ExMin: A Routing Metric for Novel Opportunity Gain in Delay Tolerant Networks

Jaeseong Jeong[a], Kyunghan Lee[b], Yung Yi[a], Injong Rhee[c], Song Chong[a]

[a] *Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea*
[b] *School of ECE, Ulsan National Institute of Science and Technology, Ulsan, Korea*
[c] *Dept. of Computer Science, North Carolina State University, Raleigh, NC, 27695 USA*

Abstract

Delay Tolerant Networks (DTNs) are characterized by intermittently connected links formed by mobile nodes' probabilistic encounters. Most DTN routing techniques use the first encountered node who has smaller routing metric as a relay node. Prior work on DTN routing can be broadly classified into one which takes the minimum out of expected delays for all possible individual routing paths, referred to as MinEx, as a routing metric to decide the next hop relay node. Fundamentally, MinEx has no difference from the shortest path computation in conventional multi-hop networks, where a link weight is the expected inter-meeting time. However in DTNs, nodes meet intermittently by their mobility, hence the links formed from the meetings are probabilistic. In this environment, MinEx often fails to accurately estimate the actual delay since opportunism in nodes' intermittent meeting is not properly taken into account. In this paper, to exploit the true opportunism, we first propose a metric called ExMin which stochastically calculates the metric by taking the expectation of the minimum delays over all possible routes. We further show that ExMin can be computed online by relying only on local information sharing. Our extensive experiments involving three realistic network scenarios created by two vehicle traces (about 1500 Shanghai taxies and 500 San Francisco taxies) and one human mobility trace (93 KAIST students) show that ExMin outperforms MinEx by up to 30% under either of DTN environments allowing single-copy or multi-copies of a packet.

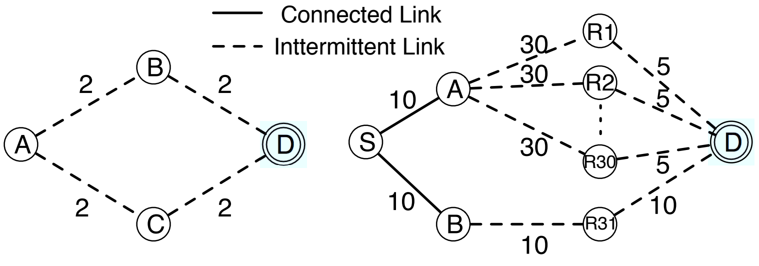
Keywords: Delay Tolerant Networks, Opportunistic forwarding, routing

1. Introduction

DTNs have emerged as an attractive networking paradigm for mobile networks where transmission links among mobile nodes are dynamically established or torn down and communication is accomplished through multi-hop message relaying. Such environments are becoming increasingly prevalent due to the rapid growth of heterogeneous mobile devices and delay-insensitive applications.

A key problem in DTNs is the selection of relay nodes. The problem is unique in DTNs because relays are not necessarily determined by a source a priori. This is due to the probabilistic nature of DTNs where it is unknown in advance whether a node will meet a particular node in the future and use that node as a relay. Since a relay is often determined at the time of meeting, DTN routing is called *opportunistic routing.* This opportunism is different from that used in wireless multi-hop networks (see [1, 2, 3, 4]) which relies on the broadcast nature of wireless medium to choose the next hop forwarder: after broadcasting a packet, the next relay (or forwarder) is selected among the nodes that happen to receive the packets. In such networks, packet receptions are probabilistic while in DTNs, node encounters are.

In DTNs, to choose a relay among its contemporarily established neighbors, each node measures a routing metric for each candidate which is the "cost" of delivery for its packet to the final destination if the packet is forwarded to that candidate relay. Lately there has been an extensive array of studies on both single-copy DTN routing [5, 6, 7] and multi-copy DTN routing [8, 9, 10], most of which propose a method for choosing the best relay node to maximize the probability of packet delivery. These protocols typically work as follows: 1) In case of single-copy routing [5, 6, 7], each packet is forwarded to a relay node so that the number of a packet in the network always remains one. To find the best relay, nodes compute the routing metric of each candidate relay, which is the expected cost of multi-hop forwarding path from each candidate relay to the destination. The cost is computed transitively (e.g., [6]) considering all the possibilities of routing paths starting from that candidate relay. When a node finds a candidate whose cost is the minimum among its contemporary neighbors, the node forwards the packet to the candidate relay and deletes the packet. 2) In case of multi-copy routing [8, 9, 10], each packet is replicated to a relay node allowing that the number of a packet increases as the replication happens. To find the best relay node to replicate a packet, nodes first compute the routing metric of



(a)

(b)

Figure 1: *(a)* An example to illustrate the difference between MinEX and ExMin: the MinEx of node A is 4 whereas ExMin is 3. *(b)* ExMin prefers the path through A over the path through B while MinEx does the opposite. The true expected delay of the two paths through A and B are 6 and 20 respectively.

each candidate relay in the same way as single-copy protocols. Then, for each packet, they compute the utility of replicating the packet to each candidate relay based on the routing metric. When a node finds a candidate relay and the packet whose utility is the best among all possible candidate relays and packets, the node replicates the packet to the candidate relay. In both cases, the routing metric is suitably chosen depending on the design objective such as delay [9], probability of meeting [10] and expected remaining time for meeting another relay or destination [6]. Note that all of these metrics can be ultimately translated into the expected delays for the final delivery.

Practically all the existing DTN routing studies [5, 6, 7, 10] using the metric of expected delay compute the expected delay from a node to the destination of a packet by taking the minimum of the expected delay of each individual route to the destination. We refer to such a metric as *MinEx*. As an illustration, consider a DTN with four nodes A, B, C and D in Figure 1. Each link is *probabilistic* where its associated cost of 2 is the expected delay for its two end nodes to meet in the future. Suppose that the probability distribution of the delays follows an exponential distribution with an intensity 1/2 (i.e., expected link delay is 2). MinEx from A to D yields the cost of 4, taking the minimum of the expected delays possible from A which has two choices of forwarding through either B or C.

MinEx, however, ignores an important aspect of "opportunism" enabled by multiple choices of probabilistic paths. From Figure 1, since A forwards a packet to whichever node it meets first as both nodes B and C have the same cost, the one-hop cost from A to either B or C should be the minimum of two random delays[1] which is 1. Since the second hop cost is 2, the total cost is 3. Therefore, the true expected delay from A is the expectation of the minimum random delays that B and C have for meeting A. We call such a metric the *expectation of the minimum* or *ExMin*. Intuitively, ExMin can be viewed as taking the aggregation of the arrivals of the two possible choices since A chooses as a relay whichever node it meets first out of B and C. We formally define ExMin in Section 4.

Incorrect accounting of the opportunity gain has a significant difference in the end result leading to possibly much less optimal paths. Consider a DTN in Figure 1 (b) where packets are routed from S and D. Suppose that the currently established neighbors to S are A and B. The other links are probabilistic with costs representing the expected inter-meeting times between the two end nodes of the links. S' expected delay through A is 6 $(= 30(1/30) + 5)$ while that through B is 20. However, MinEx yields 35 and 20 as the expected delays through A and B, respectively. Thus, MinEx favors B over A as the relay. But in reality, A is a much better choice since its true expected delay is 6 instead of 35. On the other hand, ExMin is able to choose A over B by exploiting the opportunism given to a relay with many potential neighbors leading to the destination.

To exploit ExMin, a metric computation method, which assigns ExMin values to all the nodes in the network is essential. However, computing it requires global information sharing because it needs to consider all possible opportunistic paths enabled by the multiple hops to a destination. The global information sharing means that all nodes get to know the entire network topology and the cost of each link in the network by flooding meta-data on local topology and local link costs. Since it is obvious that global information sharing incurs significant control overhead, computing accurate ExMin values for nodes without flooding meta-data becomes an important research challenge.

