To sum up, the overall performance gap between DMC and RAPID in San Francisco traces is larger than Shanghai traces mainly due to the impact of link scheduling. Indeed, the average number of interfering neighbors in San Francisco traces are 3.5 which is much higher than 2.1 observed in Shanghai traces.

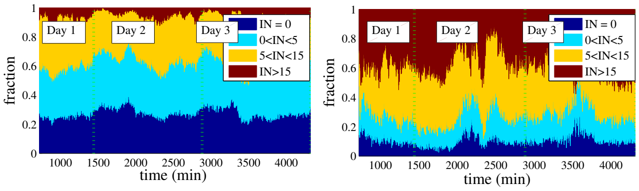
Also, we compare the algorithms for the min-delay objective **D1** using the Shanghai traces. The delay of each algorithm is tested against the offered load to randomly chosen 32 S-D pairs under a radio range of 300 meters. As shown in Fig. 6(c), DMC again closely follow the delay performance of MC-Global. On the other hand, Random is far from the other three algorithms and the delay gap of RAPID and DMC becomes larger with the increasing number of offered packets. We also test the algorithms for **D1** objective in San Francisco traces. Fig. 6(d) shows the delay performance where DMC closely follows MC-Global while RAPID and Random show much larger delivery delays.

The overhead estimation from the algorithms in the aspect of energy consumption and memory occupancy boils down to counting the average number of packet replications (i.e., transmissions). Fig. 7 plots the number of replications over the amount of packets per SD pair under the deadline of one day in Shanghai and San Francisco. This confirms that Random, RAPID, MC-Global and DMC show almost the same number of replications simply because they all try to utilize the allowed air time as much as possible. On the contrary, DF and DMC-Threshold show less numbers of replications as they aim to lower the overhead. In conclusion, MC and DMC do not inflate the energy consumption and memory occupancy compared to RAPID while DMC-threshold is successful in suppressing such overheads to the level of DF.



(a) Shanghai (b) San Francisco

Fig. 7. The number of copies of algorithms listed in Table 1 for varying offered load to (a) 32 S-D pairs in Shanghai and (b) 200 S-D pairs in San Francisco.



(a) Shanghai (b) San Francisco

Fig. 8. (a) The fraction of number of interfering nodes (IN) experienced by transmitting nodes over time in Shanghai. The average number of interfering nodes is 4.5. (b) The fraction of number of interfering nodes (IN) in San Francisco. The average number of interfering nodes is 12.5.

We further justify the need of link scheduling for interference management by showing the existence of contention for wireless channel. Fig. 8 shows the fraction of the number of interfering nodes in (a) Shanghai and (b) San Francisco traces over time. Recall that we assume two-hop interference model and use the typical radio range of WiFi, 300 meters. The blue region shown in the bottom of the figures (i.e., 'IN=0') indicates that the portion of transmitting nodes which do not interfere with other nodes. Other regions represent the portions of transmitting nodes with the notated number of interfering nodes. In both traces, we observe that less than 25% of transmitting nodes transmit packets without any interference and more than 40% of nodes compete with more than five interfering nodes. The reason why the number of interfering nodes in San Francisco is much greater than that in Shanghai is that most of taxis in San Francisco move closer to each other as they mostly stay in the downtown area. The statistics on the interfering nodes clarifies the reason why MC and DMC obtain performance gains over existing algorithms.

7 Concluding Remarks

7.1 Discussion

In this paper, we proposed a resource allocation algorithm OPT based on snapshot approximation and provided approximated and distributed algorithms, MC and DMC accordingly. Obviously, OPT is optimal in a time frame but is not the optimal algorithm when considering the entire time frames. We discuss about the remaining challenges in developing the optimal DTN resource allocation algorithm which may require a set of following additional information.

Performance gap to the optimality. In order to achieve the optimality via dynamic programming formulation in DTNs with inference, what we need to model includes the statistics on how long each connection would last, how long each disconnection would last, how many nodes would exist in the interference range when each encounter happens, how many packets are queued in each node, what would be the list of those packets, and who are their destination nodes. Due to the severe complexity involved in the optimality, our framework in DMC only adopted the inter-connection time distribution for calculating the delivery probability.

