

Article

# A Credit-Based Congestion-Aware Incentive Scheme for DTNs

Qingfeng Jiang <sup>1,2,\*</sup>, Chaoguang Men <sup>1</sup> and Zeyu Tian <sup>1</sup>

<sup>1</sup> College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China; menchaoguang@hrbeu.edu.cn (C.M.); tianzeyu@hrbeu.edu.cn (Z.T.)

<sup>2</sup> College of Computer Science and Information Technology, Daqing Normal University, Daqing 163712, China

\* Correspondence: qingfeng\_jiang@163.com; Tel.: +86-459-595-7502

Academic Editor: Willy Susilo

Received: 15 September 2016; Accepted: 3 December 2016; Published: 15 December 2016

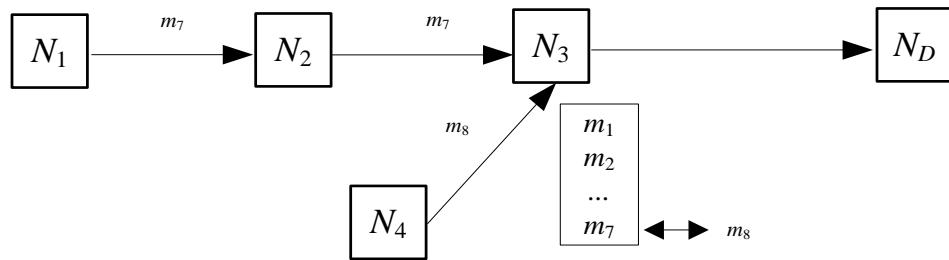
**Abstract:** In Delay-Tolerant Networks (DTNs), nodes may be selfish and reluctant to expend their precious resources on forwarding messages for others. Therefore, an incentive scheme is necessary to motivate selfish nodes to cooperatively forward messages. However, the current incentive schemes mainly focus on encouraging nodes to participate in message forwarding, without considering the node congestion problem. When many messages are forwarded to the nodes with high connection degree, these nodes will become congested and deliberately discard messages, which will seriously degrade the routing performance and reduce the benefits of other nodes. To address this problem, we propose a credit-based congestion-aware incentive scheme (CBCAIS) for DTNs. In CBCAIS, a check and punishment mechanism is proposed to prevent forwarding nodes from deliberately discarding message. In addition, a message acceptance selection mechanism is proposed to allow the nodes to decide whether to accept other messages, according to self congestion degree. The experimental results show that CBCAIS can effectively stimulate selfish nodes to cooperatively forward messages, and achieve a higher message delivery ratio with lower overhead ratio, compared with other schemes.

**Keywords:** delay-tolerant networks (DTNs); congestion; incentive scheme; security

## 1. Introduction

Delay-Tolerant Networks (DTNs) are characterized by intermittent connectivity and often lack an end-to-end contemporaneous path [1]. At present, DTNs have been applied to the areas of pocket switched networks, vehicular networks, wildlife monitoring, underwater networks, etc. Different from traditional networks, DTNs lack guaranteed connectivity. Therefore, the message forwarding process in DTNs follows a “store-carry-and-forward” manner, i.e., messages are opportunistically routed toward the destination nodes. Many data forwarding and dissemination protocols [2–8] have been proposed to effectively deliver messages. These routing protocols require nodes to forward messages in a cooperative and selfless way. For example, when the next hop is not immediately available for a node to forward a message, the node should utilize its own limited buffer to store the message, carry the message along the movement, and forward the message when it moves within the transmission range of other nodes that will help to deliver this message further. However, in the real world, most nodes exhibit uncooperative behaviors, e.g., a node may not be willing to store messages in its buffer or relay messages on behalf of others to conserve limited buffer and power resources [9,10]. That is to say, nodes are selfish in the message forwarding and relaying process. All these protocols assume that nodes are ready to forward messages to each other, and the performance will degrade seriously when nodes refuse to forward messages to others due to selfishness [9,11]. Therefore, it is imperative to design an effective incentive scheme to stimulate selfish nodes to cooperatively forward messages.

At present, some credit-based incentive schemes [12–16], which motivate selfish nodes to participate in message forwarding by virtual credits, have been proposed, but these schemes just focus on how to encourage nodes to deliver messages to others with higher delivery probability or smaller delivery delay, without considering the node congestion problem. Pujol et al. [17] has shown that in forwarding routing, most messages will be forwarded to the nodes with high connection degree, resulting in an unbalanced load among nodes. When many messages are forwarded to nodes with high connection degree, then these nodes will be congested and discard most of the messages due to limited buffer. Discarding the messages will seriously degrade the routing performance and reduce the benefits to other nodes, which is unfair to other nodes. As shown in Figure 1, suppose node  $N_3$  is a congested node with high connection degree. When receiving message  $m_8$  from  $N_4$ , it will discard  $m_7$  if which can bring more benefits. The discarding will reduce the benefits of nodes  $N_1$ ,  $N_2$ , or  $N_D$ . For example, in the scheme proposed by Zhu et al. [14], the message-forwarding nodes are paid some credits only when the message is successfully delivered, so nodes  $N_1$  and  $N_2$  will not get their credits due to the discarding. In the scheme proposed by Chen et al. [15], the destination node has to pay some credits to the forwarding nodes even if a message is not received, so destination node  $N_D$  will lose some credits due to the discarding.



**Figure 1.** Discarding message due to congestion.

To address the above problem, we propose a credit-based congestion-aware incentive scheme (CBCAIS) for DTNs. The CBCAIS scheme will check if a forwarding node has discarded messages deliberately by a check and punishment mechanism, and punish the node seriously if it has. When congested nodes do not accept messages anymore to avoid being punished, these messages will be transferred to other nodes with low congestion, and finally to the destination node. A node is prevented from discarding messages deliberately, but it can decide whether to accept other nodes' messages autonomously in CBCAIS. To obtain more credits, a congested node can decide whether to accept other nodes' messages by a message acceptance selection mechanism, where messages are accepted according to node self congestion degree. In case of congestion, the message acceptance selection mechanism will increase the acceptance threshold to accept the messages that can bring more credits. Otherwise the threshold is decreased to accept as many messages as possible.

The main contributions of this paper can be summarized as follows:

- (1) We propose a delay-based credit reward scheme, which can effectively stimulate selfish nodes to forward messages to others according to delivery delay.
- (2) We propose a check and punishment mechanism to prevent congested nodes from discarding messages deliberately.
- (3) To further enhance the routing performance, we propose a message acceptance selection mechanism to allow nodes to decide whether to accept other nodes' messages according to self congestion degree.

The rest of the paper is organized as follows. In Section 2, we review the related work. We present the system model and design goals in Section 3. We describe the incentive scheme CBCAIS in Section 4 and the simulation results of CBCAIS are given in Section 5. The paper is concluded in Section 6.

## 2. Related Work

In [1], Fall proposes the delay-tolerant networking architecture for interoperability between and among challenged networks. In addition, Fall and other authors in [2] formulate the delay-tolerant networking routing problem and propose a framework for evaluating routing algorithms in such environments. Now, many routing and dissemination protocols have been proposed to effectively deliver messages. For example, Lindgren et al. [3] propose a probabilistic routing protocol for intermittently connected networks. When a node encounters another node, they will exchange information about the delivery predictabilities, and decide to forward messages based on the delivery predictabilities. Erramilli et al. [4] develop a new strategy for forwarding, which is referred to as delegation forwarding. Hui et al. [5] design and evaluate BUBBLE, a novel social-based forwarding algorithm, which utilizes the metrics of centrality and community to enhance delivery performance. Costa et al. [6] propose SocialCast, a routing framework for publish-subscribe that exploits predictions based on the metrics of social interaction to identify the best information carriers. Burgess et al. [7] propose MaxProp, a protocol for effective routing of DTN messages. MaxProp is based on prioritizing both the schedule of packets transmitted to other peers and the schedule of packets to be dropped. MaxProp performs better than protocols that have access to an oracle that knows the schedule of meetings between peers. Spyropoulos et al. [8] propose multiple-copy routing called Spay-and-Wait, which is highly scalable and retains good performance under a large range of scenarios.

All the above works assume that nodes are cooperative to forward messages to each other, without considering the selfishness problem. Li et al. [9] analytically assesses the performance of the unrestricted and two-hop relay schemes when nodes behave selfishly in forwarding messages. The experimental results show that the performance decreases both with the number of selfish nodes and the intensity of their selfishness. Karaliopoulos et al. [11] investigate how the selfish behavior of nodes affects the performance of DTN multicast. This demonstrates that the selfishness increases the message transmission delay and cost.

The existing incentive schemes to address the selfishness problem basically fall into three categories: reputation-based [18–20], barter-based [10,21,22], and credit-based [12–16].