To address such challenges, we aim at developing an online algorithm which progressively (or iteratively) updates ExMin values of nodes by sharing only local information. The key intuition behind the online computation of ExMin lies in the iterations used in traditional distance vector routing

[1] For two exponential random variables $X_i \sim \exp(\lambda_i),\ i = 1, 2,\ Y = \min(X_1, X_2) \sim \exp(\lambda_1 + \lambda_2)$.

algorithm [11]. Our analysis to see whether the online computation assigns ExMin values to nodes close to the values using meta-data sharing or not reveals that the link costs in real DTN scenarios are consistent enough to support iterative metric updates. We claim that the consistency of link costs in DTNs formed by humans or vehicles comes from the *regularity* inherent in mobility patterns of humans and vehicles driven by humans. Our trace-driven simulation results in Section 7 demonstrate that ExMin outperforms existing protocols [5, 7, 10] with MinEx upto 30% in both single-copy and multi-copy scenarios.

The rest of this paper is organized as follows: We first briefly introduce related work in Section 2. We then explain our assumptions, detailed metric computation of ExMin and online ExMin through Section 3,4, and 5. Based on the metrics, we analyze the impact of metrics and evaluate the performance of DTNs when applying our metrics in Section 6 and 7. We conclude our work in Section 8.

2. Related Work

A number of DTN routing studies are based on epidemic routing[12] which floods packets to every connected node. While it guarantees optimal forwarding under no channel contention, flooding incurs huge overhead in terms of channel resource, storage and power consumption in the network. Epidemic routing does not achieve the optimal performance when channel contention is considered. In order to find a DTN routing path without flooding, many popular DTN routing methods[13, 14, 15, 16] are heuristically developed.

For enhancing the performance and reducing the overhead, there are a lot of researches on exploiting utility characteristics assuming a single copy approach in which nodes forward a packet and delete the packet. Most simple algorithm is direct transmission[17]. A node moves with a packet and when it meets the destination of the packet, it delivers the packet to the destination. As mentioned above, we can use last encountered time as a utility when we assume that all the nodes are following random walk mobility model like ECT [18]. Jones et al.[7] propose MED which minimize the end-to-end expected delay in DTN. Compared to ExMin, both of ECT and MED do not account for the effect of the opportunistic paths which are established by multiple future neighbors. Note that, when the inter-contact time follows exponential distribution, ECT and MED can be regarded as MinEx. Lebrun et al.[14] propose a location-based algorithm where utility is characterized by future path. Leguay et al.[19] suggests a pattern-based algorithm. Each node's utility is reflected by pointing each node's pattern in n-dimensional virtual coordinate. Recently there are social-based utilities which is presented by Hui et al.[20] and Elizabeth et al.[21]. One difference between those algorithms and ExMin is that they use additional information such as location, social contacts. [22, 23] uses dynamic programming using backward induction and presents threshold schemes as a forwarding decision method. However, the backward induction works only when there is no transmission contention in the networks and is not properly validated for high node density scenario. Also, they can not compute the forwarding metric in distributed way, but require global information sharing. On the other hand, ExMin can be computed on-line without any global information and applicable to high node density case by incorporating existing DTN resource allocation algorithms.

In order to fully utilize the network resources, previous DTN studies[24, 7, 8, 25, 26, 9, 10] have taken multi-copy routings with their own routing utility metrics. How to optimally allocate routing and scheduling resources in DTN has been focused up to this point. [9] proposes a technique to reorder packets in the transmission queue in order to maximize a given network utility. [10] introduces snapshot optimality which approximates the optimal link and copy scheduling in DTNs. However, when [24, 7, 8, 25, 26, 9, 10] decide which packet to replicate, they use MinEx as the estimated delay of a certain path.

Unfortunately, the existing protocols of both single-copy and multi-copy routings do not consider a routing metric accounting for the path opportunity gain, as ExMin does, enabled by the probabilistic nature of inter-contact times among nodes. Plugging our proposed metric, ExMin into existing routing frameworks, we can show explicitly higher performance rather than using MinEx.

3. Assumptions

We assume that each DTN node is equipped with a wireless radio and a computing device with some amount of storage. Each node can track its current location, either using GPS or GSM. Packets are the basic means of communication and is typically large enough to carry one self-contained message with proper address information about its source and destination. Each node is identified by a unique ID. There are n nodes in the system and all nodes always move within a predefined area. We say that when two nodes are in the radio range of each other, they *meet* each other, or they are *connected*. Because of mobility, the neighbors of each node change over time. We assume that when two nodes are connected, they are connected sufficiently long enough to exchange packets of one node to another and there is no loss of packets. The transmission delays of packets after two nodes are connected are much smaller than the time it waits to meet each other. So we assume that the transmission delays are negligible. The *inter-contact time* (ICT) of two nodes is the time interval between two consecutive contacts. We assume that each node i is aware of its own mobility patterns such as who it "regularly" meets and the distribution of inter-contact times (ICTs) with those nodes j. We mean by "regularly" that their meeting frequencies are statistically significant often to compute the ICT distribution. In section 7, we verify the existence of "regularity" by analyzing the mobility pattern of real traces. For any two nodes that meet regularly v and w, we can define a random variable $I_{v,w}$ representing the inter-contact time. If two nodes have history of regular meetings and they are aware of their ICT distribution, we say they are the *neighbors* of each other. When we represent a *link cost* between two nodes that are not currently connected, we use the mean inter-contact time (ICT), or simply *expected delay* and its distribution. Such links are probabilistic links. When two nodes meet, their delays are zero.

Throughout this paper, we consider a canonical form of opportunistic routing[5, 6, 7, 8, 9, 10] commonly used in DTNs called *CORD*. CORD is used as a reference routing protocol for designing efficient routing metrics. It works as follows. A source node $\mathtt{S}$ has a packet to deliver to a destination node $\mathtt{D}$. When $\mathtt{S}$ meets a set of nodes $\mathcal{N}$ while holding the packet, then it evaluates the following *forwarding conditions*. For each node v in $\mathcal{N}$, it computes a metric $\mathcal{M}(v)$, which is essentially the cost of delivering the packet to $\mathtt{D}$ through v. If (1) $\mathcal{M}(v)$ is less than $\mathcal{M}(\mathtt{S})$ and (2) $\mathcal{M}(v)$ is the minimum among all $\mathcal{M}(l)$, $l \in \mathcal{N}$, then v is chosen as a *relay node* and $\mathtt{S}$ forwards the packet to v. After forwarding, it deletes the packet from its storage. Any relay node holding the packet performs the same operation as $\mathtt{S}$ until the packet is delivered to $\mathtt{D}$. In some cases, condition 1 is checked with a fixed threshold (see [22, 23]). If multiple copies of the packet are allowed, the packet is not deleted after forwarding.

4. Expectation of Minimum(ExMin)

4.1. Definitions

Every prior work we know of that uses expected delays for routing metric uses the minimum of the expected delays. Formally, the minimum of the expected delays from v to the destination is computed as follows. For each node w in the neighbor set of v, we can define the minimum of the expected delays recursively as follows.

$$\mathrm{MinEx}(v) = \min_{w \in v\text{'s neighbors}} \mathbb{E}\Big[I_{v,w} + \mathrm{MinEx}(w)\Big], \tag{1}$$

MinEx($\mathtt{D}$) for destination node $\mathtt{D}$ is zero. We can limit the computation of MinEx only over the paths of k hops or less to the destination. Any paths not reachable to the destination within k hops have infinite cost. We denote such a metric by MinEx-k. MinEx-n considers all possible simple paths without any cycles.

