Multicent: A Multifunctional Incentive Scheme Adaptive to Diverse Performance Objectives for DTN Routing

Kang Chen and Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC, USA 29631
Email: {kangc, shenh}@clemson.edu

***Abstract*—Due to intermittent connection and limited communication opportunity, routing in Delay Tolerant Networks (DTNs) is usually conducted in a store-carry-forward manner. Consequently, different packet forwarding or storage strategies can lead to different performance objectives, such as minimal average delay, maximal hit rate, and minimal maximal delay, which are desired by applications with different purposes. However, nodes may not be willing to follow these strategies. Further, selfish nodes may even refuse to carry or forward packets for others if they cannot obtain benefits in return. Though many incentive systems have been proposed to encourage packet forwarding, none of them aim to encourage nodes to realize the aforementioned performance objectives. In this paper, we first discuss the strategies that can realize different performance objectives and then propose Multicent, a game theoretical incentive scheme that not only provides cooperative incentives but also encourages nodes to follow defined rules to realize the desired performance objective. Multicent also makes the Quality of Service (QoS) of packet routing adjustable for specific sources, destinations, or source-destination pairs. Extensive trace-driven experimental results verify the effectiveness of Multicent.**

I. Introduction

In recent years, the usage of mobile devices has increased rapidly. Thus, Delay-Tolerant Networks (DTNs) [1] comprised of mobile devices (e.g., laptops and smart phones) have attracted considerable research interest. In a DTN, nodes are assumed to experience frequent network partitioning, and no stable routing path can be assured between any source-destination pairs. Due to these characteristics, routing algorithms in DTNs [2]–[9] usually work in a store-carry-forward manner and only a limited number of packets can be transmitted when two nodes meet. Therefore, the forwarding sequence and storage priority of a packet on a node determine the dissemination speed of the packet and the final successful delivery rate and delay, which provides the possibility to realize different performance objectives, such as minimal average delay, maximal hit rate, or minimal maximal delay.

Different performance objectives are desired by different application scenarios. For example, in a DTN based environment monitoring system, the dissemination of control messages usually require maximal hit rate while the report of collected disaster data needs minimal delay. The work in [6], [9] has proposed packet forwarding and storage rules/sequences to realize different performance objectives. However, they all assume that nodes are cooperative and follow these rules to forward and store packets, which may not be true in DTNs. Nodes may be selfish and are not willing to carry or forward packets for others. Second, nodes may not necessarily to follow the strategies to realize the desired performance objective. For instance, in a DTN consisted of nodes from different entities, some nodes may not give priorities to important control messages from another nodes but only be willing to put equal importance to all nodes. This phenomenon is more evident when a DTN is composed of nodes owned by different organizations. Therefore, an incentive scheme is needed to encourage nodes to follow these rules in order to realize different performance objectives.

Recently, a number of DTN incentive schemes [10]–[13] have been proposed. These works mainly focus on ways to reward packet forwarders so that nodes are encouraged to be cooperative in DTN routing. Most of the schemes build an off-line virtual bank (OVB) to be responsible for credit clearance. Each packet forwarder imprints its ID to the packet and the OVB determines the amount of credits for forwarders based on their contribution by examining arrived packets. However, the cooperation in these methods only means that nodes are willing to receive, store and forward packets for others. We name this as the first aspect of cooperation. They fail to consider how to further encourage nodes to collaborate and follow certain rules in order to achieve different performance objectives as mentioned in previous paragraph. Thus, we claim that cooperation has the second aspects that nodes are willing to follow the strategies that can lead to the designed routing performance objective.

In this paper, we propose a game theoretical incentive scheme called Multicent for DTN routing to achieve both of the two aspects of cooperation. We assume nodes are selfish and rational in nature, and they participate in packet forwarding and storage to maximize their payoff. We regard the packet forwarding between two nodes as a game in which each node adopts the strategy that can maximize its remuneration. Then, we propose Multicent with a payoff function for the game, so when nodes follow their nature to choose the best strategies for themselves, the two aspects of cooperation are simultaneously realized. Multicent can also adjust the QoS (i.e., delay and hit rate) for specific sources, destinations, or source-destination pairs by adjusting the payoff function. In summary, the contributions of this paper are threefold:

- First, we identify the two aspects of cooperation in DTN routing, which should be considered simultaneously to realize a specific performance objective.
- Second, while current schemes only consider the first aspect of cooperation, we propose a game theoretical incentive scheme that can realize the two aspects of cooperation in multi-copy DTN routing algorithms.
- Third, we further propose ways to realize adjustable QoS

for packet routing from, to and among specific sources, destinations, and source-destination pairs.

The remainder of this paper is arranged as follows. Related works are presented in Section II. Section III introduces the network model and design principles. Section IV presents the detailed system design of Multicent. In Section V, the performance of Multicent is evaluated through trace-driven experiments. Section VI concludes this paper.

II. Related Work

A. Incentive Schemes for MANETs

Many incentive schemes have been proposed for MANETs, which can be regarded as a dense scenario of DTNs. These methods can generally be classified as either reputation-based schemes [14]–[17] or credit-based schemes [18]–[22]. In reputation-based schemes, nodes usually adopt neighborhood monitoring or overhearing to detect misbehaving nodes and calculate the reputation of their neighbors. The nodes disseminate reputation information to other nodes, so the selfish nodes can be excluded from the network. However, such techniques are not suitable for the DTN environment in which monitoring and reputation dissemination tend to be extremely difficult due to node sparsity and high node mobility.

In the credit-based schemes, nodes earn credits by forwarding packets for others and pay for the forwarding service offered by others. The SIP protocol [18] lets each node imprint its forwarding behavior into the packet to calculate its remuneration. iPass [19] introduces an auction game between service providers (forwarding nodes) and consumers (forwarding requesters) so that each node would bid honestly according to the actual amount of system resource (i.e., bandwidth) needed. Mahmoud and Shen [20] combined the reputation and incentive schemes to achieve fairness by rewarding credits to cooperative nodes. The works in [21], [22] integrate game theory into the credit-based scheme to model the packet forwarding process and provide effective incentive schemes.

However, these credit-based schemes cannot be directly applied to DTNs since most of them need a contemporaneous end-to-end path between two nodes, which can hardly be found in DTNs due to opportunistic node encountering.