Zhang et al. [18] propose a practical reputation-based incentive scheme, named PRI, to stimulate honest forwarding and discipline selfish behavior. In PRI, each node provides service of forwarding for honest nodes in advance, and excludes selfish nodes whose values are below a deterministic reputation threshold, to encourage nodes to cooperatively forward messages. To promote the selfish nodes to cooperatively forward messages, Wei et al. [19] propose SUCCESS, a secure user-centric and social-aware reputation-based incentive scheme for DTNs, is proposed. Two concepts of self-check and community-check are defined for reputation evaluation and speeding up reputation establishment. To address the selfishness problem, Li et al. [20] propose a reputation-based incentive scheme to motivate participants to disseminate reliable data in a participatory sensing system, named RIDD, while minimizing the incentive cost for maintaining a sufficient number of reliable participants. Reputation-based schemes are vulnerable to the collusion attack, in which several nodes may collude with each other to get a better reputation by giving false praise.

Shevade et al. [10] study the impact of selfish behavior and propose the use of pair-wise Tit-for-Tat (TFT) as an incentive mechanism for DTNs. In TFT, each node statistics the number of messages have been forwarded by others in each interval, and then forwards the corresponding number of messages to others in the next interval. For the node selfishness in opportunistic networks, Buttyan et al. [21] propose a barter-based cooperation system to increase the message delivery ratio. The authors develop a game-theoretic model in which the proposed approach indeed stimulates the cooperation of the nodes. The results show that the message delivery ratio considerably increases if the mobile nodes follow the Nash equilibrium strategy in the proposed mechanism, compared to the data dissemination protocol when no encouraging mechanism is present. Ning et al. [22] propose a self-interest-driven (SID) incentive scheme to stimulate cooperation among selfish nodes for dissemination in autonomous mobile social networks. They formulate the nodal interaction as a two-player cooperative game whose

solution is obtained by the Nash bargaining theorem. The barter-based schemes will dramatically degrade routing performance if one of the two encountering nodes has fewer messages.

Wang et al. [12] propose a data dissemination scheme, called MuRIS, to allow nodes to cooperatively deliver information of interest to one another via chosen paths utilizing few transmissions. Chen et al. [13] propose a credit-based incentive system, called MobiCent, to stimulate DTN selfish nodes to cooperatively forward messages to others. MobiCent also provides different payment schemes to cater to a client that wants to minimize either payment or delivery delay. Zhu et al. [14] propose a secure multilayer credit-based incentive scheme, SMART, to stimulate cooperation among DTN nodes. In SMART, a layered coins model is presented and serves as virtual currency to pay for the relays who participate in forwarding messages. Chen et al. [15] propose Multicent, a game theoretical incentive scheme that not only provides cooperative incentive but also encourages nodes to follow defined rules to realize the desired performance objective. Wu et al. [16] present a novel incentive scheme to stimulate selfish nodes to participate in probabilistic routing with credit and bargaining game theory. They model the message exchange process between a pair of nodes as a bargaining game. Hence, when a message is transferred from its source to its destination, it goes through a series of bargaining games. When this series of bargaining games is completed, the message reaches its destination.

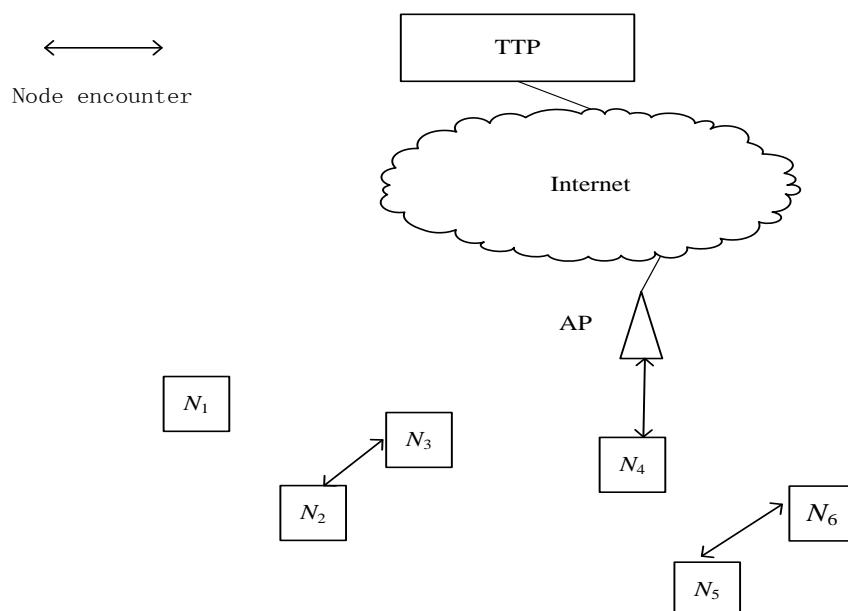
The above credit-based schemes only motivate nodes to deliver messages to other nodes with higher delivery utility, without considering the node congestion problem. So we propose the credit-based congestion-aware incentive scheme CBCAIS for DTNs to improve the single-copy routing performance when nodes are congested.

### 3. System Model and Design Goal

#### 3.1. Network Model

As shown in Figure 2, the components of the system network model are as follows:

Trusted Third Party (TTP) as the central authority is responsible for issuing public and private key certificates and completing the verification and credit clearance service for DTN mobile nodes.



**Figure 2.** Network model.

Fixed Network, including Internet and access points (AP), is responsible for connecting DTN nodes and TTP. DTN mobile nodes can obtain public and private keys from the TTP via AP and the

Internet. When encountering an AP, the message destination node will send the ticket information to TTP through AP, then TTP will achieve the credit clearance for mobile nodes.

Mobile nodes  $N_1-N_6$  in Figure 2 denote some pedestrians who carry equipment with Bluetooth and Wi-Fi function. Some mobile nodes send messages about traffic and location to other remote nodes. The value of these messages will decrease over time. When a node encounters another node with smaller delivery delay, it will forward its messages to the node. When a message is successfully delivered to the destination node, TTP will deduct some credits from the destination node to pay for the forwarding nodes according to the reward scheme in Section 4.1.

### 3.2. Node Contact Model and Delivery Delay Calculate

In the CBCAIS system model, messages will be forwarded to nodes with smaller delivery delay, so each node needs to calculate the message delivery delay to destination node. Some studies [23,24] found that the pair-wise node inter-contact time (the time gap separating two contacts of the same pair of nodes) in realistic traces follows an exponential distribution. So we also assume that the distribution of pair-wise node inter-contact time in DTNs follows the exponential distribution with mean value  $1/\lambda_{ij}$ , where  $\lambda_{ij}$  represents the contact frequency between nodes  $N_i$  and  $N_j$ . The value of  $\lambda_{ij}$  can be calculated by the time average method in Equation (1):

$$\lambda_{ij} = n / \sum_{l=1}^n Time_{N_i-N_j}^l, \quad (1)$$

where  $Time_{N_i-N_j}^1, Time_{N_i-N_j}^2, \dots, Time_{N_i-N_j}^n$  are inter-contact time samples between nodes  $N_i$  and  $N_j$ . So when node  $N_i$  receives a message  $m$  whose destination node is  $N_j$ , the delivery delay of node  $N_i$  for message  $m$  is calculated by Equation (2):

$$Time_{N_i}^{delivery}(m) = 1/\lambda_{ij} + t_{rec} - t_{gen}, \quad (2)$$

where  $t_{gen}$  and  $t_{rec}$  are the message generation time and message receipt time, respectively.

### 3.3. Design Goal

Our goal is to design a congestion-aware incentive scheme for single-copy routing in DTNs, which can stimulate selfish nodes to effectively forward messages to others when the buffer is limited. More specifically, the scheme is to achieve the following goals:

1. Incentive compatible. Our incentive scheme can effectively stimulate selfish nodes to forward messages to others and truthful participation is adopted by all the forwarding nodes despite their selfish nature.
2. Congestion aware. Our scheme can prevent congested nodes from discarding messages deliberately, and improve the routing performance when nodes are congested.

## 4. A Credit-Based Congestion-Aware Incentive Scheme

### 4.1. Delay-Based Reward Scheme

In order to stimulate other nodes to forward messages, the message destination node needs to pay some credits to the forwarding nodes when receiving a message. We propose a delay-based credit reward scheme. In the reward scheme, when a message is successfully delivered, the destination node will pay the forwarding nodes according to the delivery delay, as in Equation (3):

$$R(Time^{delivery}) = \begin{cases} Credit(1 - Time^{delivery}/TTL), & 0 < Time^{delivery} < TTL \\ 0, & Time^{delivery} \geq TTL \end{cases}, \quad (3)$$

where  $R(Time^{delivery})$  is the total number of credits that the destination node will pay for forwarding nodes according to delivery delay  $Time^{delivery}$ ;  $Credit$  is the most credits that the destination node can pay for a message, assuming 1 in this scheme; and TTL is the lifetime of a message.

It can be seen that the smaller the delay, the more credits the forwarding nodes can get. If the delivery delay is larger than TTL, the credits will be 0.

