View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Crossref

SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks

Haojin Zhu, *Member, IEEE*, Xiaodong Lin, *Member, IEEE*, Rongxing Lu, *Student Member, IEEE*, Yanfei Fan, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Delay-tolerant networks (DTNs) provide a promising solution to support wide-ranging applications in the regions where end-to-end network connectivity is not available. In DTNs, the intermediate nodes on a communication path are expected to store, carry, and forward the in-transit messages (or bundles) in an opportunistic way, which is called opportunistic data forwarding. Such a forwarding method depends on the hypothesis that each individual node is ready to forward packets for others. This assumption, however, might easily be violated due to the existence of selfish or even malicious nodes, which may be unwilling to waste their precious wireless resources to serve as bundle relays. To address this problem, we propose a secure multilayer credit-based incentive scheme to stimulate bundle forwarding cooperation among DTN nodes. The proposed scheme can be implemented in a fully distributed manner to thwart various attacks without relying on any tamperproof hardware. In addition, we introduce several efficiency optimization techniques to improve the overall efficiency by exploiting the unique characteristics of DTNs. Extensive simulations demonstrate the efficacy and efficiency of the proposed scheme.**

***Index Terms*—Delay-tolerant networks (DTNs), incentive scheme, security.**

I. Introduction

Most popular Internet applications rely on the existence of a contemporaneous end-to-end link between the source and the destination, with moderate round-trip times and small packet loss probabilities. This fundamental assumption is not expected in some challenged networks, which are often referred to as delay-tolerant networks (DTNs). Applications of this emergent communication paradigm are wide ranging and include low-cost Internet service provision in remote or developing localities [1], vehicular DTNs for dissemination of location-dependent information (e.g., local ads, traffic reports, and parking information) [2], [3] or for providing multihop Internet access [4], social-based networks to allow humans to communicate without network infrastructure [5], [6], pocket-switched networks [7], underwater networks [8], etc. In DTNs, the in-transit messages, which are also called bundles, can be sent over an existing link and buffered at the next hop until the next link in the path appears (e.g., a new node moves into the range or an existing node wakes up). This message propagation process is usually referred to as the "store-carry-and-forward" strategy, and routing is made in an "opportunistic" fashion.

Previously reported studies have focused on opportunistic data propagation in DTNs [9]–[13], which depends on the hypothesis that each individual node is ready to forward packets for others. This hypothesis, however, might easily be violated in the presence of selfish or even malicious nodes, which may choose to save their precious wireless resources by refusing to serve as bundle relays [14]. Such selfishness actions may be more challenging for researchers in certain applications of DTNs such as vehicular DTNs and social networks, which are decentralized and distributed over a multitude of devices that are controlled and operated by individuals. In these applications, it is highly possible that there exist some selfish users who may not want to forward such bundles without compensation. Furthermore, even from the security point of view, naive packet forwarding may open a new door for malicious users, who may intentionally try to launch denial-of-service attacks on the network by flooding the network with dummy messages. Thus, to deploy an applicable DTN in real-world scenarios, proper incentives and security mechanisms should be in place.

One of the most promising ways to address the selfishness issue and stimulate cooperation among selfish nodes in DTNs is using *incentive schemes*, which basically fall into two categories, i.e., reputation- and credit-based schemes. Reputation-based schemes rely on individual nodes to monitor neighboring nodes' traffic and keep track of each others' reputation so that uncooperative nodes are eventually detected and excluded from the networks [14]–[17], whereas credit-based schemes introduce some form of virtual currency to regulate the packet-forwarding relationships among different nodes [18]–[20]. The previously reported incentive schemes, which were proposed for conventional mobile ad hoc networks, may not be suitable for DTNs, for the following two reasons: First, a common assumption adopted in existing incentive schemes is that a full end-to-end path between the source and the destination can be determined before data forwarding occurs. This assumption does not hold in DTNs due to its intrinsic opportunistic forwarding nature. Second, the reported schemes are mainly designed for single-copy forwarding. However, multicopy forwarding or even flooding is often adopted to enhance the

Manuscript received July 26, 2008; revised January 31, 2009. First published April 3, 2009; current version published October 2, 2009. This work was supported by a joint grant from the Natural Sciences and Engineering Research Council and Research In Motion, Canada. The review of this paper was coordinated by Prof. A. Boukerche.