Quantifying the performance gap between DMC and the optimality is mathematically intractable, but we at least know when our approximation becomes close to the optimality. The only difference from the optimal backward induction of dynamic programming to our solution is that we calculate the delivery probability using the inter-connection statistics whereas the backward induction will calculate the probability with statistical knowledge on interfering nodes, packet compositions in those interfering nodes, and the allowed time durations for the encounters. Given that Max Contribution tries to maximize delivery probabilities of packets for a given time slot (i.e., a given encounter duration), the fact that the chosen sequence of scheduling by Max Contribution is different from the sequence chosen by the backward induction makes the performance gap. Therefore, if the duration of each encounter is sufficiently long to allow all the recommended forwarding irrespective of its sequence, our algorithm can work closely to the optimality.

Detailed statistics on contact patterns. In most of research work in DTNs including our work, ICT distributions are accumulated over several weeks to capture regularity in mobility patterns of humans or human-driven vehicles. According to [42], regularity can be observed in both a daily scale and a weekly scale. Considering human mobility behaviors in their daily lives, observing a new regularity in hourly scale is also possible when we condition a day of the week. Hence, predicting future events statistically based on long-term aggregated traces can lead to suboptimal resource allocations if a packet is delivered from a source to a destination within much shorter time scale than that of aggregated traces or vice versa. Thus, the optimal DTN resource allocation needs to leverage more detailed statistics on contact patterns, adaptively to the time scale of packet delivery.

Statistics on the number of interfering nodes. When a DTN has light traffic affordable within each of contact durations, the dynamic programming approach introduced in [12], [13] relying on the statistics of ICT, is a good enough framework in developing the optimal resource allocation algorithm. However, when the traffic volume becomes heavy, new statistics on the number of interfering nodes needs to be considered in the framework resulting in substantial increment in complexity. For example, when there are two paths of which the first one meets the destination more frequently but has extremely many interfering nodes when meeting and the second one meets less frequently but the meeting is exclusive, the optimal algorithm should jointly consider the frequency of meeting and the chance of being scheduled when meeting. Unfortunately, there are only few observations on the interfering nodes and no framework can handle this issue properly. However, recent work [43] which applied mean field theory to predict evolution of cluster sizes of mobile nodes can be a theoretical foundation to this issue as the theory is shown to predict the number of nodes in a cluster (i.e., neighborhood) very closely to that in reality. We are interested in combining mean field theory and resource allocation problem over multiple time frames in DTNs as our future work.

The effect of encounter duration. We characterize the effective duration of each encounter during a time slot by calculating the amount of time being in the range of each other. As the traces are recorded at every 30 or 40 seconds, for the quantification, we estimated the detailed positions via interpolation of the recorded positions. Shanghai and San Francisco showed 17 and 10 seconds of effective encounter duration out of a time slot of 30 seconds [7], respectively. Thus, to obtain more realistic simulation results, the size of packet should be adjusted accordingly.

7.2 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 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,

[7] The effective encounter duration is defined as 0.1-percentile value of all connection durations.

we systematically approach the problem; we use the snapshot approximation in temporal dimension, which restricts nodes to use only contemporarily available knowledge, and then approximate various components in spatial dimension to reduce its complexity without much loss in the performance. Our approach clearly shows how we derive our heuristic solutions and provides some confidence over the expected performance. Another contribution is that we demonstrate how our developed solutions can be applied to solving real world problems, such as information dissemination over a network of over 1000 taxies, 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 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.

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, Georgia Institute of Technology*, 2004.

[2] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," technical Report, CS-200006, Duke University, April 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, 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 VTC*, 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 MobiHoc*, 2003.

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," vol. 16, no. 1, 2008, pp. 63–76.

[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., "Evaluating mobility pattern space routing for DTNs," in *Proceedings of IEEE INFOCOM*, 2006.