Assuming the forwarding path is  $N_s, N_1 \dots N_k, N_D$ , and the delivery delays are  $Time_{N_s}^{delivery}, Time_{N_1}^{delivery}, Time_{N_2}^{delivery} \dots Time_{N_k}^{delivery}$  respectively,  $Time_{N_s}^{delivery} > Time_{N_1}^{delivery} > Time_{N_2}^{delivery} > \dots Time_{N_k}^{delivery}$ , then the rewards for each forwarding node are  $R(Time_{N_s}^{delivery}) - 0, R(Time_{N_1}^{delivery}) - R(Time_{N_s}^{delivery}), \dots R(Time_{N_k}^{delivery}) - R(Time_{N_{k-1}}^{delivery})$ , respectively. We can see that the credits obtained by each node in CBCAIS only relate to the delivery delay enhancement from the previous node, and the sum is determined by the delivery delay of node  $N_k$  as Equation (4):

$$R(Time_{N_s}^{delivery}) - 0 + R(Time_{N_1}^{delivery}) - R(Time_{N_s}^{delivery}) \dots + R(Time_{N_k}^{delivery}) - R(Time_{N_{k-1}}^{delivery}) = R(Time_{N_k}^{delivery}). \quad (4)$$

**Theorem 1.** *The CBCAIS scheme is incentive-compatible, i.e., truthful participation is adopted by all the forwarding nodes, despite their selfish nature.*

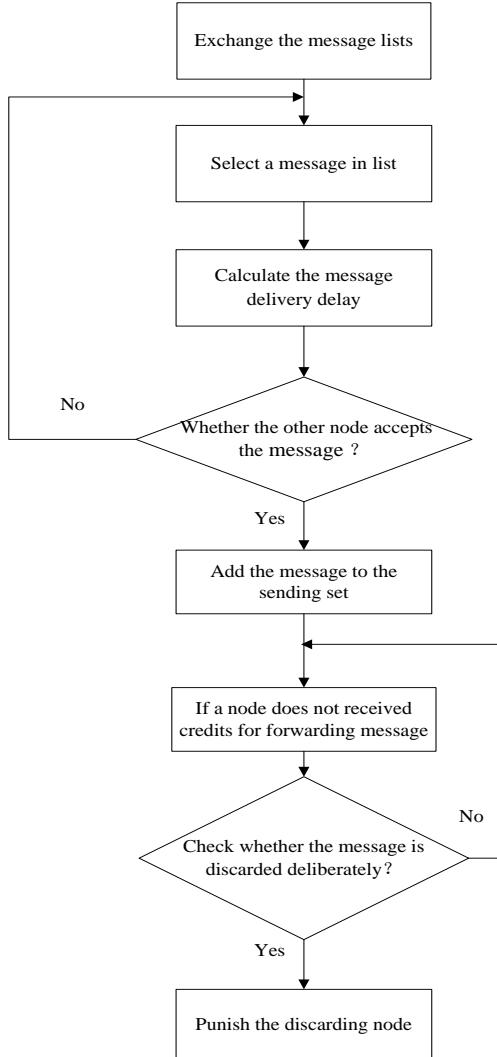
**Proof.** In the delay-based reward scheme, the obtained credits of a node are only determined by the delivery delay enhancement from the previous node, and have no relation with the subsequent node, so a node will forward its messages immediately when encountering another node with smaller delivery delay. On the other hand, if the node does not forward the messages immediately, it will miss some chances to buffer other messages to obtain more credits. Therefore, the CBCAIS scheme is incentive compatible.  $\square$

#### 4.2. CBCAIS Incentive Scheme Overview

The CBCAIS scheme is mainly composed of two functional components: a check and punishment mechanism and a message acceptance selection mechanism. The check and punishment mechanism is used to check whether some congested nodes have discarded messages deliberately, and punish them severely if they have. A node will be punished if it has been discarding messages deliberately, but it can decide whether to accept other nodes' messages autonomously before the buffer is full. To obtain more credits, a message acceptance selection mechanism is used to decide whether to accept other nodes' messages according to the congestion degree. If a node is congested, it can increase the acceptance threshold to accept messages that bring more credits. Otherwise it will decrease the threshold to accept more messages.

Assuming nodes  $N_i$  and  $N_j$  meet, the overview of credit-based congestion-aware incentive scheme CBCAIS is shown in Figure 3; it includes the following steps:

1. Both nodes exchange message lists  $L_i, L_j$ .
2. For a message  $m_i$  in the list  $L_i$  of node  $N_i$ , nodes  $N_i$  and  $N_j$  calculate the message delivery delay.
3. If the message delivery delay of  $N_j$  is smaller than  $N_i$ , then node  $N_j$  will decide whether to accept message  $m_i$  by the message acceptance selection mechanism.
4. If node  $N_j$  decides to accept, then message  $m_i$  is added to the sending set  $SendS_i$ , otherwise node  $N_i$  will go to Step 2 and select another message to send.
5. If node  $N_i$  does not receive the corresponding credits for forwarding message  $m_i$ , it will require TTP to check if a forwarding node has discarded message  $m_i$  deliberately by the check and punishment mechanism. If a misbehavior node is checked, it will be punished for the discarding.
6. The process of sending message  $m_j$  of node  $N_j$  is the same as message  $m_i$ .

**Figure 3.** CBCAIS overview.

#### 4.3. Check and Punishment Mechanism

If a message is discarded deliberately, the misbehavior node must be the last forwarding node on the forwarding path. In order to check the misbehavior node, TTP needs the message forwarding path information. To achieve this, when sending node  $N_S$  encounters a forwarding node  $N_1$  with smaller delivery delay, they will create a forwarding ticket as in Figure 4.

$EK_{N_S}(H(MessageID N_S N_D Credit TTL t_{gen} Path))$								
MessageID	$N_S$	$N_D$	Credit	TTL	$t_{gen}$	Path( $N_S, N_1, t_{N_S-N_1}, \dots$ )	$Sig_{N_S}$	$Sig_{N_1}$

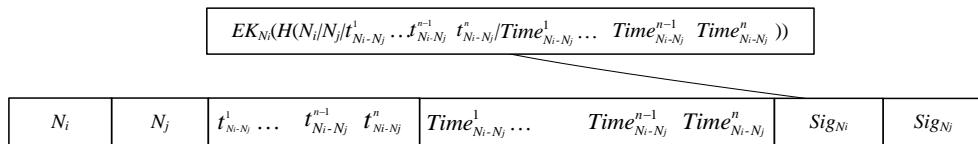
**Figure 4.** Forwarding ticket.

In the forwarding ticket,  $MessageID$  is the message identifier;  $N_S$  and  $N_D$  are the node identifiers of the message sending node and receiving node, i.e., destination node, respectively;  $Credit$  is the most credits to be paid for a successfully delivered message, as in Equation (3);  $TTL$  is the message lifetime;  $t_{gen}$  is the message generation time;  $Path$  is the message forwarding path;  $t_{N_S-N_1}$  is the time when nodes  $N_S$  and  $N_1$  meet;  $Sig_{N_S}$  and  $Sig_{N_1}$  are digital signatures of nodes  $N_S$  and  $N_1$ , respectively;

$EK_{NS}$  is the private key of node  $N_S$ ; and  $H$  is the hash function to generate a summary to ensure the message integrity.

After generating a forwarding ticket, nodes  $N_S$  and  $N_1$  will keep a copy separately as the forwarding proof. When node  $N_1$  encounters another node  $N_2$  with smaller delivery delay, they will insert their node identifiers and encounter time to the forwarding path, and sign the information by the secret key, to generate a new forwarding ticket. Other forwarding nodes will continue the same operation until encountering the destination node. We can see that any node cannot modify the forwarding path and deny the forwarding behavior, because of their digital signatures. So TTP must be able to find the message's last forwarding node by forwarding ticket.

Although TTP can find the last forwarding node, it cannot know whether the message has been dropped deliberately or due to TTL expiring. In order to check whether a message has been dropped deliberately, TTP needs the contact records between the forwarding nodes and destination node. In the CBCAIS scheme, when two nodes  $N_i$  and  $N_j$  meet, they will generate an encounter ticket, as shown in Figure 5.



**Figure 5.** Encounter ticket.

In the encounter ticket,  $N_i$ ,  $N_j$  are node identifiers;  $t^1_{Ni-Nj} \dots t^n_{Ni-Nj}$  are the times when nodes  $N_i$  and  $N_j$  encounter;  $Time^1_{Ni-Nj} \dots Time^n_{Ni-Nj}$  are inter-contact time samples between nodes  $N_i$  and  $N_j$  as in Equation (1);  $Sig_{Ni}$ ,  $Sig_{Nj}$  are digital signatures of nodes  $N_i$  and  $N_j$ , respectively;  $EK_{Ni}$  is the private key of node  $N_i$ ; and  $H$  is the hash function to generate a summary to ensure the integrity of the message.

Combining the forwarding ticket and the encounter ticket, TTP can check if the message has been dropped deliberately by a forwarding node. As shown in Figure 6, we can know the message receiving time  $t_{rec}$ , TTL, and the last encounter time  $t^n_{Ni-N_D}$  between forwarding node  $N_i$  and destination node  $N_D$  from the forwarding ticket and the encounter ticket, respectively. If a forwarding node drops the message at time  $t_{drop}$  before  $t^n_{Ni-N_D}$ , the misbehavior must be found by TTP, because the destination node cannot accept the message at time  $t^n_{Ni-N_D}$ . The misbehavior node will be docked some credits to compensate for the loss of other nodes. In CBCAIS, we assume that the misbehavior node is punished by the removal of lots of credits, so any rational node will not discard messages deliberately.