H. Zhu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200030, China (e-mail: zhu-hj@cs.sjtu.edu.cn).

X. Lin is with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON L1H 7K4, Canada (e-mail: Xiaodong.Lin@uoit.ca).

R. Lu, Y. Fan, and X. Shen are with the Center for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: h9zhu@bbcr.uwaterloo.ca; rxlu@bbcr.uwaterloo.ca; yfan@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2009.2020105

0018-9545/$26.00 © 2009 IEEE

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

reliability of DTN communication [9], which makes most existing incentive schemes incompatible with diverse DTN routing.

In this paper, we propose a secure multilayer credit-based incentive (SMART) scheme for DTNs afflicted with selfish nodes. Similar to other credit-based incentive schemes, SMART uses credits to provide incentives to selfish nodes. One of its novel and distinguishing features is that it allows the credit to be transferred/distributed by the current intermediate node without the involvement of the sender. Such a design is well suited for DTNs since, in DTNs, the bundle sender cannot predict the bundle forwarding path, and intermediate nodes may also suffer from delayed or frequent loss of connectivity to the bundle sender.

In specific, SMART is based on the notion of a *layered coin* that provides virtual electronic credits to charge for and reward the provision of data forwarding in DTNs. Such a coin is composed of multiple layers, each of which is generated by the source/destination or an intermediate node. The first layer, which is also named the *base layer*, is generated by the source to indicate the payment rate (credit value), remuneration conditions, the class-of-service (CoS) requirement, and other reward policies. During the subsequent bundle propagation process, each intermediate node will generate a new layer based on the previous layers by appending a nonforgeable digital signature. This new layer is also called the *endorsed layer*, which implies that the forwarding node agrees to provide forwarding service under the predefined CoS requirement and will be rewarded according to the reward policy in the future. With endorsed layers, it is easy to track the propagation path and determine each intermediate node by checking the signature of each endorsed layer. In the rewarding and charging phase, if the provided forwarding service satisfies remuneration conditions defined in the predefined reward policy, each forwarding node along one or multiple path(s) will share the credit defined in this coin depending on different data-forwarding algorithms (single-copy/multicopying forwarding) and the actual forwarding results (bundle delivered along one or multiple paths).

However, the main challenge in designing SMART is to ensure that the security properties of the scheme are not compromised. Since all security related to a coin, particularly during the store-carry-and-forward process, is managed by the intermediate nodes, a selfish node (or even a group of colluding nodes) may attempt to cheat the system to maximize its expected welfare. As an example, a selfish node may arbitrarily inject a fake layer into the current coin or remove several valid layers from it, if such actions can maximize its welfare. This is the security perspective of SMART. Second, any security functionality will incur extra computation and transmission overhead. A secure credit-based incentive scheme should be efficient enough to not significantly compromise the system performance. This is the performance perspective of our system.

The contributions of this paper can be summarized as follows: First, we propose a SMART scheme to stimulate cooperation among selfish nodes in DTNs. The proposed scheme can be made compatible with diverse data-forwarding algorithms in DTNs. Second, SMART can withstand a wide range of cheating actions because of its novel *layer concatenation* technique. Third, we propose two performance optimization techniques to minimize the computation and transmission overhead. Furthermore, SMART is a one-way noninteractive protocol, which is particularly suitable for DTNs, where interactive communication suffers from long round-trip delays and frequent disconnection [1]. Finally, extensive simulations are conducted to demonstrate the efficiency and effectiveness of the proposed SMART scheme.

The remainder of this paper is organized as follows: Section II provides a comprehensive overview of related work. In Section III, we present the system model, the node model, and the design goals. In Section IV, we first give an overview of the SMART scheme and then present SMART in detail, as well as two performance optimization methods. Performance evaluation is given in Section V, followed by a discussion and the conclusion in Sections VI and VII, respectively.

II. Related Work

There is a large amount of literature on incentive mechanisms for different kinds of networks. These reported mechanisms basically fall into two categories, i.e., reputation- and credit-based schemes.

Reputation-based schemes rely on the individual nodes to monitor neighboring nodes' traffic and keep track of each others' reputation so that uncooperative nodes can eventually be detected and excluded from the networks [14]–[17]. However, in DTNs, existing reputation-based incentive schemes may face the challenge of indistinguishability of sending and not sending a message, since data forwarding cannot be observed during the store-carry-and-forward process. Furthermore, it is also challenging to efficiently and effectively propagate the reputation.

