This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

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

Kang Chen, *Student Member, IEEE*, Haiying Shen*, *Senior Member, IEEE, Li Yan*

Abstract—In Delay Tolerant Networks (DTNs), nodes meet opportunistically and exchange packets only when they meet with each other. Therefore, routing is usually conducted in a store-carry-forward manner to exploit the scarce communication opportunities. As a result, different packet routing strategies, i.e., which packet to be forwarded or stored with priority, can lead to different routing performance objectives, such as minimal average delay and maximal hit rate. On the other hand, incentive systems are necessary for DTNs since nodes may be selfish and may not be cooperative on packet forwarding/storage. However, current incentive systems for DTNs mainly focus on encouraging nodes to participate in packet forwarding/storage but fail to further encourage nodes to follow a certain packet routing strategy to realize a routing performance objective. We name the former as the first aspect of cooperation and the latter as the second aspect of cooperation in DTN routing. Therefore, in this paper, we first discuss the routing strategy that can realize different performance objectives when nodes are fully cooperative, i.e., are willing to follow both aspects of cooperation. We then propose Multicent, a game theoretical incentive scheme that can encourage nodes to follow the two aspects of cooperation even when they are selfish. Basically, Multicent assigns credits for packet forwarding/storage in proportional to the priorities specified in the routing strategy. Multicent also supports adjustable Quality of Service (QoS) for packet routing between specific sources and destinations. Extensive trace-driven experimental results verify the effectiveness of Multicent.

Index Terms—Incentive system, Performance objectives, Routing, Delay tolerant networks

1 Introduction

With the increasing popularity of mobile devices, Delay-Tolerant Networks (DTNs) [1] comprised of mobile devices, e.g., laptops and smart phones, have attracted considerable research interests recently. In a DTN, nodes move continuously and can only communicate with nearby nodes due to the limited communication range. Therefore, no stable routing path can be assured between any nodes, and only a limited number of packets can be transmitted when two nodes meet. Due to these characteristics, DTN routing algorithms [2]–[9] work in a store-carry-forward manner, i.e., a packet is stored on current node until a better forwarder is encountered. Therefore, the forwarding and storage priority of a packet determines its dissemination speed. This provides the possibility to realize different performance objectives for DTN routing. A routing performance objective means to improve a specific routing metric for the most, such as maximal hit rate and minimal average delay.

Different routing performance objectives are desired by different application scenarios. For example, in a DTN based monitoring system, the dissemination of control messages usually requires maximal hit rate while the report of collected disaster data needs minimal delay. Actually, the works in [7], [9] have proposed the routing strategy that can realize different routing performance objectives. In this paper, we define a *routing strategy* as a set of rules to decide the priority of each packet in forwarding/storage. However, these methods simply assume that nodes are willing to follow the strategy to forward and store packets, which may not be true in DTNs. Firstly, nodes may be selfish and do not want to carry or forward packets for others [10], [11]. Further, nodes are not necessarily to follow the priority specified in a routing strategy to forward and store packets. For instance, some nodes may not give priority to important control packets but only give equal importance to all packets. Therefore, an incentive scheme is needed to encourage not only the cooperation on forwarding and storing packets but also the willingness to follow a routing strategy to realize a desired performance objective. We name the former as the first aspect of cooperation and the latter as the second aspect cooperation.

Recently, a number of incentive schemes [12]–[18] have been proposed for DTNs. They mainly focus on rewarding packet forwarders so that nodes are encouraged to be cooperative in DTN routing. Most of these schemes build an off-line virtual bank (OVB) for credit clearance. During the packet forwarding, each node imprints its ID into the packet it just forwards. Then, the OVB can determine credit remuneration for forwarders based on their contributions stored in packets. Though effective, the cooperation in these methods only refers to the receive, storage, and forwarding of packets. In other words, they cannot encourage nodes to follow a specific routing strategy to store and forward packets to realize a performance objective, i.e., cannot encourage nodes to realize the second aspect of cooperation.

In this paper, we propose a game theoretical incen-

- * Corresponding Author. Email: shenh@clemson.edu; Phone: (864) 656 5931; Fax: (864) 656 5910.
- The authors are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634. E-mail: {kangc, shenh,lyan}@clemson.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

tive scheme called Multicent for DTN routing that can encourage nodes to achieve both aspects of cooperation mentioned above. We assume nodes are selfish and rational in nature, i.e., they participate in packet forwarding and storage to maximize their benefits. In this paper, we first summarize the routing strategy that can realize different routing performance objects when nodes are fully cooperative, i.e,. willing to follow the two aspects of cooperation. We then design Multicent to encourage nodes to follow the two aspects of cooperation even when they are selfish. In detail, we regard the packet exchange between two nodes as a packet forwarding game. Based on the analysis of the packet forwarding game, we design a payoff function for the game that assigns credits for packet forwarding/storage in proportional to the priorities specified in the corresponding routing strategy. Note that we use *payoff function* to represent the component in an incentive system that determines credit reward. As a result, when each node follows its interest to choose packets to forward/store in the packet forwarding game, the two aspects of cooperation are simultaneously attained. Multicent can also adjust the Quality of Service (QoS), i.e., delay and hit rate, for packets with specific sources/destinations or between specific source-destination pairs by adjusting the payoff function for these packets.