CORD is highly opportunistic as a source (or relay node) forwards its packets to the first meeting node whose metric satisfies the forwarding condition. Therefore, the expected delay must be computed using the expectation of the minimum expected delays possible through any neighbors of a candidate node. ExMin accounts for the increased probability of meetings when a node has many neighbors. Formally, for each node w in the neighbor set of v, we can define ExMin (v) as follows.

$$\mathrm{ExMin}(v) = \mathbb{E}\Big[\min_{w \in N_v} (I_{v,w} + \mathrm{ExMin}(w))\Big] \tag{2}$$

where the set of next-hop neighbors $N_v = \{w : w \in v\text{'s neighbors}, \mathrm{ExMin}(w) \leq \mathrm{ExMin}(v)\}$ and ExMin($\mathtt{D}$) is zero. The reason why next-hop neighbors should have less delay than v's current delay is to protect a loop of routing path. In the same way as we define MinEx-k, we can define ExMin-k to limit the computation of ExMin to the paths of k hops or less. Throughout this paper, we simply use MinEx and ExMin instead of MinEx-n and ExMin-n respectively.

4.2. Metric computation

In the computation of the routing metric in Equation (2), we essentially compute the expectation of X, where

$$X = \min_{j \in \mathcal{N}} \{I_j + m_j\},$$

for a positive integer j and a finite index set $\mathcal{N}$. Note that I_j denotes a random variable representing inter-contact time and m_j denotes a constant representing a neighbor's ExMin metric respectively. Recent studies reveal that the human-assisted devices show a truncated power-law distribution [27] whereas taxies in a city produce an exponential distribution [10]. It is reasonable to assume that the inter-contact time distribution between nodes v and w are given because it can be collected from private contact histories and is known to v and w.

Assuming that I_js' are independent (i.e., the inter-contact times are independent for two different neighbors), it is convenient to first compute the CCDF of X ($\mathbb{P}[X > x]$), and then obtain the PDF of $\mathbb{P}[X = x]$ by differentiating the CCDF. When the distribution of the inter-contact time I_j is known, using the following:

$$\begin{aligned} \mathbb{P}[X > x] &= \mathbb{P}[\min_{j \in \mathcal{N}} \{I_j + m_j\} > x] \\ &= \prod_{j \in \mathcal{N}} \mathbb{P}[I_j + m_j > x]. \end{aligned} \tag{3}$$

Then, by the definition of expectation, it is easy to compute $\mathbb{E}[X]$ either in the closed-form or numerically. In particular, if I_j is exponentially distributed, we can verify that the closed-form solution can be found in Appendix 8.

4.3. Overheads in a centralized computation

When each node computes ExMin or MinEx by itself, the iteration of (1) or (2) requires network resources for message passing, storage and computing. Since those overheads can make ExMin computation infeasible, we analyze the amount of each overhead experienced by each node.

4.3.1. Meta-data exchange

To compute (2) recursively, each node should know the information of every link in the network *e.g.*, inter-contact time distribution of all possible pairs. As each node can keep track of the contacts with its current neighbors, it can easily track its own meta-data for routing, *e.g.*, inter-contact time distribution with neighbors. However, tracking the meta-data of other links in which neighbors are not attached requires nodes to flood their meta-data, such as link-state routing protocols. Hence, when two nodes encounter each other, they exchange there combined meta-data whose size is bounded by the number of entire links in the network. To this end, the meta-data flooding additionally consumes wireless resources which is limited by the bandwidth and the contact duration.

Since the meta-data should contain the distribution information of all links in the network, the size of meta-data is $O(N^2)$ where N is the number of all nodes in the network. Assuming that the inter-contact distribution of each link is represented by one variable *e.g.*, λ in exponential distribution, the size of information may be 8 bytes. When there are 10000 nodes in the network, the size of combined meta-data exchanged by currently established nodes can be up to $8\binom{10000}{2} \approxeq 400$MBytes. Thus, as the number of nodes increases, the message passing incurs a critical overhead in wireless resources.

4.3.2. Computation complexity

When each node computes ExMin, they compute (2) iteratively for every source destination pair. The number of iteration for each S-D pair is the number of nodes N, since it is the maximum hop-count among all possible paths of the S-D pair. Hence, the computation complexity is $O(N^2 D f(N))$ where D is the number of destinations and $f(n)$ is the complexity of computing the equation (2). In Appendix 8, we show an equation (7) which is closed form of (2) in case where the inter-contact time distribution is assumed to be exponential distribution. According to our derivation, the dominant computing part in (7) is sorting neighbors' ExMin value. If we use a quick sort algorithm, $f(N)$ can be $O(NlogN)$. To this end, the complexity of centralized computation of ExMin is $O(N^3 DlogN)$ when the link distribution follows exponential distribution. Note that the centralized computation of MinEx is $O(N^3 D)$.

4.3.3. Storage

The amount of meta-data that each node maintains in its storage is same to the overhead of message passing, $O(N^2)$. Moreover, to compute ExMin recursively, each node should maintain the list which contain ExMin values of every source-destination pair whose size scales with ND. Hence, the total size of storage required for centralized ExMin is $O(N^2 + ND)$. Suppose that there are 10000 nodes and 100 destinations in the network. Then the amount of required storage is dominated by $O(N^2)$ and bounded by 400MByte.

5. Online computation of ExMin

5.1. Computing ExMin Online

Due to the high control overheads in centralized computation of ExMin, we aim at developing the online computation algorithm whose overheads are much less than centralized method. The online computation is enabled by the iterative form in the equation (2). Thus, if each node monitors its local neighbor's ExMin values and the link costs to them, the iterations distributed over time finally lead to ExMin values for the entire topology.

Then, the main issue is how to design the online computation algorithm on mobile nodes and we propose the following scheme. Nodes maintain ExMin vector (EV) whose element is ExMin value from itself to each destination and exchange their EVs with other nodes in contact. Upon receiving EVs of other nodes, they update their neighbors' EVs in their storages. Based on their neighbors' EVs, they re-compute their EVs. The detailed online algorithm is summarized as follows.

Online ExMin computation

At each slot t,

Step 1. *Probe current neighbors*
Each node exchange the probing packets with nodes in contact.

Step 2. *Update each link distribution to each neighbor.*
If there exists a node who arrives at time t, the node updates the inter-contact time distribution between itself and the encountering node.

Step 3. *Update EVs of neighbors.*
Nodes exchange the meta-data which only contains their EVs. When a node receives meta-data from other nodes in contact, it updates EVs of next-hop neighbors. The set of next-hop neighbors of node v, N_v is defined as
$\{w : w \in v\text{'s neighbors}, \text{ExMin}(w) < \text{ExMin}(v)\}$

Step 4. *Compute ExMin in distributed way.*
Based on the next-hop neighbors' EVs, each node computes the following equation and updates its EV.
$\text{ExMin}(v) = \mathbb{E}\Big[\min_{w \in N_v}(I_{v,w} + \text{ExMin}(w))\Big].$

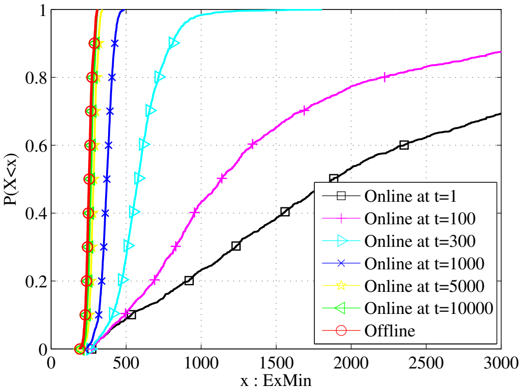


Figure 2: CDFs of online ExMin for every source-destination pair at simulation time t. The online computation starts at $t = 1$ where one-hop inter-contact times are initially assigned as metrics to each node. As t increases, the online ExMin converges to offline ExMin which is computed in centralized way.

Step 5. *Packet forwarding or replication.*
Based on the computed routing metric (ExMin), CORD replicates or forwards the packet to other nodes.