**Figure 6.** Dropping a message deliberately.

#### 4.4. Message Acceptance Selection Mechanism

As above, we know that a node is prevented from discarding messages before the messages are delivered to the destination node. Therefore, if a node accepts too many messages quickly, it may lose the chance to buffer other messages that can potentially bring more benefits. Thus, a node should assess the buffer congestion degree and then adjust the message acceptance threshold according to the self congestion degree. When congested, a node should increase the message acceptance threshold to accept those messages that bring more benefits. Otherwise, the node should decrease the acceptance threshold to store as many messages as possible.

In order to assess the node congestion degree, the time is divided into many periods of time  $T$ , and a node calculates the *messages changing number* in a period  $T$ .

**Definition 1.** (*incoming message*): When nodes  $N_i$  and  $N_j$  meet, the incoming message of node  $N_i$  is the message from node  $N_j$ , whose delivery delay satisfies the condition that  $\text{Time}_{N_j}^{\text{delivery}} - \text{Time}_{N_i}^{\text{delivery}} \geq \text{Threshold}$ , where  $\text{Threshold}$  denotes a delay threshold.

It is noticed that node  $N_i$  may not accept an *incoming message* if congested.

**Definition 2.** (*outgoing message*): When nodes  $N_i$  and  $N_j$  meet, the outgoing message of node  $N_i$  is the message that has been transferred from the buffer to node  $N_j$ .

**Definition 3.** (*messages changing number*): The messages changing number is the difference between the number of incoming messages and outgoing messages in a period  $T$ , as shown in Equation (5):

$$\text{MessagesChange}_T = \text{MessagesIn}_T - \text{MessagesOut}_T, \quad (5)$$

where  $\text{MessagesIn}_T$  denotes the number of *incoming message* in a period  $T$ ,  $\text{MessagesOut}_T$  denotes the number of *outgoing message* in a period  $T$ , and  $\text{MessagesChange}_T$  denotes *messages changing number*.

To restrain the effect of temporary burst, when the current period  $T$  is time out, the new *message changing number* is predicted by the Exponentially Weighted Moving Average (EWMA) method in Equation (6):

$$\text{MessagesChange}_T^{\text{new}} = \alpha \times \text{MessagesChange}_T^{\text{cur}} + (1 - \alpha) \text{MessagesChange}_T^{\text{old}}, \quad 0 \leq \alpha \leq 1, \quad (6)$$

where  $\text{MessagesChange}_T^{\text{new}}$  is the new *message changing number*,  $\text{MessagesChange}_T^{\text{cur}}$  is the *message changing number* of current period,  $\text{MessagesChange}_T^{\text{old}}$  is the history value, and  $\alpha$  is the weight of  $\text{MessagesChange}_T^{\text{cur}}$ .

When the *messages changing number* is larger than a node's current free buffer size, it means that the node will be congested; then the message acceptance selection mechanism adopts the *additive increase* method to slowly increase the acceptance threshold value to accept those messages that can bring more credits. When *messages changing number* is smaller than node's current free buffer size, the node will adopt the *multiplicative decrease* method to quickly decrease the acceptance threshold value to store as many messages as possible. The pseudocode of message acceptance selection mechanism is shown in Algorithm 1.

---

**Algorithm 1.** Message acceptance selection mechanism.

---

```

1: if a period  $T$  is time out then
2:   Calculate the new messages changing number by EWMA
3:   if the messages changing number is larger than the current free buffer size then
4:      $\text{Threshold} = \text{Threshold} + Ad$ 
5:   else
6:      $\text{Threshold} = \text{Threshold} \times Mul$ 
7:   end if
8: end if

```

---

In Algorithm 1,  $\text{Threshold}$  is the message acceptance threshold in definition 1,  $Ad$  is the message acceptance threshold increment, and  $Mul$  ( $0 < Mul < 1$ ) is a multiplicative factor. When  $Ad$  is small, the threshold will increase slowly and the node's buffer will be full quickly, so it cannot accept any message that can obtain more credits. On the other hand, when  $Ad$  is large, the threshold will

increase to a high value quickly, and the node may not accept enough messages due to the high threshold, so a proper increment  $Add$  is important for the design of a message acceptance selection mechanism. Same as  $Add$ , a node should also choose a proper multiplicative factor  $Mul$  for the message acceptance selection mechanism. Experimentally, we found that a smaller  $Add$  means a low increment; when coupled with a smaller  $Mul$ , which means a quick decrement, it can result in good performance. So we use  $Add = 100$  s,  $Mul = 0.3$ , and initial  $Threshold = 500$  s in this paper.

## 5. Performance Evaluation

### 5.1. Simulation Setup

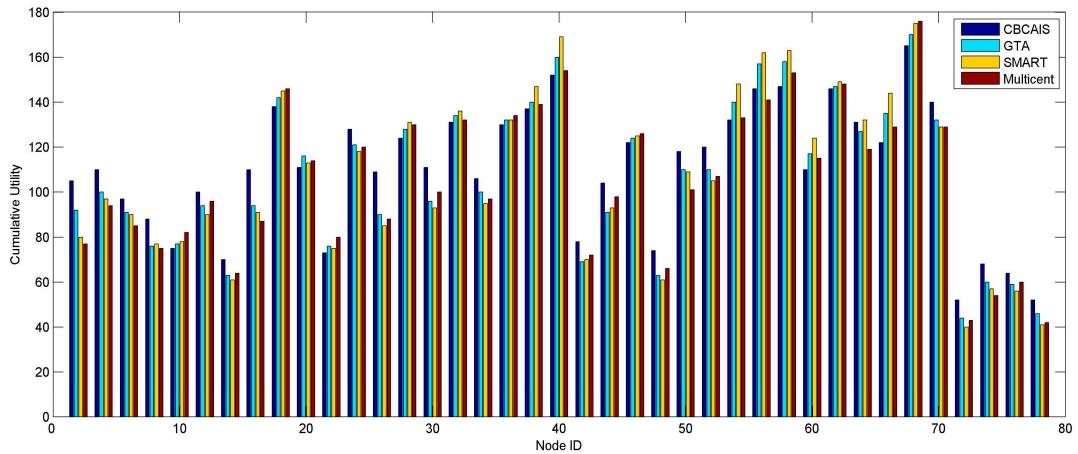
We implement CBCAIS on the Opportunistic Networking Environment simulator [25], and evaluate its performance in the data set of Infocom 06 [26]. Infocom 06 is a dataset collected during the Infocom 2006 student workshop. It contains 78 devices and covers approximately four days. The iMotes are programmed to log contacts of all visible mobile devices and a number of external devices that are not used in this paper. Every node has initial credits of 15, and pays one credit for each delivered message. The period  $T$  of calculating the *messages changing number* is 2500 s, and the  $\alpha$  in Equation (6) is 0.7. By default, the message generation interval is 5–10 s; the message size is 1 MB; the buffer size is 50 MB; the data transmission speed is 200 kBps; and the warm-up phase is 35,000 s. All nodes have the same initial energy, 10,000 J. The scan energy every time for scanning/discovering neighbors is 0.1 J. The transmission energy for a message is 1 J.

### 5.2. Performance Comparison

CBCAIS is compared with the direct routing scheme, where each node only stores its own messages but does not forward messages to others; the SMART scheme [14]; the Multicent scheme [15]; and the scheme proposed by Wu [16], which we denote as GTA. We are interested in the following metrics for performance evaluation: packet delivery ratio, average delivery delay, overhead ratio, average energy consumption, cumulative utility, and number of failed transmissions due to short of credits. The delivery ratio is defined as the ratio of successfully delivered messages to total generated messages in the network. Delay is measured by the delay sum of all delivered messages divided by the number of delivered messages. Overhead ratio is defined as the ratio of the total number of forwarded messages to the number of delivered messages, which denotes how many times forwarding was needed to successfully deliver a message. Average energy consumption is the average energy consumed by each node in the simulation process. Cumulative utility is the total credits obtained by a node, including the credits to pay for delivered messages. We evaluate the impact of buffer size, TTL, message generation interval, and data transmission speed on performance.