The contributions of this paper are threefold:

- First, we identify the two aspects of cooperation that are needed to realize a specific performance objective in DTN routing.
- Second, while current methods only encourage the first aspect of cooperation among nodes, we propose a game theoretical incentive scheme that can encourage nodes to realize the two aspects of cooperation in DTN routing simultaneously.
- Third, we propose a way to realize adjustable QoS for packet 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 2. Section 3 introduces the background on routing performance objectives. Section 4 presents the system design of Multicent. In Section 5, the performance of Multicent is evaluated through trace-driven experiments. Section 6 concludes this paper with remarks on future work.

2 Related Work

2.1 DTN Routing Algorithms

DTN routing algorithms can be classified into either single-copy methods [2]–[5] or multi-copy methods [6]–[9]. In single-copy methods, each packet only has one copy. These methods usually rank each node's probability of encountering the destination node and forward a packet from low rank nodes to high rank nodes so that the packet can gradually reach the destination. In multi-copy methods [6]–[9], packers are replicated, rather than forwarded, to the encountered node, thereby leading to better routing reliability.

The works in [7] and [9] further discuss the packet routing strategy that can achieve different routing performance objectives, e.g., minimal average delay and maximal hit rate. However, they focus on packet routing and assume all nodes are fully cooperative, which may be true in reality. In this paper, we study how to provide incentives so that even when nodes are selfish, they will still follow the packet routing strategy to realize a routing performance objective.

2.2 Incentive Schemes for MANETs

Many incentive schemes have been proposed for Mobile Ad hoc Networks (MANETs), which is similar to DTNs but has denser node distribution. These methods can be classified as either reputation-based schemes [19]–[22] or credit-based schemes [23]–[26]. In reputation-based schemes, nodes usually adopt neighborhood monitoring or overhearing to calculate the reputation of their neighbors and detect misbehaving nodes. Nodes also disseminate reputation information to other nodes to exclude selfish nodes. However, such techniques are not suitable for DTNs in which neighbor monitoring and reputation dissemination are extremely difficult due to sparse node distribution and high node mobility.

In the credit-based schemes, nodes pay for the forwarding service offered by others and earn credits by forwarding packets for others. iPass [23] introduces an auction game between forwarding nodes and forwarding requesters so that nodes bid honestly based on actual bandwidth needs. Mahmoud and Shen [24] combined the reputation and incentive schemes to achieve fairness by rewarding credits to cooperative nodes. The works in [25], [26] integrate game theory into the credit-based scheme to model the packet forwarding process and provide effective incentive schemes. Though effective, these 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 sparse node distribution.

2.3 Incentive Schemes for DTNs

Research on incentive schemes for DTNs has emerged in recent years [10]–[18]. The works in [10] and [11] investigate the influence of selfish nodes on the routing performance in DTNs, which justifies the necessity of incentive systems for DTN routing.

SMART [12] is a secure credit-based incentive scheme in DTNs. 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 nodes have forwarded the packet to a center for remuneration calculation. PI [13] aims to build a fair and practical incentive scheme for DTNs. Besides rewarding nodes on successful paths, it also increases the reputation values of forwarders on failed paths to recognize their contribution. The work in [14] builds a distributed incentive system for DTNs, which requires

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

each pair of nodes to provide a close amount of forwarding services to each other. Mobicent [15] deliberately designs a payoff function to prevent nodes from earning more credits by inserting or hiding transaction connections. Give2Get [16] proposes a test phase after each forwarding to check whether the relay node has dropped the packet to prevent selfishness in DTNs. The work in [17] designs an incentive system that incorporates both credit based stimulation and node interests to encourage cooperation among nodes. MobiID [18] allows each node to maintain its reputation evidence to demonstrate its reputation. A node's cooperative packet transmission is demonstrated by the previous/next hop node or its community members.

Though these methods are effective on encouraging cooperation, they only aim to realize the first aspect of cooperation introduced in the introduction. They cannot motivate nodes to forward/store packets following a routing strategy to achieve a performance objective.

3 BACKGROUND

In this section, we first introduce the network modeling and routing performance objectives considered in this paper. We then present the routing strategy that can realize different performance objectives when nodes are fully cooperative. Later, in Section 4, we introduce how to encourage nodes to follow the strategy even when they are selfish, which is the main goal of Multicent.