B. Incentive Schemes for DTNs

Research on incentive schemes for DTNs has emerged in recent years [10]–[13]. SMART [13] is a secure credit-based incentive scheme in DTNs with the notion of a *layered coin*. In this scheme, each node adds one layer, which includes its ID and authentication information, to the transferred packet. Then the destination node reports which node has forwarded the packet to a center so that the remuneration can be distributed correctly even without a contemporary path. Also, SMART uses a *layer concatenation technique* to prevent intermediate nodes from modifying the attached layers. PI [12] aims to build a fair and practical incentive scheme for DTNs. Besides rewarding nodes on the successful path with remuneration, it also increases reputation values to forwarders on the failed path to recognize their contribution. When the reputation of a node is low, others refuse to forward its packets despite the change to obtain credits. The work in [11] builds a distributed incentive system for DTNs, which requires that each pair of nodes provides a close amount of forwarding service to each other. However, this method does not suit the DTN environment because some nodes may connect more nodes than others and take more forwarding responsibility [10]. Mobicent [10] deliberately designs a payoff function to prevents nodes from earning more credits by inserting connections (edge insertion attack) or hiding connections (edge deletion attack).

Though these methods are effective, they only aim to realize the first aspect of cooperation introduced in the introduction. They cannot motivate nodes to forward packets in different sequences to achieve different performance objectives.

III. Modeling and Design Principles

A. Network and Node Model

We regard a DTN as consisting of K mobile nodes denoted by N_i ($i = 1, 2, 3, \cdots, K$). We refer to the period of time during which two nodes can communicate with each other as a communication session. In DTNs, nodes meet with each opportunistically, and the communication session length often is limited. Therefore, only a limited number of packets can be forwarded in one communication session. Also, each node has limited storage space. In this paper, we assume nodes are selfish but do not have malicious behaviors (i.e., collusion, cheating, *etc.*). Due to page limit, we leave how to effectively prevent malicious behaviors as our next research topic.

For simplicity, we assume that each packet has a fixed size. Packets with various lengths can be divided into a number of same-size segments. According to the number of copies for a packet, DTN routing algorithms can be divided into single-copy routing [2]–[4] or multi-copy routing [5]–[9]. Though single-copy routing has low resource consumption, it is less reliable due to the opportunistic characteristic of DTNs [6]. Therefore, we focus on multi-copy routing in this paper, which means when a node forwards a packet to another node, it still keeps the packet, i.e., the packet is replicated to the other node. We then interchangeably use forwarding and replication with the same meaning in this paper.

B. Performance Objectives and Design Principles

1) Performance Objectives: In this work, we consider four objectives: 1) *Minimal average delay with TTL (Time to Live)*; 2) *Maximal hit rate with TTL*; 3) *Minimal average delay without TTL*; and 4) *Minimal maximal delay without TTL*. In the above definition, with and without TTL refers to whether each packet is configured with a TTL. The first objective minimizes the average delay of successfully delivered packets when TTL is configured for each packet. The second objective maximizes the percentage of successfully delivered packets when TTL is configured for each packet, which is the common objective in DTN routing algorithms. The latter two objectives minimize the average delay and the maximal delay of all successfully delivered packets when there is no TTL, respectively. The reason that we have two objectives with TTL and two objectives without TTL is to show that the proposed method is applicable to objectives with different configurations.

2) General Principles: The work in [6] designs the strategies for different performance objectives under the assumption that nodes are naturally cooperative. We first introduce the strategies in this work, and then present the goal of Multicent: encouraging nodes to follow these strategies.

In detail, each packet is associated with a utility value, denoted U_k, that is positively related to the desired performance objective. Then, when the packet that causes more increase on the utility is forwarded first and the packet with the higher utility is stored with the higher priority, each packet forwarding and storage operation enhances the desired routing objective for the most. Finally, the overall routing objective is maximized [6]. In summary, two encountered nodes need to follow two rules for packet forwarding and storage in below:

R1: packets in two nodes are forwarded in descending order of their increase in utility;

R2: the packet with the least utility value is replaced if the storage is full when a new packet arrives.

The goal of Multicent is to provide incentives so that only when nodes are cooperative in forwarding packet (first aspect of cooperation) and also strictly follow the above two rules (second aspect of cooperation), they can earn the most remuneration. As a result, the contribution of each packet forwarding/storage to the specific objective is maximized, thereby realizing the desired performance objective.

3) Utility Calculation: We then discuss how to calculate the utility. For a packet k, we use D_k to denote its estimated total delivery delay, T_{l_k} to denote the time it has lived, and T_{r_k} to represent the remaining time needed for delivery. Then, $Pr\{T_{r_k} < TTL - T_{l_k}\}$ represents the probability that packet k is successfully delivered within TTL (Time To Live) in DTN routing. Then the utility value (U_k) for each performance object is calculated as below:

- *Minimal average delay with TTL*: $U_k = -D_k / Pr\{T_{r_k} < TTL - T_{l_k}\}$.
- *Maximal hit rate with TTL*: $U_k = Pr\{T_{r_k} < TTL - T_{l_k}\}$.
- *Minimal average delay without TTL*: $U_k = -D_k$.
- *Minimal maximal delay without TTL*:

$$U_k = \begin{cases} -D_k & \text{if } D_k \geq D_i \text{ for all } i \in \mathcal{M} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{M}$ represents all messages in a node. Note that we use a negative utility to ensure that a smaller maximal delay leads to a larger utility value. Then, maximizing the utility value is still the desired objective.

We further introduce how to calculate D_k and $Pr\{T_{r_k} < TTL - T_{l_k}\}$. As indicated in [6], the modeling of meeting times in DTN is very difficult even with mixture models. Therefore, we adopt the same assumption in the paper that the separation time between two nodes, say N_i and N_j, follows the exponential distribution with mean value $1/\lambda_{ij}$. The separation time means the period of time between two consecutive encountering. Although this is a rough assumption and may cause inaccurate estimation, we adopt it since it makes delay estimation feasible and performs well in practice. Both our experiment and [6] validate such an assumption. We also assume that the packet size is small so that a packet only needs to meet the destination node once to be delivered.