#### 5.2.1. Cumulative Utility Comparison

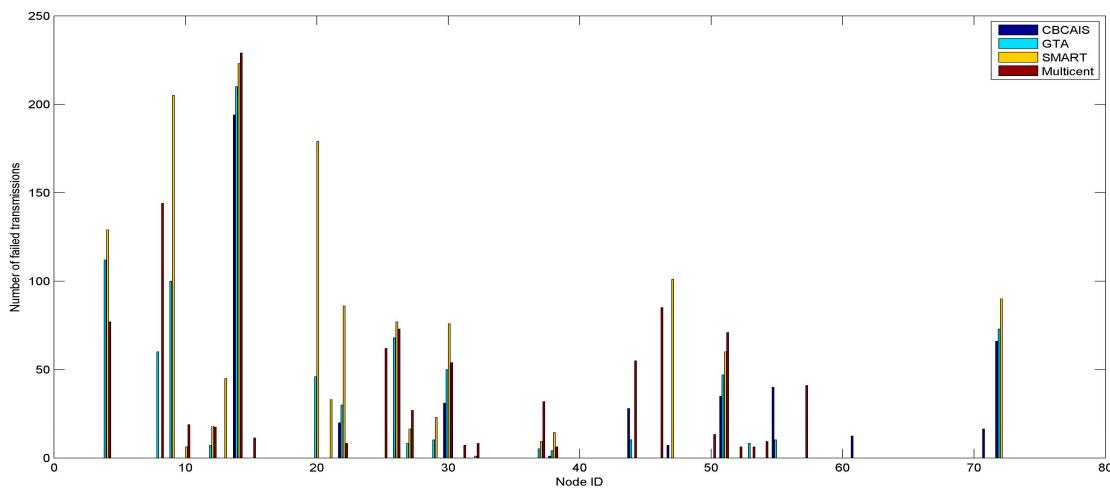
Figure 7 shows the distribution of cumulative utilities, where only even nodes are selected due to the space limit. We can see that in CBCAIS, most of the nodes can get more utilities than the SMART, Multicent, and GTA schemes. In the SMART, Multicent, and GTA schemes, many messages are forwarded to nodes with higher connection degree, and discarded due to node congestion. In SMART, the total credits paid for a message are divided by all the forwarding nodes. Therefore, all the forwarding nodes will lose their credits and get fewer utilities than CBCAIS when a message is discarded. In Multicent, the message destination node has to pay the forwarding nodes even when a message is discarded, so the destination node will get fewer utilities due to the discarding. In GTA, many messages are only sold to the high connection degree and congested node, therefore the nodes with more buffer and low connection degree will have no chance to get more credits. However, CBCAIS can effectively reduce the number of discarded messages by the check and punishment mechanism. In addition, by adjusting the message acceptance threshold according to congestion degree, a node can buffer the messages with more benefits. Therefore, most nodes in CBCAIS can get more utilities than other schemes.



**Figure 7.** Cumulative utility comparison.

### 5.2.2. Number of Failed Transmissions

From Figure 8, we can see that in SMART, Multicent, and GTA, about 20 nodes have some messages fail in transmission due to a shortage of credits. However, in CBCAIS, only about 10 nodes have some failed transmissions. In addition, the total number of failed transmissions due to a shortage of credits in CBCAIS is much smaller. In the SMART, Multicent, and GTA schemes, since most messages will be forwarded to the nodes with a high connection degree, many will be discarded. As a result, previous forwarding nodes or destination nodes will lose their credits. In addition, many nodes with larger delivery delay but free buffer will have no chance to earn credits, so these nodes will not have enough credits to pay for messages, leading to an increase in failed transmissions. In CBCAIS, the check and punishment mechanism can prevent nodes from discarding messages deliberately, so the nodes will have a better credit balance. Therefore, the number of failed transmissions will be small.

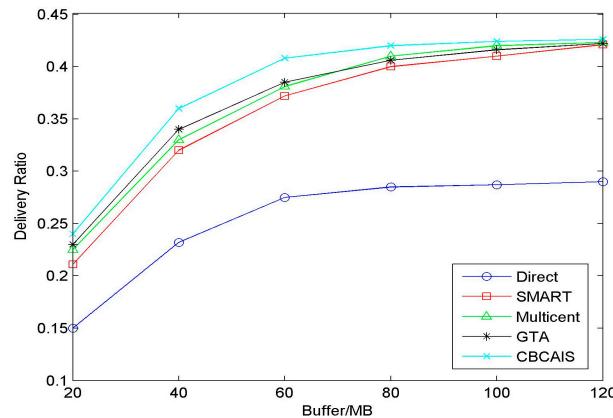


**Figure 8.** Number of failed transmissions.

### 5.2.3. Influence of Buffer Size

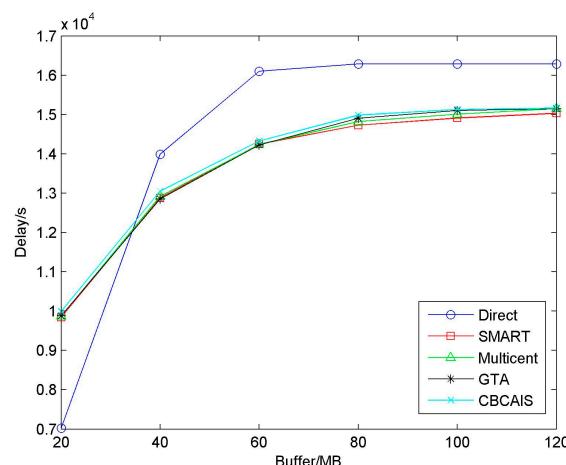
As shown in Figure 9, the delivery ratio increases as the buffer size increases for all the routing schemes. This is because nodes can carry more messages with larger buffer, and consequently deliver more messages successfully. The delivery ratio of the direct scheme is the lowest, because each node only buffers its own messages and cannot forward the messages to others with higher delivery probability. As a result, some messages are discarded due to the overflow of buffer or expiration of TTL. The delivery ratio of CBCAIS is much higher than SMART, Multicent, and GTA when the

buffer is small. This is because the SMART, Multicent, and GTA schemes only encourage nodes to deliver messages to other nodes with higher delivery probability, without considering the node congestion problem. Many messages will be discarded when a receiving node does not have free buffer. In addition, some messages fail in transmission due to a shortage of credits, as in Section 5.2.2. Since nodes are prevented from discarding messages deliberately, and have a good credit balance, the CBCAIS scheme has a higher delivery ratio.



**Figure 9.** Delivery ratio for varying buffer size.

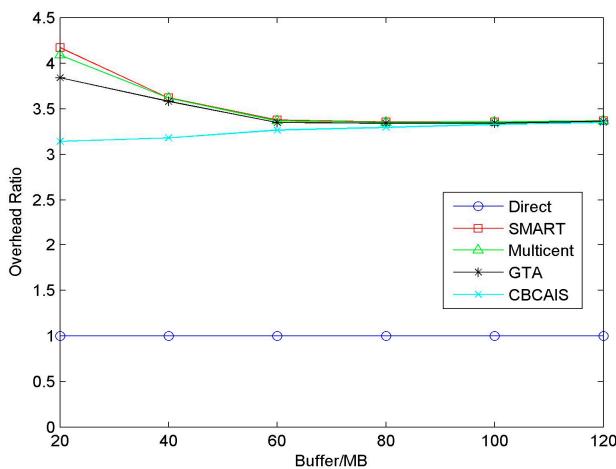
The impact of buffer size on delay is shown in Figure 10. It can be seen that delivery delay increases as buffer size increases for all schemes. This is because, when buffer size increases, nodes can deliver more messages with longer delay, leading to an increase in the average delivery delay.



**Figure 10.** Delivery delay for varying buffer size.

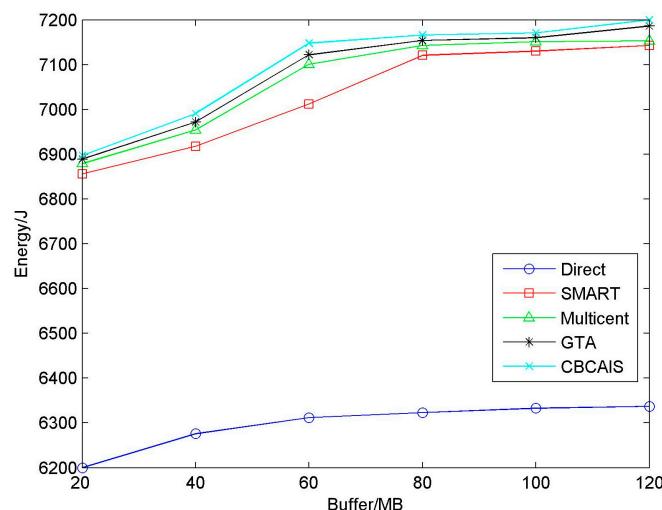
In the direct scheme, each node only stores its own messages and cannot forward the messages to others with smaller delivery delay, so the average delivery delay is largest when the buffer is larger than 20 M. When the buffer is small, the direct scheme has the smallest average delivery delay because only very few messages are delivered successfully, as shown in Figure 9, and many messages with longer delivery delay are discarded due to the overflow of buffer or expiration of TTL. The average delivery delay of CBCAIS scheme is slightly larger than SMART, Multicent, and GTA. One reason is that the delivery ratio is higher, as in Figure 9, and more messages with longer delay are delivered. Another reason is that a receiving node with small delivery delay will decline to forward messages when the buffer is full, and then these messages will be forwarded to other nodes with more free buffers but slightly larger delivery delay.

The impact of buffer size on overhead ratio is shown in Figure 11. In the direct scheme, each node only stores its own messages, which are never forwarded to others, so the overhead ratio is always 1. For other schemes, the impact of buffer size has two aspects. On the one hand, as buffer size increases, nodes get more forwarding chances, so the number of forwards will increase, leading to an increase in the overhead ratio. On the other hand, more messages are successfully delivered, leading to a decrease in the overhead ratio. When the first impact outweighs the second one, the overhead ratio increases; otherwise the overhead ratio decreases. The overhead ratios of SMART, Multicent, and GTA are higher than CBCAIS when the buffer is small. This is because many messages are forwarded to the nodes with high connection degree through many hops, but finally discarded due to node congestion. The invalid forwarding will increase the overhead ratio.