3.1 Network Model

We consider a DTN consisting of K mobile nodes denoted by N_i ($i = 1, 2, 3, \cdots, K$). We define the period of time during which two nodes can communicate with each other as a communication session. In DTNs, nodes meet opportunistically, and the length of a communication session often is limited. Therefore, only a limited number of packets can be forwarded in one communication session. Also, each node has limited storage space. We first assume nodes are selfish but do not have malicious attacks, e.g., collusion and cheating. We discuss how to prevent some attacks in Appendix A.2.

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 [3]–[5] or multi-copy routing [6]–[9]. Though single-copy routing has low resource consumption, it is less reliable due to the opportunistic characteristic of DTNs [7]. Therefore, we first 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 also discuss how to extend Multicent to single-copy routing in Appendix A.3.

3.2 Performance Objectives

In DTN routing, hit (success) rate and delay are the two major performance metrics [2]–[9]. Then, a desired performance objective often tends to minimize or maximize one of it. On the other hand, TTL (Time to Live) may or may not be configured for each packet in different application scenarios. Therefore, following [7], [9], we consider four routing performance objectives: 1) *Minimal average delay with TTL*; 2) *Maximal hit rate with TTL*; 3) *Minimal average delay without TTL*; and 4) *Minimal maximal delay without TTL*. In above definition, with and without TTL refers to whether a TTL is configured for each packet. The first objective aims to minimizes the average delay of successfully delivered packets when TTL is configured for each packet. The second objective aims to 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 aim to minimize the average delay and the maximal delay of all successfully delivered packets when there is no TTL, respectively.

3.3 Realizing Different Routing Performance Objectives When Nodes are Fully Cooperative

Following the work in [7], we summarize the routing strategy that can realize different performance objectives when nodes are fully cooperative, i.e. willing to follow the two aspects of cooperation in this section. However, the strategy cannot be adopted by node naturally when they are selfish. This means an incentive scheme, e.g., Multicent, is needed to encourage nodes to follow it.

3.3.1 General Principles

In the strategy, each packet is associated with a utility value, denoted U_k. It is a metric that is positively related to the desired routing 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 higher priority, the forwarding or storage of a packet enhances the desired routing objective for the most. Finally, the overall routing objective is maximized [7]. In summary, two encountered nodes need to follow below two rules to realize the desired performance objective,

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

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

The calculation of the utility value is introduced in next subsection.

3.3.2 Utility Calculation

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

- *Minimal average delay with TTL*: $U_k = -D_k / Pr\{T_{r_k} < TTL - T_{l_k}\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

- *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 packets on a node. We use a negative utility to ensure that the packet with a smaller maximal delay has a larger utility value.

We need to deduce D_k and $Pr\{T_{r_k} < TTL - T_{l_k}\}$ to calculate these utilities. As indicated in [7], 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. Both our experiment and [7] validate such an assumption. We also assume that the packet size is small so that a packet only needs to meet the destination once to be delivered.

Suppose node N_i has a packet k for node N_j. Then, the probability that the remaining time needed to deliver packet k to N_j is less than T can be expressed by

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

Considering the memorylessness of the exponential distribution, the average remaining time ($\bar{d}_{ij}$) for packet k to meet N_j is $\bar{d}_{ij} = 1/\lambda_{ij}$.

In multiple-copy DTN routing, each packet usually has several copies in the network. 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. Similarly, the calculation of the probability that the remaining delivery time of packet k is less than T should also consider all copies.

Although above calculation matches with 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 [7] 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. Then the D_k and $Pr\{T_{r_k} < TTL - T_{l_k}\}$ for packet k on node N_i are calculated as

$$D_k = T_{l_k} + \bar{d}_{ij} = T_{l_k} + 1/\lambda_{ij} \tag{2}$$

$$Pr(T_{r_k} < TTL - T_{l_k}) = 1 - e^{-\lambda_{ij}(TTL - T_{l_k})} \tag{3}$$

The λ represents the encountering frequency between two nodes. It is updated when 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.

4 System Design

In this section, we introduce the design of Multicent based on the discussion in previous section. We first introduce the design goal in Section 4.1 and then model the packet forwarding process between two nodes as a game in Section 4.2. We further present the incentive schemes in Multicent in Section 4.3 and the validation of Multicent in Section 4.4 based on the game. We also discuss the credit clearance in Section 4.5 and how to realize adjustable QoS in Section 4.6.

4.1 Design Goal

In Section 3.3, we present how to realize different routing performance objects when nodes are fully cooperative. However, nodes may be selfish or malicious. Therefore, the goal of Multicent is to provide incentives in DTN routing so that nodes can earn the most remuneration when they are willing to participate in packet forwarding (first aspect of cooperation) and follow the routing strategy (i.e., $R1$ and $R2$) introduced in Section 3.3.1 (second aspect of cooperation).