We first look at the single-copy DTN routing. Suppose there is a packet k on node N_i for node N_j. Then, the probability that the remaining time needed (d_{ij}) to deliver packet k to N_j is less than T can be expressed by

$$Pr_{ij}(T) = Pr(d_{ij} < T) = 1 - e^{-\lambda_{ij} T} \quad (1)$$

Further, as aforementioned, the average separation time between N_i and N_j is $1/\lambda_{ij}$. Considering the memorylessness of the exponential distribution, the average remaining time ($\bar{d}_{ij}$) for packet k to meet N_j is

$$\bar{d}_{ij} = 1/\lambda_{ij} \quad (2)$$

However, in multiple-copy DTN routing, there usually exist several copies for the same packet in different nodes. As a result, the actual remaining time needed to deliver packet k should be the time needed by the first copy that arrives at the destination. That is, suppose there are three nodes holding a copy of packet k: a, b, and c, the time needed to deliver packet k (d'_{kj}) is

$$d'_{kj} = min\{d_{aj}, d_{bj}, d_{cj}\} \quad (3)$$

where $d_{xj}, x \in a, b, c$ represents the expected time needed for packet k's holder x to meet its destination. Similarly, the probability that the remaining delivery time of packet k is less than T ($P'_{kj}(T)$) is 1 minus the probability that all nodes holding the copy fail to deliver the packet within T:

$$P'_{kj}(T) = 1 - (1 - Pr_{aj}(T))(1 - Pr_{bj}(T))(1 - Pr_{cj}(T)) \quad (4)$$

where $Pr_{xj}(T)$ is the probability that the remaining time for node x to meet node j and is calculated by Formula 1. Then, we have the expected delivery delay ($\tilde{D}_k$) and the probability of successful delivery ($\tilde{Pr}(T_{r_k} < TTL - T_{l_k})$) as

$$\tilde{D}_k = T_{l_k} + d'_{kj}. \quad (5)$$

$$\tilde{Pr}(T_{r_k} < TTL - T_{l_k}) = P'_{kj}(TTL - T_{l_k}) \quad (6)$$

Although above calculation matches the real situation, it 1) complicates the estimation steps and 2) cannot realize the calculation in a distributed manner. Therefore, we followed the heuristic method in [6] to only consider current holder's expected delay and delivery probability. Such a simplification is reasonable since enhancing the delivery probability of one replica also enhances that of all replicas, and the replication of packets to nodes with high utility values actually reflects the relaying. Then the D_k and $Pr\{T_{r_k} < TTL - T_{l_k}\}$ for packet k on node N_i is calculated as

$$D_k = T_{l_k} + \bar{d}_{ij} = T_{l_k} + 1/\lambda_{ij} \quad (7)$$

$$Pr(T_{r_k} < TTL - T_{l_k}) = 1 - e^{-\lambda_{ij}(TTL - T_{l_k})} \quad (8)$$

The λ represents the encountering frequency between two nodes. It is updated whenever two nodes meet with each other. When a packet is forwarded to another node N_i, its utility value U_k is updated according to the λ_{ij} between N_i and the packet's destination N_j.

IV. System Design

In this section, we first model the packet forwarding process between two nodes as a game. Then, we introduce the detailed design of Multicent.

A. Packet Forwarding Game

When two nodes meet, each node can forward several packets to the other or accept/reject received packets during the communication session. Such a process can be regarded as a packet forwarding game consisting of a number of rounds of interactions between two nodes. In each interaction, both nodes select their forwarding or receiving strategy. The number of rounds is determined by the length of the communication session. Longer sessions lead to more rounds. We first analyze the remuneration of a node when it selects different strategies in the forwarding game. Based on the analysis of the node activities in the game, we design Multicent in the next section.

Suppose node N_i meets another node N_j ($i, j \in [1, K]$ and $i \neq j$) in a DTN and there are m_i and m_j packets in the two nodes, respectively. We use $P_{i1}, P_{i2}, P_{i3}, \cdots, P_{im_i}$ to represent packets in N_i and $P_{j1}, P_{j2}, P_{j3}, \cdots, P_{jm_j}$ to represent the packets in N_j. Then, in each interaction, each node selects one strategy from its strategy set. For a packet P_{ia} ($a \in [1, m_i]$) in N_i, we use S_{ia} to represent N_i's behavior of *sending/forwarding packet* P_{ia} and use R_{ia} to represent N_i's behavior of *receiving the forwarded packet and replacing packet* P_{ia}. We use R_{i0} to represent N_i's behavior of *storing a forwarded packet directly* without replacing any packets. NS means *no forwarding* and NR means *not accepting the forwarded packet*. The notations apply to S_{jb} and R_{jb} ($b \in [1, m_j]$) similarly. Then, the strategy sets of the two nodes denoted by ($\{\vec{S_i}, \vec{R_i}\}$ and $\{\vec{S_j}, \vec{R_j}\}$) are

$$\vec{S_i} = \{S_{i1}, S_{i2}, S_{i3}, \cdots, S_{im_i}, NS\}, \tag{9}$$

$$\vec{R_i} = \{R_{i0}, R_{i1}, R_{i2}, R_{i3}, \cdots, R_{im_i}, NR\}, \tag{10}$$

$$\vec{S_j} = \{S_{j1}, S_{j2}, S_{j3}, \cdots, S_{jm_j}, NS\}, \tag{11}$$

$$\vec{R_j} = \{R_{j0}, R_{j1}, R_{j2}, R_{j3}, \cdots, R_{jm_j}, NR\}. \tag{12}$$

Note that if the storage on a node is full, the node cannot choose R_{i0} (or R_{j0}). Let s_i (or s_j) represent the selected forwarding strategy of node N_i (or N_j) other than NS and let r_i (or r_j) represent the selected receiving strategy of node N_i (or N_j) other than NR. Then, considering no reward will be assigned if the packet is not forwarded successfully, the remuneration matrix for one round of the packet forwarding game is shown in Table I.

TABLE I: One round of the packet forwarding game

$N_i \backslash N_j$	s_j	r_j	NS	NR
s_i	$\{-C_f, -C_f\}$	$\vec{O}_{ij}(s_i, r_j)$	$\{-C_f, 0\}$	$\{-C_f, -C_r\}$
r_i	$\vec{O}_{ij}(r_i, s_j)$	–	–	–
NS	$\{0, -C_f\}$	–	$\{0, 0\}$	$\{0, 0\}$
NR	$\{-C_r, -C_f\}$	–	$\{0, 0\}$	$\{0, 0\}$

In Table I, $\vec{O}_{ij} = \{O_i, O_j\}$ representing the benefits for N_i and N_j after considering the reward from the incentive system. C_f and C_r are the unit cost of forwarding and receiving a packet, respectively. From table I, we have following observations on the forwarding game:

- Two nodes can possibly earn profit only when one forwards a packet while the other accepts the packet. Otherwise, they only waste resources.
- Let $\{s_m, r_n\}$ represent the pair of sending and receiving strategy that results in maximal O_i and O_j, denoted O_i^* and O_j^*, where $m, n \in \{i, j\}$ and $m \neq n$. Then, if $O_i^* > 0$ and $O_j^* > 0$, $\{s_m, r_n\}$ can maximally benefit both N_i and N_j, the corresponding action would be taken.
- Otherwise, $\{s_m, r_n\}$ can only benefit one or none of N_i and N_j, so no action will be taken.