**Figure 11.** Overhead ratio for varying buffer size.

As shown in Figure 12, the energy increases as the buffer size increases for all the routing schemes. Nodes can store more messages with larger buffer, so the number of message forwards will increase, leading to more energy consumption. The energy consumption of the direct scheme is the lowest because each node only buffers its own messages and cannot forward the messages to others. In CBCAIS, more messages are successfully delivered, so the energy consumption of CBCAIS is highest.

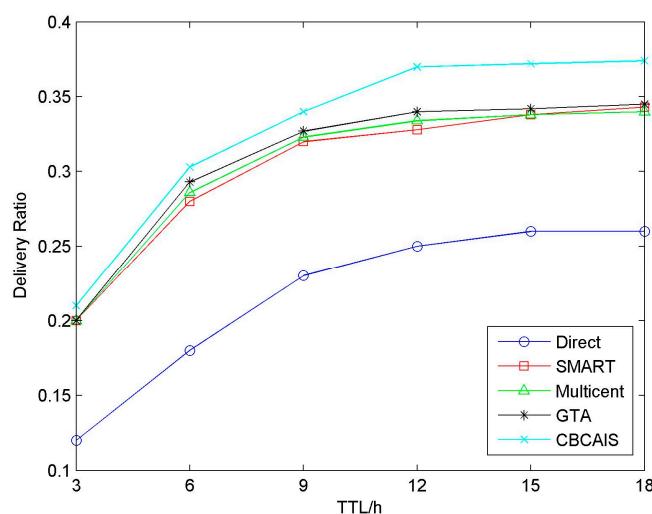


**Figure 12.** Energy for varying buffer size.

From the above, we know that CBCAIS can achieve a higher message delivery ratio with lower overhead ratio compared to SMART, Multicent, and GTA, although it has a slightly larger average delay and energy consumption.

#### 5.2.4. Influence of TTL

The impact of TTL on delivery ratio is shown in Figure 13. It can be seen that the delivery ratio increases as the TTL increases for all schemes. This is because the messages with larger TTL can stay longer in the buffer and the delivery chances are better. The delivery ratio of the direct scheme is the lowest. This is because each node only stores its own messages and cannot forward messages to other nodes with higher delivery probability. As a result, some messages are discarded due to the overflow of buffer or expiration of TTL. The delivery ratio of the CBCAIS scheme is much higher than SMART, Multicent, or GTA when TTL is larger. This is because nodes are more congested when TTL is larger, and CBCAIS can effectively avoid node congestion.

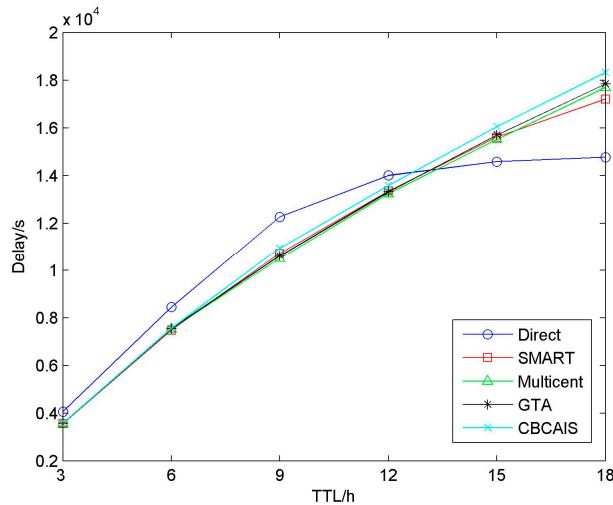
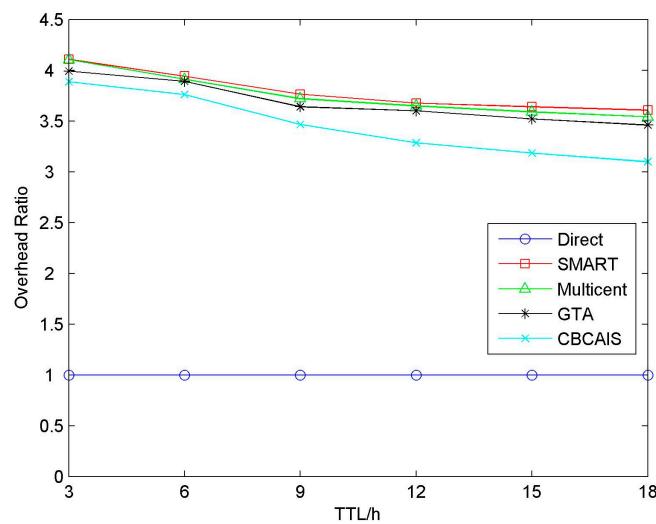
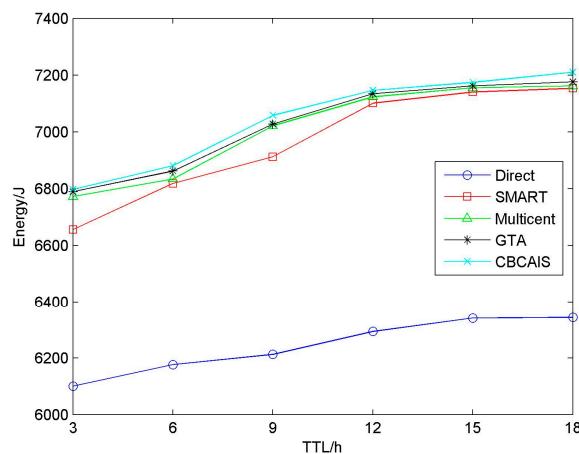


**Figure 13.** Delivery ratio for varying TTL.

Figure 14 shows the impact of TTL on delay. It can be seen that delivery delay increases as TTL increases for all schemes. This is because, as the TTL increases, nodes can store and deliver more messages with longer delay. The delivery delay is largest in the direct scheme when TTL is no larger than 12 h, because a node cannot forward messages to others with smaller delivery delay. The delivery delays of other schemes are larger when TTL is larger than 12 h, because more messages with longer delay are delivered. The delivery delay of CBCAIS is slightly larger than SMART, Multicent, and GTA. This is because more messages with longer delivery delay are delivered, and many messages are forwarded to the forwarding nodes with more free buffer but a slightly larger delivery delay to avoid being discarded.

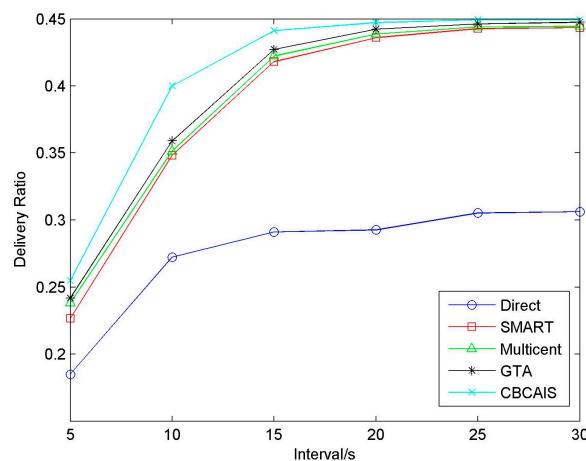
Figure 15 shows the impact of TTL on overhead ratio. In the direct scheme, the overhead ratio is always 1, because the messages are not forwarded by other nodes. For other routing schemes, more messages are successfully delivered as TTL increases, so the overhead ratio will decrease. The overhead ratio of CBCAIS is lower than SMART, Multicent, and GTA, as it can reduce message discarding and avoid many invalid forwards.

The impact of TTL on energy is shown in Figure 16. It can be seen that energy increases as TTL increases for all schemes. This is because messages have more chances to be forwarded with longer TTL, leading to more energy consumption. In the direct scheme, the energy consumption is lowest because the messages are not forwarded by other nodes. The energy consumption of CBCAIS is slightly higher than SMART, Multicent, and GTA, as it can deliver messages more effectively.

**Figure 14.** Delivery delay for varying TTL.**Figure 15.** Overhead ratio for varying TTL.**Figure 16.** Energy for varying TTL.

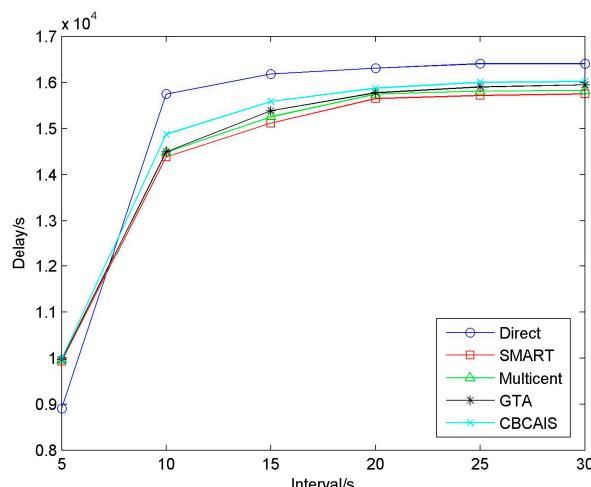
### 5.2.5. Influence of Message Generation Interval

As shown in Figure 17, it can be seen that the delivery ratio increases with the increase of message generation interval. When the message generation interval is larger, the total messages generated in the network will be smaller, and nodes can have enough buffer and bandwidth to forward messages, so the delivery ratio is higher. In the direct scheme, a node only forwards its own messages, so the delivery ratio is lowest. The delivery ratio of CBCAIS is much higher than SMART, Multicent, and GTA when message generation interval is small. This is because when the message generation interval is small, more nodes will be congested, and the CBCAIS scheme can effectively reduce the impact of congestion.