4.2 Packet Forwarding Game

When two nodes meet, they exchange packets during the communication session. Such a process can be regarded as a packet forwarding game consisting of a number of interactions between the two nodes. The number of interactions is determined by the length of the communication session. In each interaction, both nodes select their forwarding or storage strategies, i.e., which packet to forward to the other and which packet to discard if the storage is full. Note that the forwarding/storage strategy refers to a specific action of a node while the routing strategy defined in Section 1 denotes the general rules to realize a routing performance objective. We first analyze the remuneration for a node under different forwarding/storage strategies in one interaction, based on which we design the payoff function in Multicent in the next section.

Suppose node N_i meets node N_j ($i, j \in [1, K]$ and $i \neq j$) 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. We use S_{ia} to represent the forwarding strategy that N_i *forwards packet* P_{ia} *to* N_j, and R_{ia} to represent the storage strategy that N_i *discards packet* P_{ia} *after receiving a packet from* N_j. Further, R_{i0} denotes *storing the received packet directly without discarding any packet*, NS means *no forwarding*, and NR means *rejecting the forwarded packet*. The notations apply to S_{jb} and R_{jb} ($b \in [1, m_j]$) similarly. Finally, the strategy sets of the two nodes, denoted by $\{\vec{S_i}, \vec{R_i}\}$ and $\{\vec{S_j}, \vec{R_j}\}$, are

$$\vec{S_i} = \{S_{i1}, S_{i2}, S_{i3}, \cdots, S_{im_i}, NS\}, \tag{4}$$

$$\vec{R_i} = \{R_{i0}, R_{i1}, R_{i2}, R_{i3}, \cdots, R_{im_i}, NR\}, \tag{5}$$

$$\vec{S_j} = \{S_{j1}, S_{j2}, S_{j3}, \cdots, S_{jm_j}, NS\}, \tag{6}$$

$$\vec{R_j} = \{R_{j0}, R_{j1}, R_{j2}, R_{j3}, \cdots, R_{jm_j}, NR\}. \tag{7}$$

Then, in each interaction, each node selects one strategy from its strategy set. 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 storage 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 interaction is shown in Table 1.

TABLE 1: One interaction 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 1, $\vec{O}_{ij} = \{O_i, O_j\}$ represents 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 1, 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 and r_n, where $m, n \in \{i, j\}$ and $m \neq n$, represent the forwarding strategy and storage strategy that result in maximal O_i and O_j, denoted O_i^* and O_j^*. Then, if $O_i^* > 0$ and $O_j^* > 0$, $\{s_m, r_n\}$ can maximally benefit both N_i and N_j.
- Otherwise, $\{s_m, r_n\}$ can only benefit one or none of the two nodes, i.e. N_i or N_j.

4.3 Game Theoretical Incentive Scheme

In this section, we first introduce the principles needed to encourage nodes to achieve a performance objective in Section 4.3.1. Then, the key of Multicent is to design a payoff function that can realize these principles, which is introduced in Section 4.3.2.

4.3.1 General Discussion

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

- First, as mentioned in Section 4.2, two nodes can possibly earn profit only when one node forwards a packet and the other accepts it. Then, we need to encourage both nodes to be cooperative in deciding the sequence of sending packets between them.
- Second, to realize a performance objective, we need to encourage nodes to forward and store packets following *R1* and *R2* described in Section 3.3.1.

In below, we introduce how the two goals can be attained in Multicent.

How to decide the packet sender? To encourage the two encountered nodes to be cooperative in determining who is the sender and who is the receiver, Multicent splits the reward for each forwarding evenly to both packet sender and receiver. Consequently, they would agree that the holder of the packet that can bring about the most benefit will be the sender in the first since this can bring about the most benefit for both of them. Then, the holder of the packet with the second highest benefit will be the sender, 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, thereby 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 the qualified holder of a packet by the amount of credit that is in proportion to the packet's utility. A node is qualified for the storage reward for a packet when it stores the packet until the packet is successfully delivered to the destination or is expired due to TTL. 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. In Multicent, we still reward nodes that have forwarded these unsuccessful packets. This is to ensure that when a node decides the forwarding or storing priority of a packet, it does not need to consider the probability of successful delivery but only the designed utility value, as required by the aforementioned routing strategy.

4.3.2 Payoff Function Design

With all above discussion, we summarize the payoff function in Multicent that can attain the design goal. For better demonstration, we formalize the routing of a packet in multi-copy routing algorithms as a tree structure, as shown in Figure 1. In the multi-copy routing, each node that contains the packet further replicates it until learning that one of the replica of the packet has been delivered to the destination or the replica on 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 the destination. The latter case occurs when a packet is expired or is replaced due to storage limit on a node.