5.2. Overhead

5.2.1. Meta-data exchange

As each node should keep track of EVs of other nodes, their meta-data just contain their own EV whose size is $O(D)$. Compared to message passing overhead of centralized computation in Section 4, the online algorithm consumes much less wireless resource. Suppose that there are 10000 nodes and 100 destinations. Then, the size of meta-data is up to 800 Bytes, while the size of meta-data in centralized computation is bounded by 400MBytes.

5.2.2. Computation complexity

Since each node only computes (2) without any iteration, the computation overhead in online computation is $O(Df(n))$ which is much less than that of the centralized computation.

5.2.3. Storage

Each node should maintain neighbors' EV whose size is $O(ND)$. In case of centralized computation, this is dominated by the size of every link distribution, $O(N^2)$. Hence, the required storage is also reduced compared to the centralized computation. When there are 10000 nodes and 100 destinations, the amount of storage for each node should be up to 1MBytes, while the required storage of the centralized computation is 100MBytes.

5.3. Convergence

We denote convergence time as the time taken to have saturated (or stabilized) ExMin values for all nodes such that the values are not further updated with more iterations. In real scenarios, if the convergence of ExMin for a network is not guaranteed due to variability of link costs, ExMin looses its practicality. Hence we verify whether ExMin can be converged in Figure 2. We generate a mobile network of 100 nodes and assign a random inter-contact time for each pair of nodes. Then, we operate the online ExMin algorithm whose computation occurs whenever two nodes contact each other. The result shows that the online ExMin converges to the offline ExMin as the simulation time increases.

6. Metric Analysis

Both ExMin and MinEx as routing metrics represent the estimated delay of DTN opportunistic routing. To illustrate the difference between MinEx and ExMin in estimating the true delivery delays from a source to destination, we run a simple simulation using an exponential distribution of inter-contact times. We assign a random inter-contact time to each unique pair of nodes in the network of 300 nodes. We randomly choose 20 source and destination (S-D) pairs and measure their message delivery delay for CORD. We ignore all the queuing delays to focus on the network delays caused by inter-contact times. The experiment measures average message delays for each S-D pair when MinEx and ExMin are used for routing metrics, respectively. Figure 3 shows measured delays as well as estimated delays by MinEx and ExMin. From the figure, we can see that ExMin can track the true message delivery delays very well while MinEx shows significant difference from the true delays. This indicates that ExMin accurately estimates the opportunism exploited by the routing algorithm.

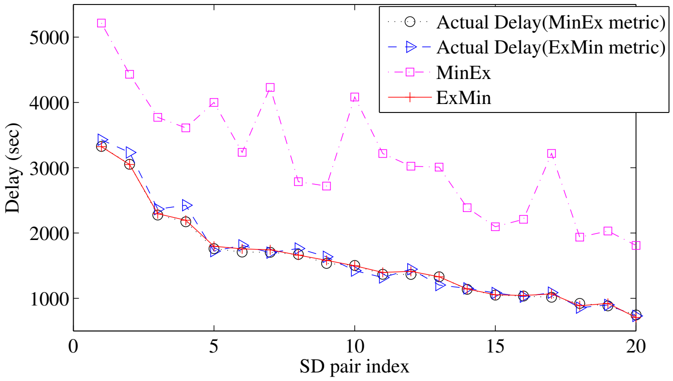


Figure 3: Message delivery delays between each S-D pair measured from a simulation experiment when the CORD routing algorithm adopts MinEx and ExMin respectively. The estimated delivery delays by MinEx and ExMin are also plotted. It shows ExMin can track the true message delivery delays of an opportunistic DTN routing algorithm (CORD).

Further, we provide a simple analysis to compare the metric values that ExMin and MinEx compute on the same graph topology. Suppose that a random inter-contact time from a source $\mathtt{S}$ to a future neighbor j is $I_j(j = 1, \cdots, n)$ and x_j is an expected delay from node j to destination at 1 hop.

Theorem 1. *For any source node* $\mathtt{S}$*, metrics computed by ExMin and MinEx have a following inequality.*

$$\textit{ExMin-}k(\mathtt{S}) \leq \textit{MinEx-}k(\mathtt{S}), \forall k \tag{4}$$

Proof. We prove theorem 1 by mathematical induction. When $k = 1$, both ExMin-1($\mathtt{S}$) and MinEx-1($\mathtt{S}$) are equal to $I_{\mathtt{D}}$, where $\mathtt{D}$ is a destination node.

Suppose the statement holds at $k = n$ and denote $\mathcal{N}_E^n(\mathtt{S})$ and $\mathcal{N}_M^n(\mathtt{S})$ by a set of node $\mathtt{S}$'s neighbors that are determined by ExMin-n and MinEx-n as follows.

$$\mathcal{N}_E^n(\mathtt{S}) = \{j|\text{ExMin-}n(j) < \text{ExMin-}n(\mathtt{S})\} \tag{5}$$

$$\mathcal{N}_M^n(\mathtt{S}) = \{j|\text{MinEx-}n(j) < \text{MinEx-}n(\mathtt{S})\} \tag{6}$$

At $k = n + 1$,

$$\text{ExMin-}(n+1)(\texttt{S}) = \mathbb{E}\Big[\min_{j\in\mathcal{N}_E^n(\texttt{S})}(I_j + \text{ExMin-}n(j))\Big]$$
$$\text{MinEx-}(n+1)(\texttt{S}) = \min_{j\in\mathcal{N}_M^n(\texttt{S})}\Big(\mathbb{E}\Big[I_j + \text{MinEx-}n(j)\Big]\Big)$$

Note that $\mathcal{N}_E^n(\texttt{S})$ and $\mathcal{N}_M^n(\texttt{S})$ are not equivalent since a neighbor node j in each set should satisfy different conditions that are ExMin-$n(j)$ < ExMin-$n(\texttt{S})$ or MinEx-$n(j)$ < MinEx-$n(\texttt{S})$.

For convenience of notation, we let $M_n(j)$ be MinEx-$n(j)$ and $E_n(j)$ be ExMin-$n(j)$. Let a be a next hop node of $\texttt{S}$ for MinEx at iteration $n+1$ (i.e., a solution of $\arg\min_{j\in\mathcal{N}_M^n(\texttt{S})}(I_j + M_n(j))$). We divide this into two cases based on whether node a is a member of a set $N_E^n(\texttt{S})$ or not.

- when a is not a member of $\mathcal{N}_E^n(\texttt{S})$, $E_{n+1}(\texttt{S}) \leq E_n(\texttt{S}) \leq E_n(a)$ by (5). Then, by the statement at $k = n$, $E_n(a) \leq M_n(a) \leq M_{n+1}(\texttt{S})$.
- when a is a member of $\mathcal{N}_E^n(\texttt{S})$, $I_a + M_n(a)$ stochastically dominates $I_a + E_n(a)$ and $I_a + E_n(a)$ stochastically dominates $\min_{j\in\mathcal{N}_E^n(\texttt{S})}(I_j + E_n(j))$. Thus, $E_{n+1}(\texttt{S}) \leq M_{n+1}(\texttt{S})$

In both cases, $E_{n+1}(\texttt{S}) \leq M_{n+1}(\texttt{S})$ and this proves the theorem. □Theorem 1 shows that ExMin always gives a lower metric value than MinEx, which is also conjectured by Figure 3. This can be reinterpreted as the true delivery delay of opportunistic routing estimated by ExMin is always less than the delivery delay along the pre-determined shortest path (MinEx).