Credit-based incentive schemes introduce some form of virtual currency to regulate the packet-forwarding relationships among different nodes. There are two different ways to realize such kind of credits: 1) game-theory-based schemes and 2) security-protocol-based schemes. The first approach tries to investigate such noncooperative communication scenarios within a game theory framework [21], [24], [25], whereas the second approach focuses on ensuring the security of the credits by using various cryptographic tools [18], [20]. Most of these schemes always assume that an end-to-end path exists and is determined before the data-forwarding process. However, this assumption obviously does not hold in DTNs, which makes them not suitable in DTNs. In [19], a virtual-cash-based incentive scheme is proposed to stimulate commercial advertisement dissemination in vehicular networks. In [6], it is suggested to use a multilevel coupon-based scheme to stimulate exchanging information about places of interest or local restaurants. However, in both schemes, the focus is on how to stimulate advertisement dissemination, and transmission is based on simple broadcasting, whereas DTN routing is not taken into consideration.

We incorporate a secure credit-based incentive scheme into DTN data routing/forwarding, which distinguishes SMART from previous work. The existing routing or data-forwarding schemes in DTN can be categorized into single- and multicopy schemes. Some protocols (e.g., First Contact [11] and Direct

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

Transmission/Delivery [10]) only generate a single copy, others enable the source to limit the forwarding copies to a fixed number [9], whereas epidemic [12] and probabilistic routing [13] potentially create an "infinite" number of messages. A latest study shows that, although single-copy schemes can considerably reduce resource waste, they are often orders of magnitude slower than multicopy algorithms and are inherently less reliable [9]. Therefore, in this paper, we consider a generalized multicopy data-forwarding scheme as the foundation, and therefore, our SMART scheme can be made compatible with diverse multicopy data-forwarding schemes.

III. System Model and Design Goals

This section describes our system model and design goals.

A. Network Model

We consider a general DTN formed by a set of mobile devices owned by individual users. Each node i is assumed to have a unique nonzero identifier $\mathcal{N}_i$, which is bound to a specific public key certificate. We interchangeably use node i and $\mathcal{N}_i$ hereafter. We also assume that each node has limited transmission and reception capabilities so that two nodes outside the transmission range of each other can communicate only via a sequence of intermediate nodes in a multihop manner. End-to-end connections are not always guaranteed, and routing, therefore, is made in an "opportunistic" way. Similar to other credit-based schemes such as [19] and [24], we assume that there exists in our scheme an Offline Security Manager (OSM), which is responsible for key distribution, and a virtual bank (VB), which takes charge of credit clearance. In many DTN application scenarios, there exist some special network components that can serve as the VB, such as the roadside unit in vehicular DTNs [19] and the information publisher in social networks [6]. The DTN nodes can exploit opportunistic links to these network components to submit collected coins to the VB. Before joining the DTN network, every DTN node should be registered with the OSM and obtain its public key certificate. At the clearance phase, the DTN nodes submit the collected layered coins to the VB for receiving their rewards.

B. Data-Forwarding Strategy

In this paper, we consider a generalized multicopy data-forwarding architecture: As shown in Fig. 1, for every bundle B originating from the source node $\mathcal{S}$, L_1 copies of B are initially spread by the source, and then, at every subsequent forwarding node $\mathcal{N}_i$, L_i message copies will opportunistically be propagated to the next hops. It is worth pointing out that existing DTN routing schemes can be treated as special cases of this routing model. For a single-copy-based forwarding scheme [10], [11], we can choose $\{L_i = 1 | i = 1, 2, \ldots, m\}$, where m is the total hop number of these forwarding paths. For epidemic and probabilistic routing [12], [13], $\{L_i | i = 1, 2, \ldots, m\}$ can be chosen as a specific large number. On the other hand, if a spray and waiting routing scheme is chosen as the basic data-forwarding scheme [9], we can assume $\{L_1 = L, L_i = 1 | i = 2, \ldots, m\}$, where L is the chosen forwarding copy number.

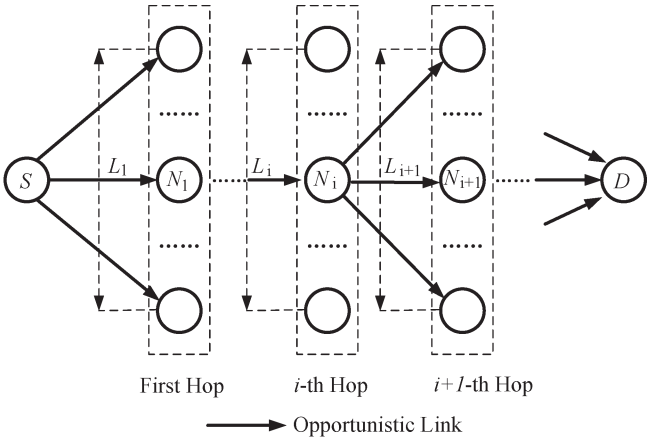


Fig. 1. Generalized data-forwarding strategy.