Then, the payoff function in Multicent includes:

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 the increase of the utility value associated with the replication,

$$C_{s_i} = \mathbb{F}_s(\triangle U_i) = \alpha \triangle U_i \ (\alpha > 0). \tag{8}$$

where $\triangle U_i$ is the increase of the utility value and is calculated as the new utility value decreases the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

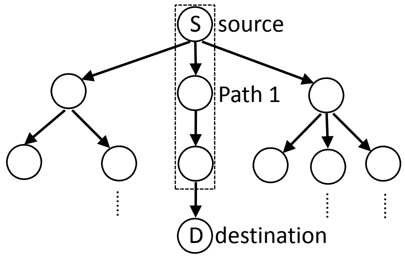


Fig. 1: Packet routing tree.

previous utility value. C_{s_i} is 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 amount of credits C_{r_i} ($C_r \ll C_s/2$), which is proportional to the packet's utility value on the node,

$$C_{r_i} = \mathbb{F}_r(U_i) = \beta U_i \ (\beta > 0); \quad (9)$$

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 $[f_{r1}, f_{r2}]$ represent the ranges of the result 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 $f_{r2} = f_{s1}/6$ and $C_d = 3 * f_{s2}/2$. The credits are assigned to nodes by a central server based on reports of the packet forwarding and storage. We explain this process later in Section 4.5.

4.4 Incentive Scheme Validation

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

4.4.1 The First Aspect of Cooperation (Forwarding/storing Packets)

In Multicent, each forwarding is recognized in Multicent, even when it eventually fails 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. Further, the additional credit C_d encourages nodes to delivery packets to their destinations upon encountering them and meanwhile save space for other packets. 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, thereby realizing the first aspect of cooperation.

4.4.2 The Second Aspect of Cooperation (Following the Routing Strategy)

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 forwarding and storage strategy set, as introduced in Section 4.2. Then, based on Equation (8) and P2, the benefits for N_i and N_j, denoted by O_{is} and O_{js}, for forwarding packet P_{ia} are

$$\begin{cases} O_{is} = \mathbb{F}_s(\triangle U_a)/2 \\ O_{js} = \mathbb{F}_s(\triangle U_a)/2. \end{cases} \quad (10)$$

Also, based on Equation (9), the benefit for N_i and N_j for the storage strategy, i.e., R_{jb} (including R_{j0}), is

$$\begin{cases} O_{ir} = 0 \\ O_{jr} = \mathbb{F}_r(U_a) - \mathbb{F}_r(U_b) \end{cases} \quad (11)$$

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

Satisfying Requirement *R1*. Based on Formula (10), we can see that each node takes the packet in its memory with the largest utility increase as the forwarding 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 (10) and (11), the remuneration for the two nodes when P_{ia} or $P_{jb'}$ is forwarded can be represented as Formulas (12) and (13), respectively.

$$\begin{cases} O_i(s_i, r_j) = \mathbb{F}_s(\triangle U_a)/2 \\ O_j(s_i, r_j) = \mathbb{F}_s(\triangle U_a)/2 + \mathbb{F}_r(U_a) - \mathbb{F}_r(U_b) \end{cases} \quad (12)$$

$$\begin{cases} O_i(r_i, s_j) = \mathbb{F}_s(\triangle U_{b'})/2 + \mathbb{F}_r(U_{b'}) - \mathbb{F}_r(U_{a'}) \\ O_j(r_i, s_j) = \mathbb{F}_s(\triangle U_{b'})/2 \end{cases} \quad (13)$$

Without loss of generality, we assume that $\triangle U_a$ is larger than $\triangle U_{b'}$. 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 $O_j(r_i, s_j)$, i.e., $\mathbb{F}_r(U_a) - \mathbb{F}_r(U_b) > 0$ and $\mathbb{F}_s(\triangle U_a)/2 > \mathbb{F}_s(\triangle U_{b'})/2$. 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 the other to send a packet, which wastes the communication session and results in no benefit for them. Therefore, we augment Multicent with an additional policy. In the policy, when two nodes find that they are waiting the other to send a packet, they would cooperate to choose the packet with larger utility increase to be forwarded. In conclusion, packet with the highest utility increase will be forwarded first, thus 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 discard the packet that has the least $\mathbb{F}_r(U)$ if its storage is used up. As a result, when a new packet arrives, the lowest-utility packet is discarded when the storage is full. This means that the second requirement (*R2*) is satisfied.

With the above analysis, we see that the payoff function introduced in Section 4.3.2 makes *R1* and *R2* the Nash equilibrium 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. Further, 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 realizing equal forwarding opportunity among packets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

4.5 Credit Clearance

As other DTN incentive systems [12], [15], [25], we assume an off-line virtual bank (OVB) 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 connecting to it. In below, we discuss the credit clearance for packet forwarding and storing separately.