[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 protocolin 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 SIGCOMM*, 2007.

[15] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings First IEEE International Workshop Sensor Network Protocols and Applications*, 2003.

[16] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proceedings of ACM MobiHoc*, 2008.

[17] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networking," in *Proceedings of IEEE INFOCOM*, 2006.

[18] 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.

[19] K. Lee, Y. Yi, I. Rhee, J. Jeong, H. Won, and S. Chong, "Max-contribution: On optimal resource allocation in delay tolerant networks," in *Proceedings of IEEE INFOCOM*, San Diego, CA, 2010.

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

[21] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong, "SLAW : A new human mobility model," in *Proceedings of IEEE INFOCOM*, 2009.

[22] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of ACM MobiHoc*, San Diego, CA, 2007.

[23] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *Proceedings of ACM Mobihoc*, 2008.

[24] P. Luo, H. Huang, W. Shu, M. Li, and M. Wu, "Performance evaluation of vehicular DTN routing under realistic mobility models," in *Proceedings of IEEE WCNC*, 2008.

[25] H. Huang, P. Luo, M., D. Li, X. Li, W. Shu, and M. Wu, "Performance evaluation of SUVnet with real-time traffic data," in *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, 2007.

[26] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner, "SUMO (simulation of urban mobility): An open-source traffic simulation," in *Proceedings of SCS Middle East Symposium on Simulation and Modelling (MESM)*, 2002.

[27] H. Soroush, N. Banerjee, A. Balasubramanian, M. D. Corner, B. Levine, and B. Lynn, "Dome: A diverse outdoor mobile testbed," in *Proceedings of ACM HotPlanet*, 2009.

[28] Q. Li and D. Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, 2000.

[29] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, "Capacity enhancement using throwboxes in DTNs," in *Proceedings IEEE Mobile Adhoc and Sensor Systems (MASS)*, 2006.

[30] W. Zhaoa, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, "Hybrid routing in clustered DTNs with message ferrying," in *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, 2007.

[31] M. Chuah, P. Yang, B. Davison, and L. Cheng, "Store-and-forward performance in a DTN," in *Proceedings of IEEE Vehicular Technology Conference*, 2006.

[32] M. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proceedings of ACM MobiHoc*, 2006.

[33] W. M. M. Chuah, "Integrated buffer and route management in a DTN with message ferry," *Journal of Information Science and Engineering*, vol. 23, no. 4, pp. 1123–1140, 2007.

[34] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of ACM MobiHoc*, 2004.

[35] Y. Yi, A. Proutiere, and M. Chiang, "Complexity in wireless scheduling: Impact and tradeoffs," in *Proceedings of ACM Mobihoc*, 2008.

[36] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 4, pp. 1132–1145, 2009.

[37] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. W. Knightly, and M. Chiang, "Experimental evaluation of optimal csma," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1188–1196.

[38] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD data set epfl/mobility (v. 2009-02-24)," Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility, feb 2009.

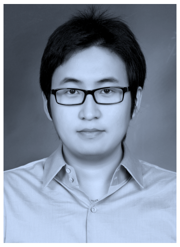
[39] M. Kim and D. Kotz, "Periodic properties of user mobility and access-point popularity," *Journal of Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 465–479, August 2007.

[40] H. Zhu, M. Li, L. Fu, G. Xue, Y. Zhu, and L. Ni, "Impact of traffic influxes: Revealing exponential intercontact time in urban vanets," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 8, aug. 2011.

[41] M. Kim and D. Kotz, "Periodic properties of user mobility and access-point popularity," *Personal Ubiquitous Computing*, vol. 11, no. 6, pp. 465–479, 2007.

[42] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, “Modeling time-variant user mobility in wireless mobile networks,” in *Proceedings of INFOCOM*, Anchorage, AL, May 2007.

[43] S. Heimlicher and K. Salamatian, “Globs in the primordial soup: The emergence of connected crowds in mobile wireless networks,” in *Proceedings of ACM MobiHoc*, Chicago, IL, 2010.