7. Performance Evaluation

7.1. Setup

We use three different GPS traces to verify our work: Shanghai taxi traces[28] which were obtained from Wireless and Sensor networks Lab (WnSN), Shanghai Jiao Tong University, San Francisco taxi traces[29], and KAIST campus traces. In the Shanghai traces, the location information of about 4000 taxies is recorded at every 40 seconds within an area of 102 km^2 for 28 days (4 weeks). In order to present more reliable results, we selected 1486 taxies (out of all 4000 taxies) which recorded more than 70 % of GPS readings with high accuracy. The San Francisco traces obtained from tracking the GPS locations of 536 taxies in San Francisco at every 30 seconds for the total period of over 25 days. For the KAIST campus traces, we gave GPS devices to about 100 students who took an introductory computer literacy class in September 2008. The GPS devices tracked and recorded the locations of all the students simultaneously in the KAIST campus for a period of one week.

We create three different network scenarios created from the three GPS traces: Shanghai, San Francisco and KAIST. The Shanghai scenario uses the actual GPS mobility traces of 1486 taxies in the area of 30 km by 30 km; the San Francisco scenario uses the mobility traces of 536 taxies in the area of 15 km by 30 km; and the KAIST scenario uses the mobility traces of 93 students in the area of 2.5 km and 2.5 km. Each node has a WiFi radio in the ad hoc mode and taxi nodes have 300 meter radio ranges and student nodes have 50 meters. The radio can transmit at a rate of 500 Kbps which is sufficient to transmit one bundle (or packet) of 1.87 MB at every 30 seconds or 311 KB at every 5 seconds. The size of buffers are assumed to be sufficiently large so that there is no queuing loss of data.

7.2. Tested Algorithms and Metrics

We test two routing metrics. There are a number of routing algorithms in the literature with various metrics [5, 6, 7, 8, 9, 10], most of which propose a new protocol for choosing a relay. It is hard to evaluate all these algorithms. Also, our interest lies in *routing metrics,* not in the detailed algorithmic features of routing protocols. To that end, we use CORD as the generic routing algorithm. Recall that the candidates of CORD are divided into two types of DTN routing protocols: single-copy routing [5, 6, 7] and multi-copy routing [9, 10]. A nodes in single-copy routing protocols forwards their packets to the node in contact whose routing metric is better than the node and deletes the forwarded packets. In multi-copy routing protocols [9, 10], a node replicates their packets to other nodes based on maximizing the corresponding utility. The utility for each replication is determined by a resource allocation and a routing metric which can be described as how many copies of each packet are currently spread over the entire network and how long each copy takes to be delivered to the destination, respectively. Note that the new routing metric ExMin is our main contribution, while the optimal resource allocation is already studied in [9, 10]. To sum up, a single-copy protocol with MinEx can be viewed as one that represents a group of algorithms like [6], [5] and multi-copy protocols with MinEx are [10, 9, 30]. We use Max

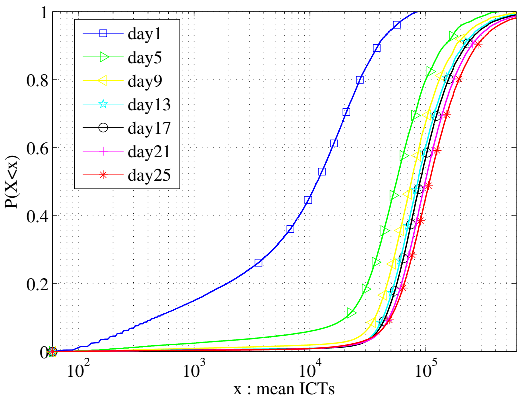


Figure 4: CDFs of average inter-contact time for every taxi pairs, which are accumulated over different amount of days.

Contribution [10] as a resource allocation in multi-copy scenario which is reported to be sub-optimal yet outperform other existing multi-copy routing protocols.

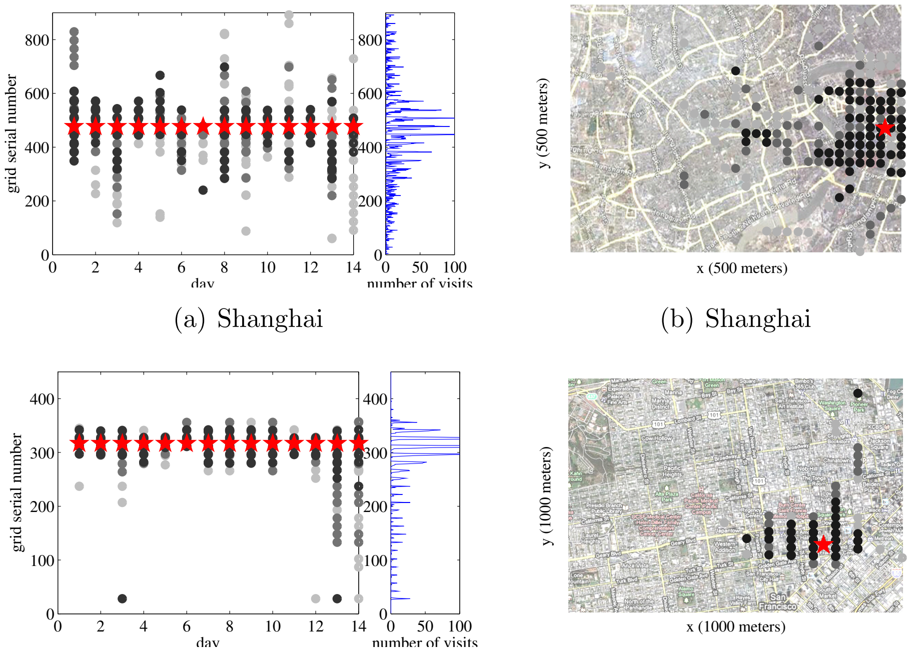
There are also many different types of routing metrics in the literature as discussed in the introduction. Depending on the design objective, the metrics are different. The examples include delay [9], probability of meeting [10] and expected remaining time for meeting another relay or destination [6]. Many of these algorithms rely on finding the shortest path to the destination. This type of routing algorithms does not take into account probabilistic meeting opportunism. Thus, they all have similar lapse as using MinEx for the routing metric. Therefore, we use MinEx as a base case for presentation of our experimental results.

Two performance metrics are considered: (i) *delivery ratio* and (ii) *delay*. The delivery ratio is the ratio of the totally delivered packets (counting only original packets) within a designated time deadline over the total number of packets that sources generate. The delay refers to the average elapsed time of all the delivered packets in all sessions.

7.3. Experimental Data Analysis

7.3.1. Regularity

People do not move randomly but move with their own schedules. Such characteristics appearing at human movements form regular patterns of visiting places or encountering other people[31, 18]. From the taxi trace in



(a) Shanghai

(b) Shanghai

(c) San Francisco

(d) San Francisco

Figure 5: (a) Locality of one particular taxi in Shanghai and (c) Locality of one particular taxi in San Francisco. We split the entire Shanghai and San Francisco city into 900 grids and 450 grids respectively. Each grid is color-coded by the average number of visits per day. Light gray: 5 visits per-day, dark gray: 6 visits, black: more than 7 visits, red star: the weighted center of the taxi. The total visits over two weeks on each grid in y-axis is plotted on the right side. (b),(d) The points in (a) and (c) are plotted on the Shanghai and San Francisco city map

shanghai, we also find regularity in the pattern of 1) encountering taxies and 2) locations each taxi visits daily. First, Figure 4 shows regularity in the pattern of encountering other taxies. In order to show the existence of such patterns, we plot CDFs of average inter-contact times whose samples are aggregated over days. For instance, the samples of CDF curve for day 17 is the average inter-contact times for every taxi pairs which are computed based on inter-contact times from day 1 to day 17. From Figure 4, we observe that the average inter-contact times become almost constant after averaging inter-contact times over more than two weeks. The curves in Figure 4 also indicate that there exist a few taxi pairs who are frequently encountering each other.