Forwarding. In order to assign reward for packet forwarding, each intermediate node in a path of a packet imprints a contribution unit into the packet during the routing. A node's contribution unit includes its identity and its contribution to the packet forwarding ($\triangle U_i$). Then, the last node on a path forms a report indicating the contributions of all forwarders on the path and sends the report to the OVB. This process follows the payoff functions P1 and P2 introduced in Section 4.3.2. 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, which follows the payoff function P4.

Storing. For each packet, Multicent rewards nodes that hold it until it is expired due to TTL or has been delivered to the destination. Every node creates a report when it finds that a packet in it satisfies either of the two requirements. The node is also required to send the packet along with the report to another node, which signs the report if it finds that the report is valid. This prevents nodes from fabricating such reports. Then, the report is sent to the OVB for credit assignment. This process follows the 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 [12], [13], [15], a certain amount of credits are charged from the destination node for the packet forwarding and storage services provided by the forwarders of the packet. 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 forwards 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 the IDs of malicious nodes to all nodes quickly and meanwhile alleviate the 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.

4.6 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 can 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 credits 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 payoff 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 for packets.

5 Performance Evaluation

5.1 Experiment Settings

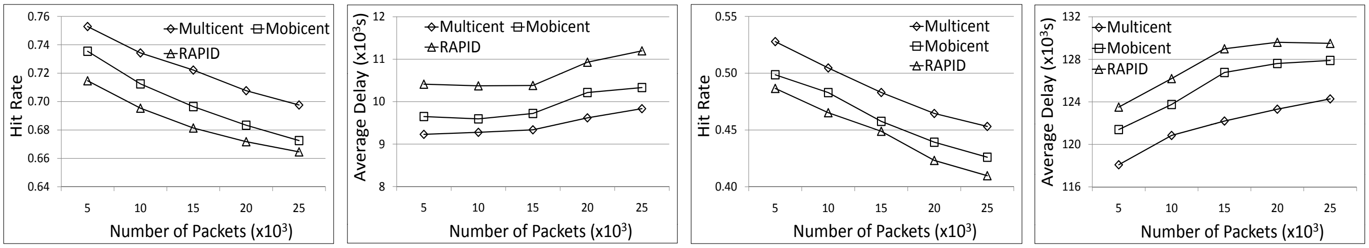
We evaluated Multicent through trace-driven tests with datasets from the MIT Reality project [27] (97 nodes) and the Haggle project [28] (98 nodes). In the test, the mean disconnection time (λ) is measured and updated whenever two nodes meet. During the test, there were no queries in the first 1/3 of 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. Following the definitions of routing performance objectives in Section 3.2, we measured three metrics in the experiment: hit rate, average delay, and maximal delay. They represent the percentage of arrived, i.e., successfully delivered, packets, the average delay of all arrived packets, and the maximal delay of all arrived packets, respectively. We set α and β in Equation (8) and (9) to 1. We adopted 95% confidence interval in analyzing experimental results.

We first validate the effectiveness of Multicent in comparison to Mobicent [15] and RAPID [7]. We also evaluate the ability of Multicent in supporting different performance objectives and adjustable QoS. Mobicent provides the same amount of reward to each forwarding action but neglects the impact of different routing strategies on 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. We further measured the ranges of utilities and utility increases in real DTN routing scenarios and the effectiveness of some extensions to justify the applicability of Multicent. Due to page limit, these additional experiment results are shown in Appendix B.

5.2 Performance Comparison

To make the results comparable, we set Multicent to the maximum hit rate mode (MaxHitRate) since Mobicent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems



(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.

and RAPID focus on hit rate. We set 10% of nodes as selfish nodes that forward or store packets only when they can benefit from it. All other nodes are cooperative and naturally follow the two aspects of cooperation. In Multicent, selfish nodes follow the two aspects of cooperation. In Mobicent, selfish nodes only follow the first aspect of cooperation. In RAPID, selfish nodes drop all packets since no incentives are provided.

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, they 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 of cooperation, Multicent takes full advantage of forwarding opportunities and gives higher priority to packets that can bring about more improvement to hit rate in forwarding and storing, thus generating the 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 the best performance.

5.3 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 realize 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 two datasets.

5.3.1 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.