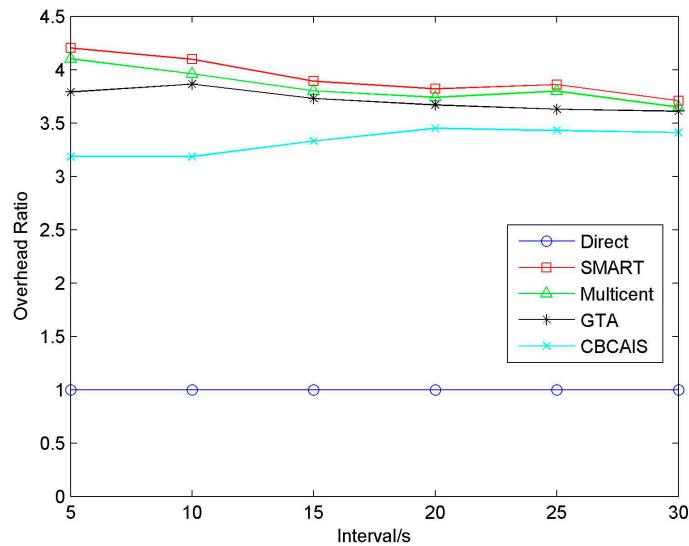
**Figure 17.** Delivery ratio for varying message generation interval.

Figure 18 shows the impact of message generation interval on delay. It can be seen that delivery delay increases for all schemes as the message generation interval increases. When the message generation interval is small, more messages are generated in the network and many messages with longer delay will be dropped due to the limited buffer. Because the messages with longer delay can be forwarded when the message generation interval is larger, the delivery delay will increase as the message generation interval increases. The average delivery delay of CBCAIS is slightly larger than SMART, Multicent, and GTA. This is because more messages are forwarded to nodes with more free buffer but a slightly larger delivery delay to avoid being dropped.



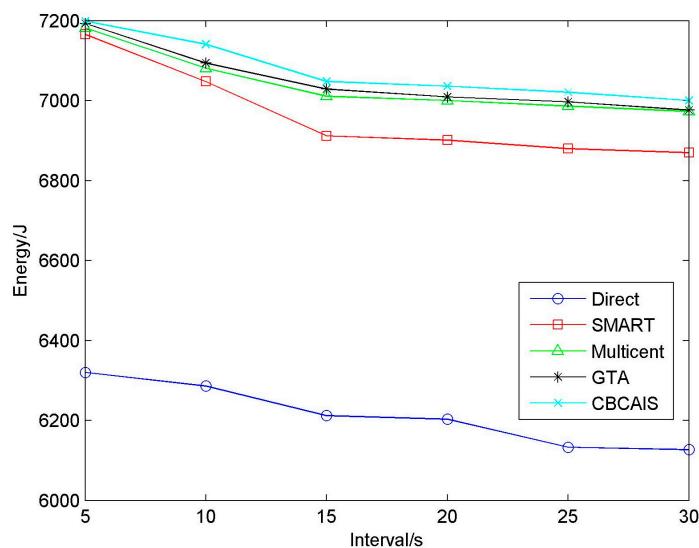
**Figure 18.** Delivery delay for varying message generation intervals.

As shown in Figure 19, the overhead ratio of the direct scheme is always 1. For the other schemes, the impact has two aspects. On the one hand, as message generation interval increases, nodes can have enough bandwidth to forward messages, so the number of forwards will increase, leading to an increase in the overhead ratio. On the other hand, nodes will also have enough buffers to deliver more messages, leading to a decrease in the overhead ratio. When the first impact outweighs the second one, the overhead ratio increases, otherwise the overhead ratio decreases. The overhead ratio of CBCAIS is lower than SMART, Multicent, and GTA schemes.



**Figure 19.** Overhead ratio for varying message generation interval.

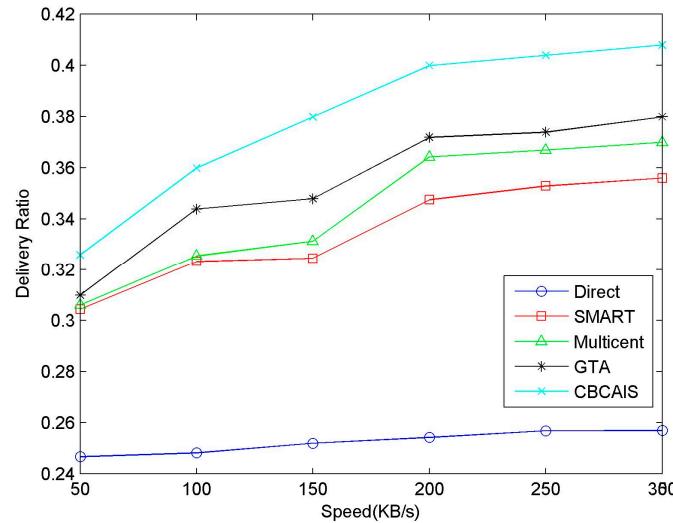
Figure 20 shows the impact of message generation interval on energy. It can be seen that the energy decreases for all schemes as the message generation interval increases. This is because when the message generation interval is larger, the total messages generated in the network will be fewer, which leads to less energy consumption. Because of the effective message forwarding, the energy consumption of CBCAIS is slightly higher than in the SMART, Multicent, and GTA schemes. The energy consumption of the direct scheme is the lowest.



**Figure 20.** Energy for varying message generation intervals.

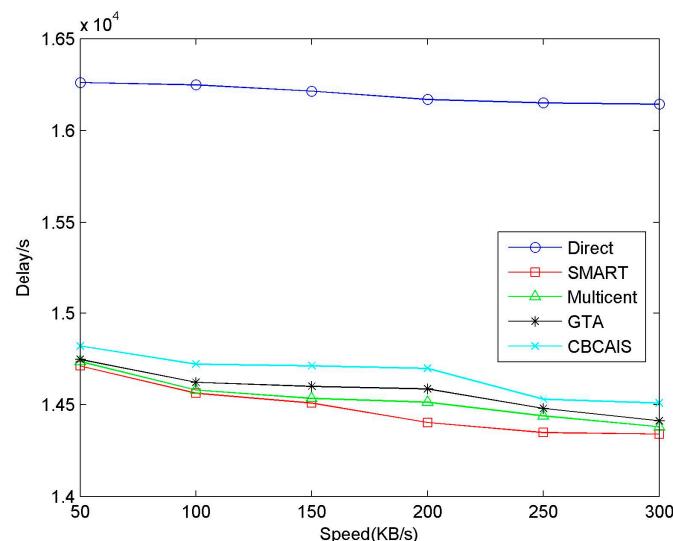
### 5.2.6. Influence of Data Transmission Speed

The impact of data transmission speed on delivery ratio is shown in Figure 21. It can be seen that the delivery ratio increases as the transmission speed increases for all schemes. This is because nodes can forward more messages with larger data transmission speed during the limited contact time. The delivery ratio of CBCAIS scheme is highest, because of the effective message forwarding. The delivery ratio of the direct scheme is the lowest.



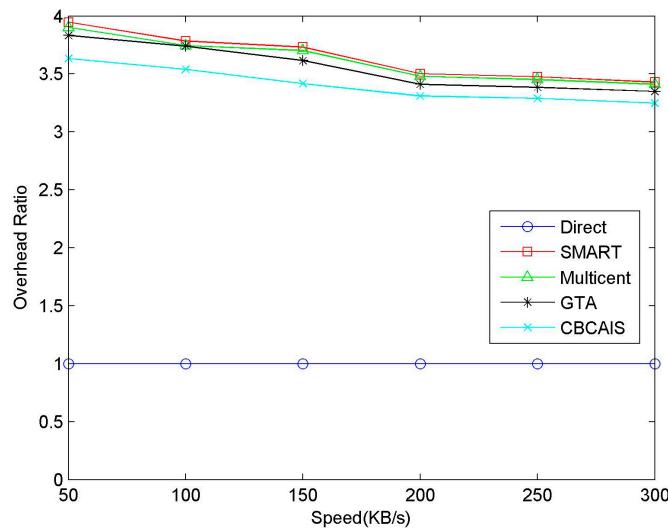
**Figure 21.** Delivery ratio for varying transmission speeds.