B. Game Theoretical Incentive Scheme

With the understanding from the packet forwarding game, we follow below steps to design the incentive scheme so that when nodes are rationale and always seek to maximize their benefits, they forward and store packets in the sequence that can achieve different performance objectives.

- First, to achieve the specified performance objective, the packet among the two nodes that brings about the most benefit should be sent first. Then, we need to encourage both nodes to be cooperative in deciding the sequence of sending packets between them.
- Second, we need to encourage nodes to forward and store packets following *R1* and *R2* for different performance objectives, as described in Section III-B2.

Below, we introduce how Multicent realize the two goals through the design of its payoff functions.

How to choose packet sender? To encourage both meeting nodes to be cooperative in determining the sequence of packet forwarding between them so that the packet that brings about the most benefit (highest utility increase) has the highest priority to be forwarded, Multicent splits the reward for each forwarding evenly to both packet sender and the receiver. Consequently, the node holding the packet that can bring about the most benefit first sends the packet and the other node receives it. Then, the packet with the second highest benefit in both nodes is sent, and so on.

How to achieve *R1*? When two nodes meet, transferring the packet that causes the highest utility increase among the packets in both nodes can contribute the most to the performance objective. However, this cannot be achieved in previous incentive schemes, which only reward a node's forwarding behavior. To realize *R1*, Multicent rewards the forwarding of a packet in proportion to its utility increase after the forwarding, encouraging nodes to first forward the packet that can bring about the largest utility increase.

How to achieve *R2*? When a node with full storage receives a packet, discarding the packet with the lowest utility can contribute the most to the performance objective. However, this cannot be achieved in previous schemes since they do not specifically reward storing behavior. To achieve *R2*, Multicent rewards a node that still stores a packet when it is successfully delivered to the destination or its TTL is expired. The reward amount is in proportion to the packet's utility. As a result, if a node wants to earn more credits from storing a packet, it would store packets with larger utility values.

Further consideration. Not all packets can be delivered to their destinations in DTN routing. Then, should nodes that have forwarded these unsuccessful packets be rewarded? In Multicent, we still reward these nodes. This is to ensure that when a node decides the forwarding or storing priority

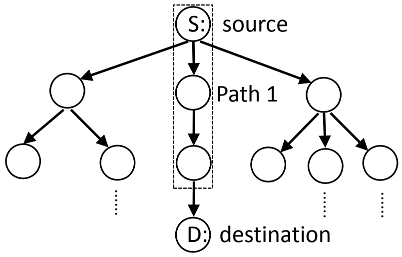


Fig. 1: Packet replication tree.

of a packet, it does not need to consider the probability of successful delivery but only the designed utility value, as required by aforementioned strategies.

Payoff functions. With all above design principles, we summarize the payoff functions in Multicent. To better demonstrate the payoff functions, we formalize the routing process of a packet in multi-copy routing algorithms as a tree structure, as shown in Figure 1. In the figure, each arrowed line represents one replication. Each node that contains the packet further replicates it until learning that it has been delivered to the destination or it expires due to TTL. In the figure, an arrow means a successful replication, and a path is the route that a packet has traversed excluding the destination node. A path may or may not connect to a destination. The latter occurs when a packet is expired or is replaced due to storage limit.

Then, the payoff functions in Multicent include:

P1: All paths for a packet are paid with credits.

P2: For a path, each arrow is paid with credits C_{s_i}, which is proportional to its contribution to the increase of the utility value,

$$C_{s_i} = \mathbb{F}_s(\triangle U_i) = \alpha \triangle U_i \ (\alpha > 0). \tag{13}$$

where $\triangle U_i$ is the increase of the utility value and is calculated as the new utility value decreases the previous utility value. Credits C_{s_i} are evenly divided between two nodes connected by the arrow.

P3: Each node on a path that holds the packet until it expires due to TTL or is successfully delivered to the destination is assigned an additional amount of credits C_{r_i} ($C_r \ll C_s/2$), which is proportional with the packet's utility value on the node,

$$C_{r_i} = \mathbb{F}_r(U_i) = \beta U_i \ (\beta > 0); \tag{14}$$

P4: If a node forwards the packet to its destination, a fixed amount of credit C_d ($C_s/2 \ll C_d$) is rewarded.

Let $[f_{s1}, f_{s2}]$ and $[f_{r1}, f_{r2}]$ represents the ranges of the results calculated by $\mathbb{F}_s(\bullet)$ and $\mathbb{F}_r(\bullet)$, respectively. Since the incentive scheme requires that $C_r \ll C_s/2 \ll C_d$ (P3 and P4), we confine that $f_{r2} < f_{s1}/2$ and $C_d > f_{s2}/2$. The credits are assigned to nodes by a central server based on reports of the packet forwarding and storage records. We explain this process later in Section IV-D .

When two nodes meet, we first let two nodes exchange the IDs of packets that have been successfully delivered to their destinations, as in multi-copy DTN routing algorithms [6]. Then, the packet delivery information is disseminated in the network quickly, and nodes can know which packets should be stopped for replication and storing, thereby avoiding wasting resources on delivered packets in the network. After this, they further exchange their meeting records to decide the utility of each packet on the other node. Then, following the rationale to earn as many credits as possible, the packets are forwarded and stored in the decreasing order of benefits, realizing to the two aspects of cooperation. We prove this in the next section.

C. Incentive Scheme Validation

In this section, we show how Multicent's payoff functions achieve the two aspects of cooperation through analysis.

1) The First Aspect of Cooperation (Forwarding Packets): In our design, each forwarding is recognized, even when the forwarded packets eventually fail to reach the destination. In multi-copy routing, packets are replicated rather than transferred to another node. Then, forwarding a packet does not affect the current node's future opportunity to earn credits from this packet since it still keeps the packet. Moreover, the additional credit C_d encourages nodes to delivery packets to their destinations upon encountering them. Therefore, in order to maximize its profit, a node will be cooperative at every opportunity to forward a packet to other nodes or its destination. This means that nodes would cooperate rather than strive for a communication opportunity, realizing the first aspect of cooperation.