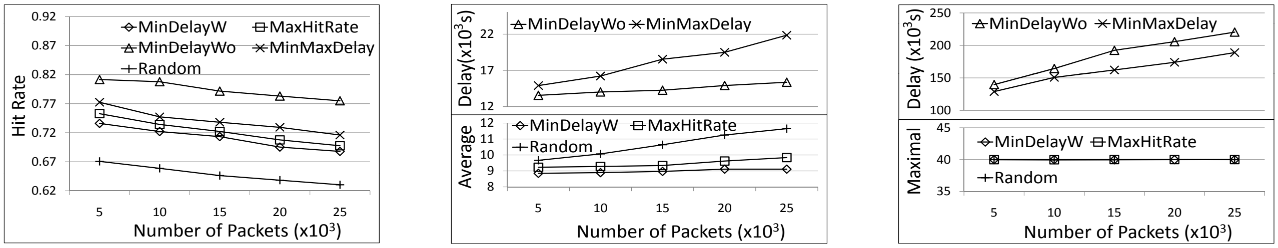
In the two methods without TTL, we find that the hit rate of MinDelayWo is much higher than that of MinMaxDelay. In MinDelayWo, nodes are motivated to first forward packets that can result in the maximal decrease in the estimated delay. In other words, the routing aims to reduce the delay of all packets,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

5.3.2 Average Delay

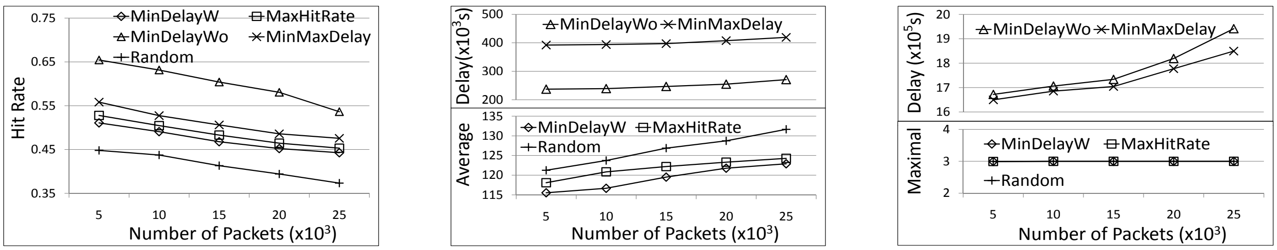
Figure 3(b) and Figure 4(b) illustrate the average delay under the four modes with the two datasets, respectively. Note the vertical axis (y-axis) is split in the two figures to better demonstrate the differences. Since modes with TTL generate much lower average delay than those without TTL, we discuss the results of modes with and without TTL separately. We observe that the average

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems



(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.

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. This confirms that Multicent is effective in achieving desired metrics.

In the two modes without TTL, we also see that the average delay of MinDelayWo is smaller than that of MinMaxDelay. Such a result proves that the designed objective 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 confirms the effectiveness of Multicent in enhancing system performance by considering forwarding and storing priorities for different packets.

5.3.3 Maximal Delay

Figure 3(c) and Figure 4(c) plot the maximal delay under the four modes with the two datasets, respectively. The vertical axis (y-axis) is also split to better demonstrate the differences. We see that MinDelay and MaxHitRate produce a similar maximal delay, i.e., around 40,000 seconds and 300,000 seconds with the two datasets, respectively. We find 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 maximal 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.

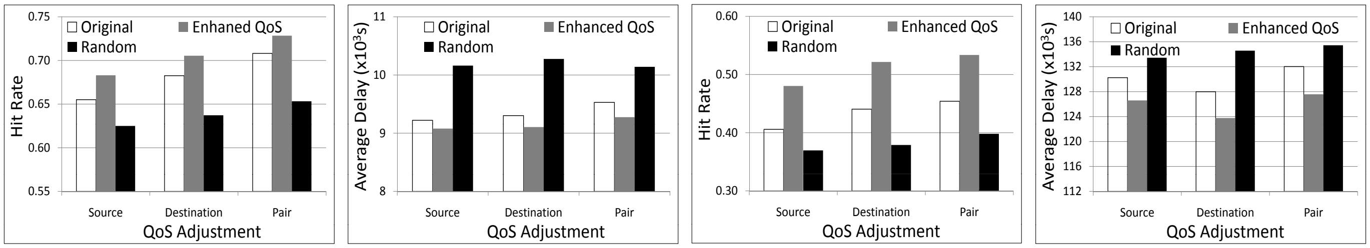
5.4 Supporting Adjustable QoS

In this section, we verify Multicent's ability to support adjustable QoS for packets from specific sources, to specific destinations, or between specific source-destination pairs. We name the three QoS adjustment options as *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.

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 similar results as those from the test with 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 between 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems



(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.

improve the performance of the enhanced objects.

6 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 needed to realize different performance objectives in DTN routing: nodes should not only participate in packet forwarding but also forward or store packets as desired by a performance objective (e.g., minimal average delay, maximal hit rate, and minimal maximal delay). To this end, we proposed Multicent, an incentive scheme for DTN routing that can encourage nodes to follow the two aspects of cooperation to realize different performance objectives. It can also realize adjustable QoS for packets of specific sources, destinations, or source-destination pairs. Trace-driven experiments verify the correctness and effectiveness of Multicent in comparison with other schemes. In the future, we plan to investigate how to thwart more advanced attacks such as Denial of Service.