The impact of data transmission speed on delay is shown in Figure 22. It can be seen that the delay decreases as the transmission speed increases for all schemes. This is because messages can be forwarded quickly with large data transmission speed, which leads to a reduction in the average delay. The delay in the CBCAIS scheme is slightly longer than in the SMART, Multicent, and GTA schemes, because more messages with longer delay are delivered. The delay in the direct scheme is the smallest because each node only buffers its own message, and cannot forward messages to others with smaller delivery delay.



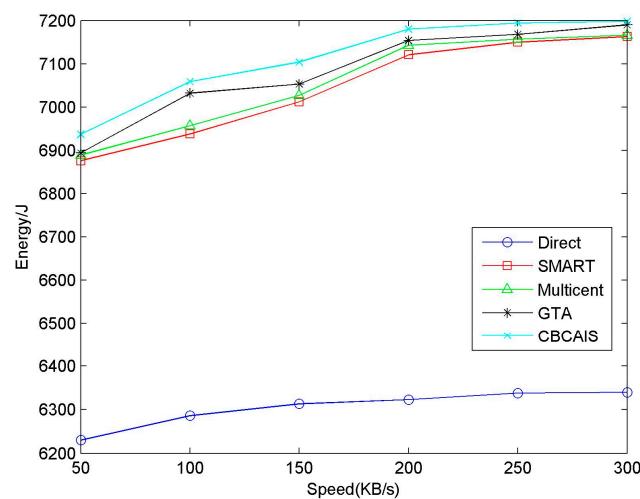
**Figure 22.** Delay with varying transmission speeds.

The impact of data transmission speed on overhead ratio is shown in Figure 23. It can be seen that the overhead ratio of the direct scheme is always 1; for other schemes, the overhead ratio decreases as the transmission speed increases. This is because more messages can be successfully forwarded with larger data transmission speed. The overhead ratio of CBCAIS is lower than in the SMART, Multicent, and GTA schemes.



**Figure 23.** Overhead ratio for varying transmission speeds.

The impact of data transmission speed on energy is shown in Figure 24. It can be seen that the energy increases as the transmission speed increases for all schemes. This is because more messages are successfully forwarded with larger data transmission speed, which will consume more energy. The energy consumption of CBCAIS is slightly higher than in the SMART, Multicent, and GTA schemes. The energy consumption of the direct scheme is lowest because the messages are not forwarded by other nodes.



**Figure 24.** Energy for varying transmission speeds.

## 6. Conclusions

In this paper, we have proposed a credit-based congestion-aware incentive scheme, called CBCAIS, to stimulate selfish nodes to cooperatively forward messages in DTNs. In CBCAIS, a delay-based credit reward scheme is proposed to effectively stimulate selfish nodes to forward messages to

others according to delivery delay. In addition, a check and punishment mechanism is proposed to prevent forwarding nodes from discarding message deliberately. To further enhance the performance, a message acceptance selection mechanism is proposed to allow nodes to decide whether to accept the messages of other nodes, according to self congestion degree. We have also verified the effectiveness of CBCAIS through extensive simulations. This work focuses on single-copy routing; our future direction is to explore an effective congestion-aware incentive scheme for multi-copy routing and data dissemination in DTNs.

**Acknowledgments:** This work was fully supported by Natural Science Foundation of Heilongjiang Province (e6838755-7-11014), the Science and Technology Research Projects of the Education Department of Heilongjiang Province (12543005) and the Natural Science Foundation of Daqing Normal University (14zr21).

**Author Contributions:** Qingfeng Jiang constructed the model and the algorithms and wrote the manuscript; and Zeyu Tian designed the experiment. Chaoguang Men organized and refined the manuscript. All authors have read and approved the final manuscript.

**Conflicts of Interest:** The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

1. Fall, K. A delay-tolerant network architecture for challenged Internets. In Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'03), Karlsruhe, Germany, 25–29 August 2003; pp. 27–34.
2. Jain, S.; Fall, K.; Patra, R. Routing in a delay tolerant network. In Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'04), Portland, OR, USA, 30 August–3 September 2004; pp. 145–158.
3. Lindgren, A.; Doria, A.; Schelen, O. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2003**, *7*, 19–20. [[CrossRef](#)]
4. Erramilli, V.; Crovella, M.; Chaintreau, A.; Diot, C. Delegation forwarding. In Proceedings of the MobiHoc'08, Hong Kong, China, 26–30 May 2008; pp. 251–259.
5. Hui, P.; Crowcroft, J.; Yoneki, E. Bubble Rap: Social-based forwarding in delay-tolerant networks. *IEEE Trans. Mob. Comput.* **2011**, *10*, 1576–1589. [[CrossRef](#)]
6. Costa, P.; Mascolo, C.; Musolesi, M.; Picco, G.P. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 748–760. [[CrossRef](#)]
7. Burgess, J.; Gallagher, B.; Jensen, D. MaxProp: Routing for vehicle-based disruption-tolerant networking. In Proceedings of the INFOCOM 2006, Barcelona, Spain, 23–29 April 2006; pp. 1–11.
8. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Efficient routing in intermittently connected mobile networks: The multiple-copy case. *IEEE/ACM Trans. Netw.* **2008**, *16*, 77–90. [[CrossRef](#)]
9. Li, Y.; Su, G.L.; Wu, D.P.O.; Jin, D.P. The impact of node selfishness on multicasting in delay tolerant networks. *IEEE Trans. Veh. Technol.* **2011**, *60*, 2224–2238. [[CrossRef](#)]
10. Shevade, U.; Song, H.H.; Qiu, L.L.; Zhang, Y. Incentive-aware routing in DTNs. In Proceedings of the IEEE International Conference on Network Protocols (ICNP), Orlando, FL, USA, 19–22 October 2008; pp. 238–247.
11. Karaliopoulos, M. Assessing the vulnerability of DTN data relaying schemes to node selfishness. *IEEE Commun. Lett.* **2009**, *13*, 923–925. [[CrossRef](#)]
12. Wang, Y.; Chuah, M.C.; Chen, Y.Y. Incentive based data sharing in delay tolerant mobile networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 370–381. [[CrossRef](#)]
13. Chen, B.B.; Chan, M.C. MobiCent: A credit-based incentive system for disruption tolerant network. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM) 2010, San Diego, CA, USA, 15–19 March 2010; pp. 1–9.
14. Zhu, H.J.; Lin, X.D.; Lu, R.X.; Fan, Y.F.; Shen, X.M. SMART: A secure multi-layer credit based incentive scheme for delay-tolerant networks. *IEEE Trans. Veh. Technol.* **2009**, *58*, 4628–4639.
15. Chen, K.; Shen, H.Y. Multicent: A multifunctional incentive scheme adaptive to diverse performance objectives for DTN routing. In Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON), New Orleans, LA, USA, 24–27 June 2013; pp. 532–540.

16. Wu, F.; Chen, T.; Zhong, S.; Qiao, C.; Chen, G. A game-theoretic approach to stimulate cooperation for probabilistic routing in opportunistic networks. *IEEE Trans. Wirel. Commun.* **2012**, *12*, 1573–1583. [[CrossRef](#)]
17. Pujol, J.; Toledo, A.; Rodriguez, P. Fair routing in delay tolerant networks. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 837–845.
18. Zhang, X.; Wang, X.F.; Liu, A.; Zhang, Q.; Tang, C.J. PRI: A practical reputation-based incentive scheme for delay tolerant networks. *KSII Trans. Internet Inf. Syst.* **2012**, *6*, 973–988. [[CrossRef](#)]
19. Wei, L.F.; Zhu, H.J.; Cao, Z.F.; Shen, X.M. SUCCESS: A secure user-centric and social-aware reputation based incentive scheme for DTNs. *Ad Hoc Wirel. Sens. Netw.* **2013**, *19*, 95–118.
20. Li, J.; Wang, X.W.; Yu, R.Y.; Liu, R. Reputation-based incentives for data dissemination in mobile participatory sensing networks. *Int. J. Distrib. Sens. Netw.* **2015**, *2015*, 1–13. [[CrossRef](#)]
21. Buttyan, L.; Dora, L.; Felegyhazi, M.; Vajda, I. Barter trade improves message delivery in opportunistic networks. *Ad Hoc Netw.* **2010**, *8*, 1–14. [[CrossRef](#)]
22. Ning, T.; Yang, Z.P.; Wu, H.Y.; Han, Z. Self-interest-drive incentives for ad dissemination in autonomous mobile social networks. In Proceedings of the IEEE INFOCOM 2013, Turin, Italy, 14–19 April 2013; pp. 2310–2318.
23. Gao, W.; Zhao, Q.; Li, B.; Cao, G.H. Multicasting in delay tolerant networks: A social network perspective. In Proceedings of the ACM MobiHoc'09, New Orleans, LA, USA, 18–21 May 2009; pp. 299–308.
24. Krifa, A.; Barakat, C.; Spyropoulos, T. Optimal buffer management policies for delay tolerant networks. In Proceedings of the Fifth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08), San Francisco, CA, USA, 16–20 June 2008; pp. 260–268.
25. Keranen, A.; Ott, J.; Karkkainen, T. The ONE simulator for DTN protocol evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, 2–9 March 2009; pp. 1–10.
26. CRAWDAD Trace Infocom06. Available online: <http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote> (accessed on 20 May 2016).



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).