2) The Second Aspect of Cooperation (Realizing Performance Objectives): Suppose node N_i meets node N_j and strategy pair (S_{ia}, R_{jb}) is selected. Note S_{ia} and R_{jb} can be any strategy in the corresponding sending and receiving strategy set, as introduced in Section IV-A. Then, based on Equation (13) and P2, the benefits for N_i and N_j, denoted by O_{is} and O_{js}, for forwarding packet P_{ia} are

$$\begin{cases} O_{is} = \mathbb{F}_s(\triangle U_a)/2 \\ O_{js} = \mathbb{F}_s(\triangle U_a)/2. \end{cases} \tag{15}$$

Also, based on Equation (14), the benefit for N_i and N_j resulting from strategy R_{jb} (including R_{j0}) is

$$\begin{cases} O_{ir} = 0 \\ O_{jr} = \mathbb{F}_r(U_a) - \mathbb{F}_r(U_b) \end{cases} \tag{16}$$

where $\mathbb{F}_r(U_b)$ is the loss of benefit by discarding P_{jb}.

Satisfying requirement *R1*. Based on Formula (15), we can see that each node takes the packet in its memory with the largest utility increase as the forward candidate. Let P_{ia} and $P_{jb'}$ represent the packets with the maximal utility increase in N_i and N_j, respectively, and $P_{ia'}$ and P_{jb} be the packets with the minimal utility value in node i and node j, respectively. Then, combining Formula (15) and (16), the remuneration for the two nodes when P_{ia} or $P_{jb'}$ is forwarded can be represented as Formulas (17) and (18), respectively.

$$\begin{cases} O_i(s_i, r_j) = \mathbb{F}_s(\triangle U_a)/2 \\ O_j(s_i, r_j) = \mathbb{F}_s(\triangle U_a)/2 + \mathbb{F}_r(U_a) - \mathbb{F}_r(U_b) \end{cases} \tag{17}$$

$$\begin{cases} O_i(r_i, s_j) = \mathbb{F}_s(\triangle U_{b'})/2 + \mathbb{F}_r(U_{b'}) - \mathbb{F}_r(U_{a'}) \\ O_j(r_i, s_j) = \mathbb{F}_s(\triangle U_{b'})/2 \end{cases} \tag{18}$$

Without loss of generality, we assume that $\triangle U_a$ is larger than $\triangle U_{b'}$. Then we can see that N_j would choose to let N_i

send P_{ia} since $O_j(s_i, r_j)$ is larger than $O_j(r_i, s_j)$. For N_i, recall that C_r is much lower than $C_s/2$, which means $\mathbb{F}_s()$ dominates the benefit for N_i. As a result, $O_i(s_i, r_j)$ is usually larger than $O_i(r_i, s_j)$, and N_i would choose to send P_{ia} first in most cases. However, if $O_i(s_i, r_j)$ is less than $O_i(r_i, s_j)$, N_i would wait for N_j to send $P_{jb'}$. In this case, both nodes are waiting for the other to send a packet, which would waste the communication session and lead to no benefit for them. Therefore, we augment Multicent with an additional policy. In the policy, when two nodes find that they are waiting for the other to send a packet, they would cooperate to choose the packet with a larger utility increase to be forwarded. In conclusion, packet with the highest utility increase will be forwarded first, satisfying the first requirement (*R1*).

Satisfying requirement R2. N_j's storage reward $O_{jr} = \mathbb{F}_r(U_a) - \mathbb{F}_r(U_b)$. We see that the value of $\mathbb{F}_r(U_a)$ is fixed since the forwarded packet is determined in the forwarding stage. O_{jr} is maximized if $\mathbb{F}_r(U_b)$ is minimized. Thus, the best strategy for the receiver N_j is to replace the packet that has the least $\mathbb{F}_r(U)$ with the newly arrived packet if its storage is used up. As a result, for each newly arrived packet, the lowest-utility packet is replaced when the storage is full. In conclusion, the second requirement (*R2*) is satisfied.

With the above analysis, we see that the remuneration strategy introduced in Section IV-B makes *R1* and *R2* the Nash equilibrium in for the two nodes, i.e., no one can earn more credits by deviating from the strategy. Such a result demonstrates that even selfish nodes would follow the designed scheme. Note that though we only mention four performance objectives in the paper, Multicent actually can motivate nodes to realize any performance objective with a defined utility function, including the strategy to realize equal forwarding opportunity among packets.

D. Credit Clearance

As other incentive systems in DTNs [10], [13], [21], we assume that there exists an OVB that stores the credits of all nodes and is responsible for credit clearance. Since our focus is the incentive scheme, we use a simple OVB structure that nodes submit reports to the OVB for credit clearance and obtain rewarding parameters (i.e., α, β, and performance objective) when meeting it. We leave the design of a more effective OVB as our future work. Below, we discuss the credit clearance for packet forwarding and storing separately.

Forwarding. In order to realize reward for packet forwarding, each intermediate node in a path imprints its unique identity and its contribution to the forwarding ($\triangle U_i$) into the packet during routing. Then, the last node on a path forms a report indicating the contributions of all forwarders on the path and send the report to the OVB. This policy follows the payoff functions P1 and P2. The report of the last node in the path for a successfully delivered packet should be signed by the destination node, which enables the OVB to reward C_d to the last node. This policy follows payoff function P4.

Storing. We reward nodes that hold packets until them no longer need to be stored. Specifically, every node creates a report when it finds that a packet in it is expired due to TTL or has been delivered to the destination. The node is also required to send the packet along with the report to another node, which signs the report if it finds the report is valid. Such a strategy prevents a node from earning credits by creating fake reports. Then, the report is sent to the OVB for credit assignment. This policy follows payoff function P3.

All credit clearance reports are stored in nodes until they can establish connections to the OVB. The OVB then updates each node's credit account based on the collected reports. As the works in [10], [12], [13], a certain amount of credits are charged from the destination node for the forwarding and storing services and are paid to all forwarders in the routing. When the amount of credits in a node's account is lower than 0, it means that the node is possibly a "free rider". When "free riders" are detected by the OVB, it first transfers such information to the first N nodes it meets after the detection. Then, similar to the packet delivery information, two encountered nodes also exchange the IDs of all deficit nodes they have already known. Such a design can disseminate IDs of malicious nodes to all nodes quickly and meanwhile alleviate OVB's load to inform all nodes. These malicious nodes will be excluded from the system. In this way, Multicent motivates nodes to follow its rules to earn credits for their future packets, which finally leads to the two aspects of cooperation.