Second, Figure 5 shows another regular patterns in visiting areas for each taxi. Figure 5(a) plots the visit counts of a particular taxi over 900 grids of Shanghai and (b) plots the visit count over 450 grids of San Francisco. The color dots represent the frequency of visits per day where darker colors imply more frequent visits. We denote the weighted center of all the visited locations of that taxi by a red star. A weighted center is defined to be the geographical center position regarding the number of visits as the weight for all visited locations. The number of frequently visited grids are fairly small and all the frequently visited grids are centered around the weighted center. This trend is also clearly visible from the other traces. We omit the plots. If we define the *locality of nodes* as the grids that are visited by the same node more than twice a day, we can verify that the average number of unique grids in a locality of each node is less than 10% of the total number of grids for vehicle traces: 65 out of 900 (with variance of 35) from Shanghai, 14 out of 200 from San Francisco, and 64 out of 100 from KAIST human traces. These data indicate that nodes (taxies and students) form their locality and the size of them is relatively small.

To sum up, those regularity in human movement stabilize the inter-contact time distribution, which in turn make online ExMin computation feasible in real traces.

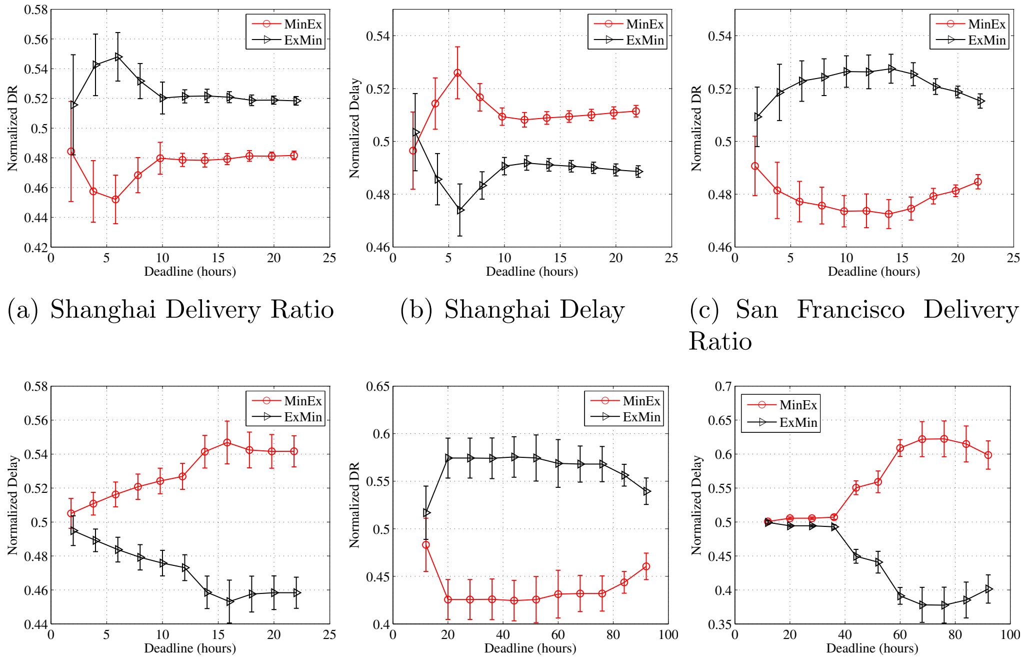
7.3.2. Randomness of inter-contact time

The inter-contact time is treated as a delay of each link in DTN. Due to the uncertainty of mobility, the link delay is assumed to follow certain stochastic distribution. Recent researches [10, 32] show that inter-contact time distribution in vehicular scenario follows exponential distribution. For human mobility trace, the distribution shows heavy-tail with exponential decays [31].

7.4. Simulation Results

7.4.1. Single copy case

For simulation, we randomly choose about 1000 S-D pair sessions for Shanghai and San Francisco and about 200 for KAIST. All the sessions start at the same time. 30 packets are injected into the network from each source in Shanghai and San Francisco taxi traces and 10 packets are injected in the KAIST student trace. Each bundle is associated with a deadline and if a bundle does not get delivered with a deadline, it is dropped. Figure 6 shows the performance of normalized delivery ratios(DR) and normalized delays of



(a) Shanghai Delivery Ratio

(b) Shanghai Delay

(c) San Francisco Delivery Ratio

(d) San Francisco Delay

(e) KAIST Delivery Ratio

(f) KAIST Delay

Figure 6: Performance of single-copy case : (a) Normalized Delivery ratio (DR) and (b) Normalized delay of two metrics for deadlines with randomly chosen 1000 S-D pairs of Shanghai taxies. The delay performance gain of ExMin is up to 20% than MinEx (c) Normalized DR and (d) Normalized delay of San Francisco trace with randomly chosen 1000 S-D pairs. The delay performance gain ExMin is up to 23% than MinEx. (e) Normalized DR and (f) Normalized delay with randomly chosen 200 S-D pairs of KAIST students. ExMin has up to 33% lower delay than MinEx.

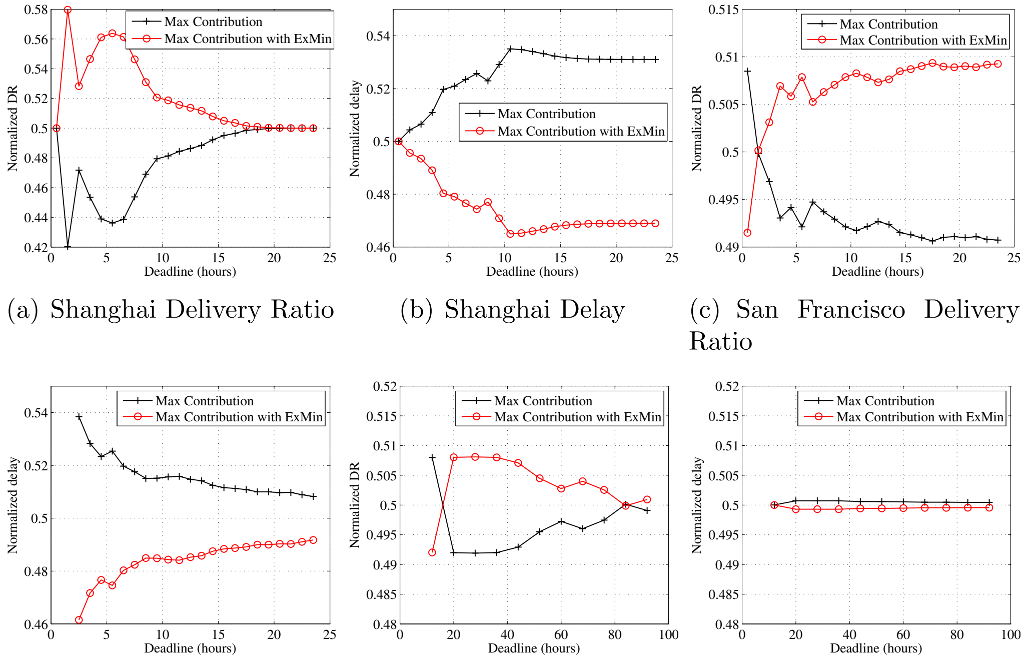
two routing metrics in the single-copy mode. For instance, a normalized DR of ExMin is calculated as $\frac{\text{(DR of ExMin)}}{\text{(DR of ExMin)+(DR of MinEx)}}$. The reason why we use a normalized gain is to clearly expose performance differences over various deadlines. We show that ExMin performs better than MinEx in all scenarios. The delivery ratio and delay of ExMin over MinEx grow up to 22% and 16%, respectively for the Shanghai traces; 12% and 18% for the San Francisco trace; and 35% and 33% for the KAIST trace. In KAIST environment where the students in the trace are all well connected as they all take the same course and scale of the campus network is small, the opportunistic routing with MinEx which computes the delay of a predetermined path to a destination fails to capture true opportunism leading to substantially lower performance.