Acknowledgements

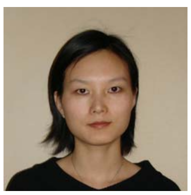
This research was supported in part by U.S. NSF grants IIS-1354123, CNS-1254006, CNS-1249603, CNS-1049947, CNS-0917056 and CNS-1025652, Microsoft Research Faculty Fellowship 8300751. An early version of this work was presented in the Proceedings of SECON 2013 [29].

References

[1] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. of SIGCOMM*, 2003, pp. 27–34.
[2] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks." *Mobile Computing and Communications Review*, vol. 7, 2003.
[3] F. Li and J. Wu, "Mops: Providing content-based service in disruption-tolerant networks," in *Proc. of ICDCS*, 2009.
[4] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *Proc. of MobiHoc*, 2008.
[5] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proc. of MobiHoc*, 2007.
[6] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep., 2000.
[7] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem." in *Proc. of SIGCOMM*, 2007.
[8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks." in *Proc. of INFOCOM*, 2006.
[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] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks." in *Proc. of INFOCOM*, 2010.
[11] Y. L. 0008, G. Su, D. O. Wu, D. Jin, L. Su, and L. Zeng, "The impact of node selfishness on multicasting in delay tolerant networks." *IEEE T. Vehicular Technology*, vol. 60, no. 5, pp. 2224–2238, 2011.
[12] 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.
[13] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss, "Pi: A practical incentive protocol for delay tolerant networks." *IEEE TWC.*, 2010.
[14] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang, "Incentive-aware routing in DTNs," in *Proc. of ICNP*, 2008.
[15] B. B. Chen and M. C. Chan, "MobiCent: a credit-based incentive system for disruption tolerant network." in *Proc. of INFOCOM*, 2010.
[16] A. Mei and J. Stefa, "Give2get: Forwarding in social mobile wireless networks of selfish individuals." in *Proc. of ICDCS*, 2010.
[17] T. Ning, Z. Yang, X. Xie, and H. Wu, "Incentive-aware data dissemination in delay-tolerant mobile networks." in *Proc. of SECON*, 2011.
[18] L. Wei, H. Zhu, Z. Cao, and X. S. Shen, "Mobiid: A user-centric and social-aware reputation based incentive scheme for delay/disruption tolerant networks." in *Proc. of ADHOC-NOW*, 2011.
[19] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks." in *Proc. of MOBICOM*, 2000.
[20] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks." in *Proc. of CMS*, 2002.
[21] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol." in *Proc. of MobiHoc*, 2002.
[22] 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.
[23] K. Chen and K. Nahrstedt, "iPass: An incentive compatible auction scheme to enable packet forwarding service in MANET." in *Proc. of ICDCS*, 2004.
[24] M. E. Mahmoud and X. Shen, "Stimulating cooperation in multi-hop wireless networks using cheating detection system." in *Proc. of INFOCOM*, 2010.
[25] S. Zhong, J. Chen, and Y. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks." in *Proc. of INFOCOM*, 2003.
[26] S. Zhong, E. Li, Y. Liu, and Y. 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.
[27] N. Eagle, A. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," *PNAS*, vol. 106, no. 36, 2009.
[28] 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.
[29] K. Chen and H. Shen, "Multicent: A multifunctional incentive scheme adaptive to diverse performance objectives for dtn routing." in *Proc. of SECON*, 2013.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2014.2323057, IEEE Transactions on Parallel and Distributed Systems

Kang Chen Kang Chen received the BS degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China in 2005, and the MS in Communication and Information Systems from the Graduate University of Chinese Academy of Sciences, China in 2008. He is currently a Ph.D student in the Department of Electrical and Computer Engineering at Clemson University. His research interests include mobile ad hoc networks and delay tolerant networks.



Haiying Shen Haiying Shen received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively. She is currently an Associate Professor in the Department of Electrical and Computer Engineering at Clemson University. Her research interests include distributed computer systems and computer networks, with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and grid and cloud computing. She is a Microsoft Faculty Fellow of 2010, a senior member of the IEEE and a member of the ACM.

Li Yan Li Yan received the BS degree in Information Engineering from Xi'an Jiaotong University, China in 2010, and the M.S. degree in Electrical Engineering from University of Florida in 2013. He currently is a Ph.D. student in the Department of Electrical and Computer Engineering at Clemson University, SC, United States. His research interests include wireless networks, with an emphasis on delay tolerant networks and sensor networks.