E. Supporting Adjustable QoS

By adjustable QoS, we mean that the priorities of packets initiated from certain sources, targeted to certain destinations or forwarded between certain source-destination pairs should be enhanced or reduced in routing. As mentioned previously, packets are forwarded or stored with priority according to their potential to bring about benefits. Thus, we can increase or decrease the rate when calculating the benefits for packets with adjusted QoS. Specifically, the QoS adjustment for certain sources, destinations, or source-destination pairs should be first authorized by the OVB. The OVB then informs all nodes about the adjusted rate for these nodes or pairs. As a result, the expected amount of credit calculated for forwarding or storing their packets is increased or decreased. Consequently, these packets are given enhanced or reduced priority during forwarding and storage. In summary, with the adjustment of the remuneration function, packets generated for the adjustment objective (specific sources, destinations or source-destination pairs) can be forwarded or stored with enhanced or reduced priority, thereby attaining adjustable QoS.

V. Performance Evaluation

A. Experiment Settings

We evaluated Multicent through trace-driven tests with datasets from the MIT Reality project [23] (97 nodes) and the Haggle project [24] (98 nodes). In the test, the mean disconnection time (λ) is measured and updated whenever two nodes meet. During the test, there were no queries in the first 1/3 of the two datasets, which enables each node to accumulate encounter records. After this, 5000-25000 packets were generated evenly. The size of each packet was set to 1 KB and each node has 100 KB storage. We mainly measured two metrics: average delay and hit rate. The former refers to the average delay of all arrived packets while the latter refers to the percentage of successfully delivered packets. We also measured the maximal delay, which is the maximal delay of all successfully delivered packets. We set α and β in Equation (13) and (14) to 1. We adopted 95% confidence interval in experiment.

We first validate the effectiveness of Multicent in comparison to Mobicent [10] and RAPID [6]. Mobicent provides the same amount of reward to each forwarding action but neglects the impact of different forwarding sequences on the system performance. RAPID studies the impact of different utilities on different system performances but does not provide an incentive scheme. Multicent, Mobicent, and RAPID provide three levels of incentive for cooperative DTN routing: both aspects of cooperation, only the first aspect of cooperation, and no incentive. Then, we evaluate the ability of Multicent in supporting different performance metrics and adjustable QoS.

B. Performance Comparison

To make the results comparable, we set Multicent to maximum hit rate (MaxHitRate) since Mobicent and RAPID on focus on hit rate. We set 10% of nodes as selfish nodes that help forward or store packets only when they can benefit from it, and others as cooperative nodes. In Mobicent, cooperative nodes follow the two aspects of cooperation and selfish nodes only follow the first aspect of cooperation. In Multicent, both cooperative and selfish nodes follow the two aspects of cooperation. In RAPID, cooperative nodes follow the two aspects of cooperation while selfish nodes drop all packets.

Figure 2(a) and Figure 2(b) illustrate the hit rates and average delays of the three methods, respectively, with different total numbers of packets using the Haggle project dataset. We see from the two figures that the hit rate follows Multicent>Mobicent>RAPID while the average delay follows Multicent<Mobicent<RAPID. Such results indicate that without a cooperation incentive scheme, 10% of non-cooperative nodes can greatly degrade the routing performance. Also, Mobicent and Multicent achieve improved performance.

When nodes are non-cooperative, they refuse to forward packets for others, thereby wasting some forwarding opportunities. Hence, some packets may not be forwarded through the optimal forwarder. They then may not be delivered in time or even be dropped, leading to a low hit rate and a high average delay. In Mobicent, selfish nodes cooperate and help forward packets for others, thus leading to a higher hit rate and a lower average delay than RAPID. However, Mobicent only focuses on the first aspect of cooperation. By focusing on both aspects, Multicent takes full advantage of forwarding opportunities and gives higher priority to packets that can bring about greater improvement to hit rate in forwarding and storing, thus generating best performance.

Figure 2(c) and Figure 2(d) show the hit rates and average delays of the three methods, respectively, with different total numbers of packets using the MIT Reality project dataset. We observe similar results as in Figure 2(a) and Figure 2(b) for the same reasons. This confirms our conclusion that in DTN routing, incentives are necessary, and the two aspects of cooperation can result in enhanced performance.

C. Supporting Different Performance Metrics

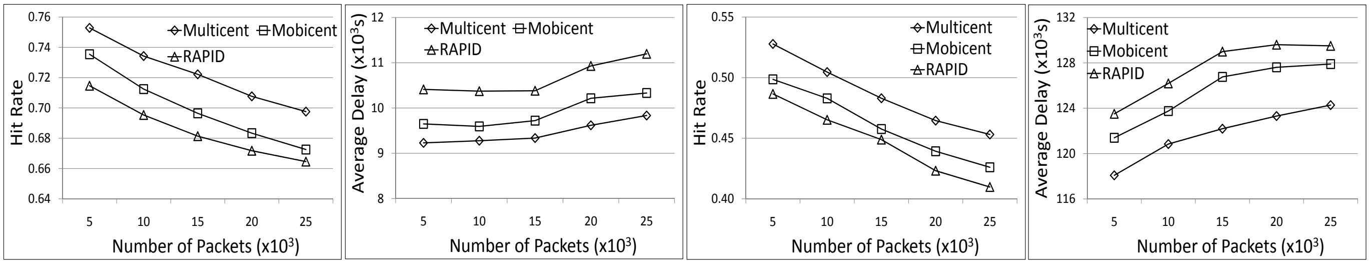
In this section, we examine the ability of Multicent to optimize performance as measured by different objectives. We tested the routing performance with the four previously proposed objectives: minimal average delay with TTL, maximal hit rate, minimal average delay without TTL, and minimal maximal delay. We denote the four modes as MinDelayW, MaxHitRate, MinDelayWo and MinMaxDelay, respectively. In order to demonstrate the effectiveness of Multicent in encouraging nodes to target on different performance objectives, we also present the results of Random for reference. In Random, all nodes only follow the first aspect of cooperation, and forward and store packets in random sequences. We tested with both the Haggle dataset and the MIT Reality dataset.

1) Hit Rate: Figure 3(a) and Figure 4(a) show the hit rates of the four modes with the two datasets, respectively. We see that the hit rate follows MinDelayWo> MinMaxDelay>MaxHitRate>MinDelayW>Random in both figures. Moreover, the hit rates of MinDelayWo and MinMaxDelay are clearly larger than those of MaxHitRate and MinDelayW. This is because there is no TTL configuration in the two modes. In the two modes with TTL, we find that MaxHitRate has higher hit rate than MinDelayW. Such results demonstrate the effectiveness of Multicent in achieving high performance for a specified objective, i.e., MaxHitRate.