Note that in all experiments, we use exponential ICT distributions. While this assumption holds for the vehicle traces [10, 32], it does not for human mobility traces as they are shown to follow heavy-tail with exponential decays [31]. Despite this discrepancy in the distributions of ICT used in the calculation of ExMin and in the actual traces, ExMin shows very good performance over MinEx. There could be a few explanation for this outcome. One thing is that this gain is mainly due to diverse paths available in the KAIST trace as most participants are well connected with each other and this provides more opportunities for ExMin to exploit the opportunism of diverse links. If ExMin uses more accurate ICT models, then we expect the performance gain would only increase. Another possible explanation for this behavior is that although human traces show heavy-tail tendency, they still have exponential decay at the end of the tail. Therefore, the exponential ICT distribution still provides a good approximation to the actual sample ICT distribution.

7.4.2. Multi-copy case

We also compare the performance of the routing metrics in multi-copy scenario where nodes replicate their packets to other relay nodes. In the multi-copy scenario, we randomly choose 1000 S-D pair sessions for the Shanghai and San Francisco traces and about 200 for the KAIST trace. In each session, 15 packets are injected in Shanghai and 100 packets are injected in both San Francisco and KAIST trace. Since Shanghai trace has much more number of nodes than San Franscisco trace, we intentionally injected less number of packets to Shanghai trace to obtain emulation results in a reasonable time duration. As mentioned in Section 7.2, we implemented Max Contribution [10] as a baseline algorithm. In [10], Max Contribution is extensively compared with other DTN algorithms and shown to outperform RAPID [9] and others by up to 50 %. Then, by replacing the routing metric MinEx by ExMin, we design a new version of Max Contribution with ExMin. Figure 7 shows the normalized performance metrics for each scenario.

The results in Shanghai tell us that the performance gain in delivery ratio is up to 30% when deadline is 6 hours. As the deadline increases, the gain is naturally reduced to zero since the delivery ratios of both algorithms converge to one. This shows that the substantial gain over Max Contribution can be achieved just by adding the true opportunistic gain in the routing metric. However, the performance of Max Contribution in San Francisco and KAIST traces is pretty comparable to the performance of Max Contribution



(a) Shanghai Delivery Ratio

(b) Shanghai Delay

(c) San Francisco Delivery Ratio

(d) San Francisco Delay

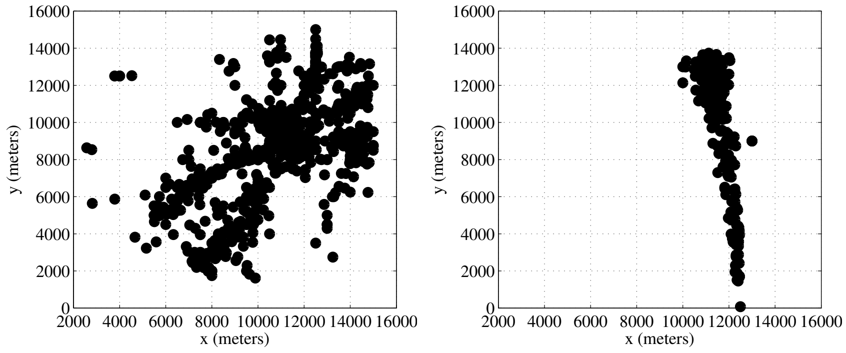
(e) KAIST Delivery Ratio

(f) KAIST Delay

Figure 7: Performance of multi-copy case : (a) Normalized Delivery ratio (DR) and (b) Normalized delay of two metrics for deadlines with randomly chosen 1000 S-D pairs of Shanghai taxies. The DR performance gain of ExMin is up to 29% than MinEx (c) Normalized DR and (d) Normalized delay of San Francisco trace with randomly chosen 1000 S-D pairs. The delay performance gain ExMin is 10% over MinEx when deadline is 5 hours. (e) Normalized DR and (f) Normalized delay with randomly chosen 200 S-D pairs of KAIST students. ExMin has up to 5% higher DR than MinEx.

with ExMin. This is because the effect of a resource allocation[2] in Max Contribution dominates the effect of opportunistic gain from the routing metric. In other words, in the San Francisco and KAIST traces, deciding the number of copies replicated by nodes is more important than estimating more precise delay for each copy's path. We can explain the phenomenon by investigating the characteristics of each trace. Figure 8 plots the weighted centers of all nodes in (a) Shanghai and (b) San Francisco, which represents the regularly visited area of each taxi. The visited areas in San Francisco

[2] As mentioned in Section 7.2, we define the resource allocation policy in multi-copy routing as controlling the number of replicated copies of a packet in the network.



(a) Shanghai W.Cs (b) San Francisco W.Cs

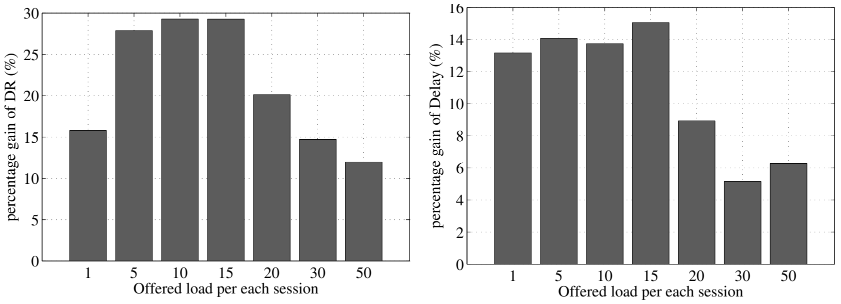
Figure 8: We plot Weighted Centers (W.Cs) which represents regularly visiting area of (a) 1400 taxies in Shanghai and (b) 600 taxies in San Francisco.

are concentrated on the downtown (top area), the airport (bottom area) and the intermediate highway road, while those in Shanghai are widely dispersed over various region. Since the San Francisco trace has much smaller hotspot areas than Shanghai, spreading copies by replications in San Francisco performs better than Shanghai. Hence, the impact of a routing metric in San Francisco is less influential under multi-copy algorithms. In KAIST trace where students share a small number of common visited places (*e.g.*, class rooms, dormitory and dining facility), we also observe the small gain due to the similar characteristic to San Francisco trace.

Figure 9 further shows the effect of the routing metric for a varying number of injected packets to Shanghai trace. At the offered load of 10∼20 per each session, Max Contribution with ExMin outperforms the original Max contribution [10] up to 29%. When the offered loads are significantly small, the network resource becomes sufficient to deliver all the loads, thus, the performance gap between MinEx and ExMin decreases. For significantly high offered load (compared to the network resource), most of the injected packets cannot take the opportunity to be replicated but just stay in the queues of each node during given deadline, in which the effect of estimating a more correct routing metric decreases.

8. Conclusions

Probabilistic links formed among mobile encounters in DTN leads challenging routing problems. So far, a popular approach is to use expected



(a) DR gain

(b) Delay gain

Figure 9: The performance gain are plotted over various number of injected packets per each session. The gain is measured by comparing the delivery ratio (DR) and Delay of 'Max Contribution with ExMin' and 'Max Contribution'. For each offered load, we pick the maximum DR and Delay gain among all possible deadlines.

delays as a routing metric to decide the next hop relay node for packet delivery to the destination, based on the minimum of the expected delays over the possible paths. This *MinEx* metric ignores the opportunity gain induced by probabilistic link delays in spite of opportunistic forwarding of selecting the relay that is first encountered. To exploit such an opportunism in the routing metric computation, we propose a new metric *ExMin* that takes the expectation of the minimum delays over multiple hops, statistically. Then, we propose the online computation algorithm of ExMin to reduce the overheads of centralized computation. Our extensive simulation results over real traces show that ExMin achieves much better performance than MinEx in both single-copy and multi-copy scenarios.