Kyunghan Lee received the B.S., M.S., and Ph.D. degrees in the Department of Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2002, 2004, and 2009, respectively. He is currently an assistant professor in the School of Electrical and Computer Engineering at UNIST (Ulsan National Institute of Science and Technology), Ulsan, South Korea. Prior to joining UNIST, he has been with the Department of Computer Science, North Carolina State University, Raleigh, US as a senior research scholar. His research interests include the areas of human mobility modeling, delay-tolerant networking, information centric networking, context-aware service design, and cloud-powered network service design.

Jaeseong Jeong received the B.S., M.S., and Ph.D. degrees in the Department of Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2008, 2010, and 2014, respectively. He is currently a postdoctoral researcher in Automatic Control Department in KTH, Stockholm, Sweden. His research interests include human mobility/behavior analysis and prediction, protocol design and implementation for mobile networks and vehicular sensor networks.

Yung Yi received his B.S. and the M.S. in the School of Computer Science and Engineering from Seoul National University, South Korea in 1997 and 1999, respectively, and his Ph.D. in the Department of Electrical and Computer Engineering at the University of Texas at Austin in 2006. From 2006 to 2008, he was a postdoctoral research associate in the Department of Electrical Engineering at Princeton University. Now, he is an associate professor at the Department of Electrical Engineering at KAIST, South Korea. He has been serving as a TPC member at various conferences including ACM Mobihoc, Wicon, WiOpt, IEEE Infocom, ICC, Globecom, and ITC. His academic service also includes the local arrangement chair of WiOpt 2009 and CFI 2010, the networking area track chair of TENCON 2010, and the publication chair of CFI 2010, and a guest editor of the special issue on Green Networking and Communication Systems of IEEE Surveys and Tutorials. He also serves as the co-chair of the Green Multimedia Communication Interest Group of the IEEE Multimedia Communication Technical Committee. His current research interests include the design and analysis of computer networking and wireless systems, especially congestion control, scheduling, and interference management, with applications in wireless ad hoc networks, broadband access networks, economic aspects of communication networks economics, and greening of network systems

Hyungsuk Won received his Bachelor degree in Computer Engineering from Kyungpook National University, Korea in 1997, and M.S. degree in Computer Science from POSTECH, Korea in 1999 and Ph.D degree in in Dept. of Computer Science at North Carolina State University in 2010. His research interests include multicast and MAC layer scheduling in wireless networks, routing in intermittently connected networks and delay tolerant networks, information retrieval, and security.

Injong Rhee (M’89) received his Ph.D. from the University of North Carolina at Chapel Hill. He is a professor of Computer Science at North Carolina State University. His areas of research interests include computer networks, congestion control, wireless ad hoc networks and sensor networks.

Song Chong is a Professor in the Department of Electrical Engineering at Korea Advanced Institute of Science and Technology (KAIST) and the founding Director of KAIST-LGE 5G Mobile Communications & Networking Research Center funded by LG Electronics. Prior to joining KAIST in March 2000, he was with the Performance Analysis Department, AT&T Bell Laboratories, Holmdel, New Jersey, as a Member of Technical Staff. His current research interests include wireless networks, mobile networks and systems, network data analytics, distributed algorithms, and cross-layer control and optimization. He is currently an Editor of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing and IEEE Transactions on Wireless Communications, and has served on the Technical Program Committee of a number of leading international conferences including IEEE INFOCOM, ACM MobiCom, ACM CoNEXT, ACM MobiHoc, and IEEE ICNP. He serves on the Steering Committee of WiOpt and was the General Chair of WiOpt ’09. He received the IEEE William R. Bennett Prize Paper Award in 2013, given to the best original paper published in IEEE/ACM Transactions on Networking in 2011-2013, and the IEEE SECON Best Paper Award in 2013. He received the B.S. and M.S. degrees from Seoul National University and the Ph.D. degree from the University of Texas at Austin, all in electrical engineering.