In the two methods without TTL, we find that the hit rate of MinDelayWo is much higher than that of MinMaxDelay. In MinDelayWo, nodes are motivated to first forward packets that can result in the maximal decrease in the estimated delay. In other words, the routing aims to reduce the delay of all packets, resulting in higher hit rates. In MinMaxDelay, packets with larger estimated delays are forwarded first, which results in more unsuccessful deliveries and leads to a lower hit rate.

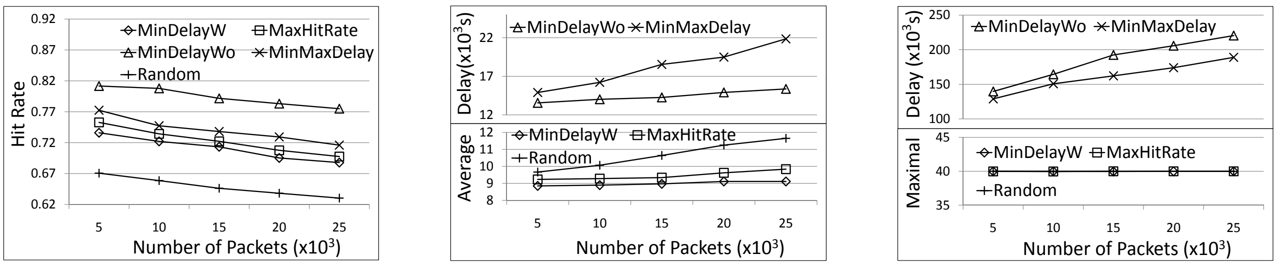
We find that Random shows the lowest hit rates in both figures. This confirms the effectiveness of Multicent in realizing different performance objectives. It also shows that the routing efficiency is not deteriorated by imposing different forwarding and storing priorities on packets. The above results demonstrate the superiority of Multicent over Random.

2) Average Delay: Figure 3(b) and Figure 4(b) illustrate the average delay under the four modes with the two datasets, respectively. Note the vertical axis (y-axis) is split in the two figures to better demonstrate the differences. The modes with TTL generate much lower average delay than those without TTL. We discuss the results separately based on whether the TTL is configured. We observe that the average delay of MinDelayW is lower than that of MaxHitRate. This justifies that MinDelayW is effective in minimizing delay under our incentive scheme. Combining the results in the previous section, we find that MinDelayW achieves a low average delay at the cost of a low hit rate while MaxHitRate leads to a higher hit rate but a larger average delay. The result confirms that Multicent is effective in achieving desired metric. The forwarding opportunities in a DTN are limited. Multicent motivates nodes to decide packet forwarding priorities by providing rewards based on utilities to optimize the performance measured by a selected metric. In the two methods without TTL, we also see that the average delay of MinDelayWo is



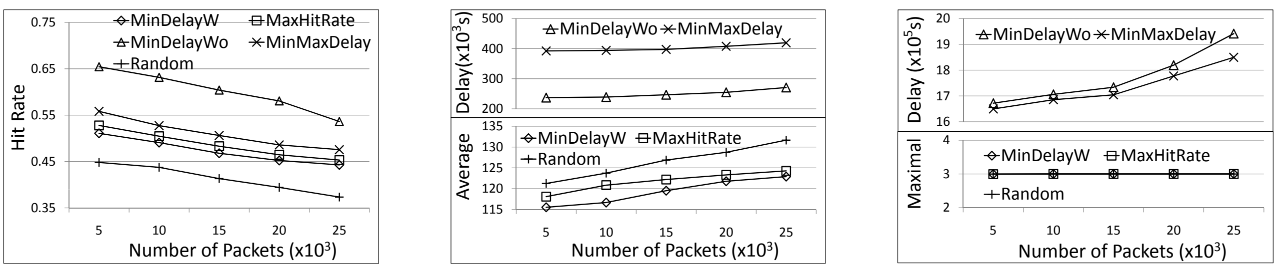
(a) Hit rates with Haggle dataset. (b) Average delay with Haggle dataset. (c) Hit rate with MIT Reality dataset. (d) Ave. delay with MIT Reality dataset.

Fig. 2: Performance comparison between Multicent, Mobicent and RAPID.



(a) Hit rate. (b) Average delay. (c) Maximal Delay.

Fig. 3: Different performance metrics with Haggle data set.



(a) Hit rate. (b) Average Delay. (c) Maximal Delay.

Fig. 4: Different performance metrics with MIT Reality data set.

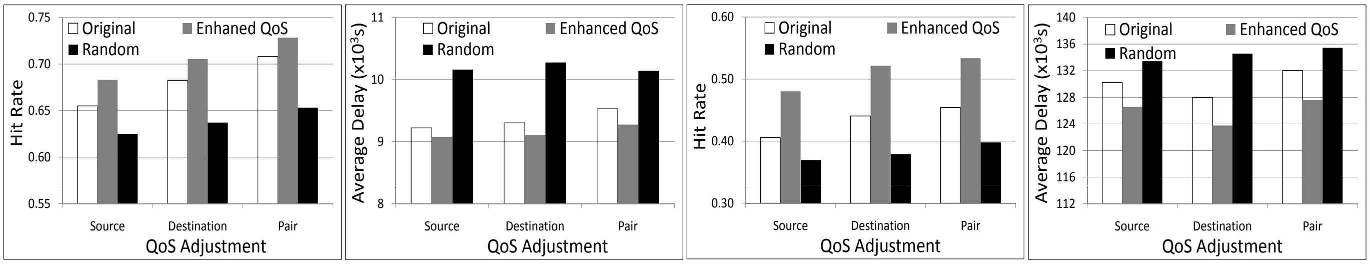
smaller than that of MinMaxDelay. This result proves that the designed metric of minimal delay is realized in Multicent. We also find that although MinDelayW and MaxHitRate have different rewarding strategies, they have much smaller average delay than Random. This result confirms the effectiveness of Multicent in enhancing system performance by considering forwarding and storing priorities for different packets.