In this paper, we discuss the proposed metrics, ExMin in the context of exponential inter-contact time distributions. Although our two taxi traces show exponential ICT distributions, several studies on human mobility indicate that the ICT distribution of people including our KAIST student traces has a shape of a power-law head and an exponential tail (much like truncated power-law). Clearly, according to our results on KAIST traces, the definition of our metrics does not preclude use of them under different ICT distributions. We believe that ExMin is widely beneficial in general DTN environment.

Appendix

ExMin computation for exponential inter-contact time

The computation of $\mathbb{E}[X]$, where $X = \min_{j \in \mathcal{J}}\{I_j + m_j\}$, for $I_j \sim \exp(\lambda_j)$ and some constant m_j, for a finite set $\mathcal{J}$,$\mathcal{J} = \{1 \ldots J\}$.When $m_1 = \ldots = m_J$, the computation is easy since it suffices to compute $\min_j I_j$ which is known to be exponentially distributed with the intensity $\sum_j \lambda_j$.

When m_js are not equal, without loss of generality, we assume that $m_1 \leq \ldots \leq m_J$. For a given j, from (3), we have:

$$\mathbb{P}[X > x] = \begin{cases} 1 & \text{if } 0 < x < m_1 \\ \prod_{i=1}^{j} \mathbb{P}[I_i + m_i > x] & \text{if } m_j < x < m_{j+1} \\ \prod_{i=1}^{J} \mathbb{P}[I_i + m_i > x] & \text{if } m_J < x < \infty, \end{cases}$$

where we divide the cases of x to consider when $x - m_i < 0$ (i.e., $\mathbb{P}[I_i > x - m_i] = 1$ for any i. Then, it follows that

$$\mathbb{P}[X = x] = \begin{cases} 0 & \text{if } 0 < x < m_1 \\ \bar{\lambda}_j \exp\left(-\bar{\lambda}_j x + \bar{m}_j\right) & \text{if } m_j < x < m_{j+1} \\ \bar{\lambda}_J \exp\left(-\bar{\lambda}_J x + \bar{m}_J\right) & \text{if } m_J < x < \infty, \end{cases}$$

where

$$\bar{\lambda}_j = \sum_{k=1}^{j} \lambda_k, \qquad \bar{m}_j = \sum_{k=1}^{j} \lambda_k m_k.$$

Then, by definition of expectation, we get:

$$\begin{aligned} \mathbb{E}[X] &= \int_0^\infty x P(X = x) dx \\ &= \sum_{j=1}^{J-1} \int_{m_j}^{m_{j+1}} \overline{\lambda}_j \cdot x \cdot e^{-\overline{\lambda}_j x + \overline{m}_j} dx \\ &\quad + \int_{m_J}^{\infty} \overline{\lambda}_J \cdot x \cdot e^{-\overline{\lambda}_J \cdot x + \overline{m}_J} dx \\ &= \sum_{j=1}^{J-1} \{ e^{-m_j \overline{\lambda}_j + \overline{m}_j} (m_j + \frac{1}{\overline{\lambda}_j}) - e^{-m_{j+1} \overline{\lambda}_j + \overline{m}_j} \\ &\quad \cdot (m_{j+1} + \frac{1}{\overline{\lambda}_j}) \} + e^{-m_J \overline{\lambda}_J + \overline{m}_J} (m_J + \frac{1}{\overline{\lambda}_J}) \qquad (7) \end{aligned}$$

References

[1] S. Biswas, R. Morris, Exor: Opportunistic multi-hop routing for wireless networks, in: Proceedings of ACM SIGCOMM, 2005.

[2] S. Chachulski, M. Jennings, S. Katti, D. Katabi, MORE: A network coding approach to opportunistic routing, in: MIT-CSAIL-TR-2006-049, 2006.

[3] E. Rozner, J. Seshadri, Y. A. Mehta, L. Qiu, SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks, IEEE Transactions on Mobile Computing 8 (2009) 1622–1635. doi:http://doi.ieeecomputersociety.org/10.1109/TMC.2009.82.

[4] Z. Zhong, S. Nelakuditi, On the efficacy of opportunistic routing, in: Proceedings of IEEE SECON, 2007.

[5] K. F. S Jain, R. Patra, Routing in a delay tolerant network, in: Proceedings of ACM SIGCOMM, 2004.

[6] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Efficient routing in intermittently connected mobile networks: The single-copy case, IEEE/ACM Transactions on Networking 16 (1) (2008) 63–76.

[7] E. P. C. Jones, L. Li, P. A. S. Ward, Practical routing in delay-tolerant networks, in: Proceedings of ACM SIGCOMM workshop on Delay-tolerant networking, 2005.

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

[9] A. Balasubramanian, B. N. Levine, A. Venkataramani, DTN routing as a resource allocation problem, in: Proceedings of SIGCOMM, 2007.

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

[11] D. P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific, Belmont, Massachusetts, 2005.

[12] A. Vahdat, D. Becker, Epidemic routing for partially-connected ad hoc networks, technical Report, CS-200006, Duke University, April 2000.

[13] R. Ramanathan, R. Hansen, P. Basu, R. R. Hain, R. Krishnan, Prioritized epidemic routing for opportunistic networks, in: Proceedings of ACM MobiSys workshop on Mobile Opportunistic Networks, 2007.

[14] J. Lebrun, C.-N. Chuah, D. Ghosal, M. Zhang, Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks, in: Proceedings IEEE Vehicular Technology Conference, 2004.

[15] S. Basagni, I. Chlamtac, V. Syrotiuk, B. Woodward, A distance routing effect algorithm for mobility (dream), in: Proceedings of ACM MOBICOM, 1998.

[16] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, ACM SIGMOBILE Mobile Computing and Communications Review 7 (3) (2003) 19–20.

[17] R. Shah, S. Roy, S. Jain, W. Brunette, Data mules: Modeling a three-tier architecture for sparse sensor networks, in: Proceedings First IEEE Intl Workshop Sensor Network Protocols and Applications (SNPA 03), 2003.

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

[19] J. Leguay, T. Friedman, V. Conan, Evaluating mobility pattern space routing for dtns, in: Proceedings IEEE INFOCOM, 2006.

[20] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, Bubble rap: Social-based forwarding in delay tolerant networksn, in: Proceedings ACM Mobihoc, 2008.

[21] M. E. Daly, M. Haahr, Social network analysis for information flow in disconnected delay-tolerant manets, IEEE Transactions on Mobile Computing 8 (5) (2009) 606–621.

[22] C. Liu, J. Wu, An optimal probabilistic forwarding protocolin delay tolerant networks, in: Proceedings of ACM MOBIHOC, 2009.

[23] E. Altman, G. Neglia, F. D. Pellegrini, D. Miorandi, Decentralized stochastic control of delay tolerant networks, in: Proceedings of IEEE INFOCOM, 2009.

[24] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, in: Proceedings First Intl Workshop Service Assurance with Partial and Intermittent Resources (SAPIR 04), 2004.

[25] J.Widmer, J.-Y. L. Boudec, Network coding for efficient communication in extreme networks, in: Proceedings of ACMSIGCOMM Workshop on Delay Tolerant Networking (WDTN), 2005.

[26] T. Spyropoulos, K. Psounis, C. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, in: Proceedings of ACM SIGCOMM, 2005.

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

[28] S. J. U. Traffic Information Grid Team, Grid Computing Center, Shanghai taxi trace data, http://wirelesslab.sjtu.edu.cn/.

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

[30] V. Erramilli, M. Crovella, A. Chaintreau, C. Diot, Delegation forwarding, in: Proceedings of ACM MobiHoc, 2008.

[31] I. Rhee, M. Shin, S. Hong, K. Lee, S. Chong, On the levy walk nature of human mobility, in: Proceedings of INFOCOM, 2008.

[32] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, L. M. Ni, Recognizing exponential inter-contact time in vanets, in: Proceedings of IEEE INFOCOM mini conference, 2010.