3) Maximal Delay: Figure 3(c) and Figure 4(c) plot the maximal delay under the four modes with the two datasets, respectively. Note the vertical axis (y-axis) is also split to better demonstrate the differences. We see that MinDelay and MaxHitRate produce a similar maximal delay, and the maximal delays are around 40,000 seconds and 300,000 seconds with the two datasets, respectively. We note that the maximal delay of Random remains roughly the same as MinDelay and MaxHitRate. This is because TTL is configured in the three modes, which limits the the maximal possible delay to TTL. We also observe that MinMaxDelay has lower maximal delay than MinDelayWo. The result shows that the MinMaxDelay mode realizes its goal under our incentive scheme.

D. Supporting Adjustable QoS

In this section, we verify Multicent's ability to support adjustable QoS for packets from specific sources, to specific destinations, or for specific source-destination pairs. We name the three QoS adjustment options *Source*, *Destination* and *Pair*, respectively. Since both performance enhancement and degradation work with the same principle (i.e., increase or decrease the utility), we only show the former in the paper. In the *Source* and *Destination* modes, we randomly picked 10 nodes as the enhancement objectives, and in the *Pair* mode, 100 source-destination pairs were selected as enhancement objectives. The α and β of forwarding or storing the packets generated by these nodes (*Source* mode), destined to these nodes (Destination mode), or for these pairs (*Pair* mode) were increased by 150%. We set the total number of packets to a medium value of 15000. We also include the results of Random for reference.

Figure 5(a) and Figure 5(b) depict the hit rates and average delays under different enhancement objectives, respectively, using the Haggle project data set. In the two figures, "Original" refers to the scenario without QoS enhancement. We see from the two figures that when the corresponding enhancement mode (i.e., *Source*, *Destination*, or *Pair*) is used, the hit rate is increased and the average delay is decreased. This justifies that the desired enhancement is realized. We also see that the observed objects generate the lowest hit rates in both datasets.



(a) Hit rates with Haggle dataset. (b) Average Delay with Haggle dataset. (c) Hit rates with MIT Reality dataset. (d) Ave. delay with MIT Reality dataset.

Fig. 5: Results with different QoS enhancement strategies.

This result shows that the routing performance of the observed object is low when no enhancement is imposed by Multicent. It verifies the effectiveness of Multicent in achieving a specific performance objective, and shows that QoS adjustment can further enhance its effectiveness.

Figure 5(c) and Figure 5(d) show the results of different enhancement modes as in Figure 5(a) and Figure 5(b) using the MIT Reality project data set. We can easily observe results similar to those from the Haggle project dataset. The delay is decreased by around 3% and the hit rate is increased by 18%. The results confirm that the proposed Multicent is capable of providing adjustable QoS for packets from specific sources, to specific destinations, or for specific source-destination pairs. Moreover, we find that in both datasets, Random generates the largest average delays. This result further justifies the effectiveness of Multicent and the QoS enhancement by showing that they indeed improve the performance of the observed objects.

VI. Conclusion

In DTNs, communication opportunities between nodes are usually limited, and the packet forwarding or storage priority affects final routing performance. Thus, we first identify the two aspects of cooperation for DTN routing to realize different performance objectives: nodes should not only participate in packet forwarding but also forward or store packets as desired by a performance metric (e.g., minimal average delay, maximal hit rate, or minimal maximal delay). To this end, we proposed Multicent, an incentive scheme for DTN routing that encourages nodes to cooperate and can realize different performance objectives and adjustable QoS for packets of specific sources, destinations, or source-destination pairs. Trace-driven experimental results verify the correctness and effectiveness of Multicent in comparison with other schemes. In the future, we plan to enhance the capability of Multicent to thwart more advanced attacks such as Denial of Service and collusion.

Acknowledgements

This research was supported in part by U.S. NSF grants CNS-1254006, CNS-1249603, CNS-1049947, CNS-1156875, CNS-0917056 and CNS-1057530, CNS-1025652, CNS-0938189, CSR-2008826, CSR-2008827, Microsoft Research Faculty Fellowship 8300751, and U.S. Department of Energy's Oak Ridge National Laboratory including the Extreme Scale Systems Center located at ORNL and DoD 4000111689.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. of SIGCOMM*, 2003, pp. 27–34.
[2] F. Li and J. Wu, "Mops: Providing content-based service in disruption-tolerant networks," in *Proc. of ICDCS*, 2009, pp. 526–533.
[3] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *Proc. of MobiHoc*, 2008, pp. 241–250.
[4] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proc. of MobiHoc*, 2007.
[5] A. Vahdat and D. Becker, "Epidemic routing for partiallyconnected ad hoc networks," Duke University, Tech. Rep., 2000.
[6] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem." in *Proc. of SIGCOMM*, 2007.
[7] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks." in *Proc. of INFOCOM*, 2006.
[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. of WDTN*, 2005.
[9] K. Lee, Y. Yi, J. Jeong, H. Won, I. Rhee, and S. Chong, "Max-Contribution: On optimal resource allocation in delay tolerant networks." in *Proc. of INFOCOM*, 2010.
[10] B. B. Chen and M. C. Chan, "MobiCent: a credit-based incentive system for disruption tolerant network." in *Proc. of INFOCOM*, 2010.
[11] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang, "Incentive-aware routing in DTNs," in *Proc. of ICNP*, 2008.
[12] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss, "Pi: A practical incentive protocol for delay tolerant networks." *IEEE Trans. Wirel. Comm.*, 2010.
[13] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A secure multilayer credit-based incentive scheme for delay-tolerant networks." *IEEE Trans. Veh. Tech.*, vol. 58, no. 8, pp. 4628–4639, 2009.
[14] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks." in *Proc. of MOBICOM*, 2000.
[15] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks." in *Proc. of CMS*, 2002.
[16] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol." in *Proc. of MobiHoc*, 2002.
[17] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. of WCNC*, 2004.
[18] Y. Zhang, W. Lou, W. Liu, and Y. Fang, "A secure incentive protocol for mobile ad hoc networks." *Wireless Networks*, 2007.
[19] K. Chen and K. Nahrstedt, "iPass: An incentive compatible auction scheme to enable packet forwarding service in MANET." in *Proc. of ICDCS*, 2004.
[20] M. E. Mahmoud and X. Shen, "Stimulating cooperation in multi-hop wireless networks using cheating detection system." in *Proc. of INFOCOM*, 2010.
[21] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks." in *Proc. of INFOCOM*, 2003.
[22] S. Zhong, E. L. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques." in *Proc. of MOBICOM*, 2005.
[23] N. Eagle, A. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," *PNAS*, vol. 106, no. 36, 2009.
[24] A. Chaintreau, P. Hui, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Impact of human mobility on opportunistic forwarding algorithms," in *Proc. of INFOCOM*, 2006.