C. Rewarding Model

There are several available rewarding models that can be adopted in SMART. For example, a popular charging method in [18] is paying per packet, which means that, for each successfully transmitted unit-sized packet, each of N intermediate nodes should receive λ credits, whereas the source needs to pay $\lambda * N$ in total. However, we argue that this method is not suitable for opportunistic data forwarding in that it is difficult for the source to predict how many copies or hops are needed to successfully deliver a message to the destination. Therefore, in this paper, we consider a profit-sharing model, which means that the intermediate nodes involved in a successful bundle delivery will be paid with a dividend of the total credit provided by the source node. The source node can also specify a diverse case-by-case basis rewarding requirements in the base layer of a layered coin, which can be regarded as a part of the DTN routing policy [26]. For example, a bundle should successfully be delivered within a particular time-to-live (TTL) period or only the intermediate nodes along the first successful delivery path can be remunerated. The study on the rewarding policy is still an open problem and therefore deserves more investigation in future incentive-related research.

D. Attack Model

Due to the selfish nature, mobile nodes will try to cheat the system to maximize their welfare. In particular, a selfish node can exhibit one of three selfish actions.

1) *Credit forgery attack (or layer injection attack).* A selfish node may attempt to forge a valid credit (e.g., collude with other nodes to inject nonexistent layers into a valid layered coin) to reward itself for work it did not do or for more than it has done.
2) *Nodular tontine attack (or layer removal attack).* Unlike in a layer injection attack, when receiving a multilevel credit, a selfish node may try to remove one or several existing layers that have been generated by the previous forwarding nodes. This attack is particularly effective in profit-sharing systems, where the profits of the removed nodes will be shared by the remaining nodes. In this

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

sense, it is similar to a *tontine system*,[1] in which participants share a common fund and have been known to try to kill each other off, thereby increasing their shares. Therefore, we denote this kind of attack as a nodular tontine attack.

3) *Submission refusal attack.* In DTNs, due to the lack of end-to-end connection, a source node and other intermediate nodes may not have a clear idea about the forwarding progress, and thus, it relies on the last forwarding node to submit the generated layered coins to a VB for clearance. However, if colluding with the source node, the last intermediate node may refuse to submit the received credits and receive behind-the-scene compensation from the source node.

Note that any of the foregoing selfish actions can further be complicated by the collusion of two or more nodes. In this paper, we only consider each selfish node with a unique identity, as well as a corresponding public key certificate. Similar to [25], in this paper, we consider a general DTN network and thus assume that no "extra communications" exist among the DTN nodes.

E. Design Goals

The design goals have four characteristics.

1) *Effectiveness.* The proposed scheme should be effective in stimulating cooperation among the selfish nodes.
2) *Security.* It should be secure and robust from various attacks.
3) *Efficiency.* It should efficiently work without introducing much extra communication and transmission overhead.
4) *Generality.* It should be compatible with the most popular DTN routing schemes.

IV. Proposed SMART Scheme

In this section, we first provide some preliminary background, which is the design foundation of SMART. Then, we give an overview of the SMART scheme, followed by a detailed presentation of SMART. Finally, we introduce two efficient performance-enhancement methods.

A. Pairing Technique

SMART is based on bilinear pairing, which will be briefly introduced in the succeeding discussion. Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order q, i.e., $|\mathbb{G}| = |\mathbb{G}_T| = q$. Let P be a generator of $\mathbb{G}$. We further assume that $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is an efficient admissible bilinear map with three properties.

1) *Bilinear.* For $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab}$.
2) *Nondegenerate.* $\hat{e}(P, P) \neq 1_{\mathbb{G}_T}$.
3) *Computable.* There is an efficient algorithm to compute $\hat{e}(P_1, Q_1)$ for any $P_1, Q_1 \in \mathbb{G}$.

According to [27], such an admissible bilinear map $\hat{e}$ can be constructed by Weil or Tate pairing on the elliptic curves.

B. Overview of SMART

Before presenting our SMART scheme, we first introduce a naive multilayer coin scheme. In such a naive scheme, the data-forwarding process can also be regarded as a layered coin generation process. When a node sends its own messages, the node will lose credit (or virtual money) to the network because other nodes incur a cost to forward the messages. The bundle sender first generates the base layer of a layered coin and then sends it together with the original bundles to a certain number of downlink nodes. At each subsequent hop, each intermediate node generates a new endorsed layer based on the previous layered coin. It is obvious that, with layered coins, each hop of a successful data-forwarding process can easily be tracked. After that, each intermediate node periodically submits its collected layered coins to the VB, which can calculate credits for each intermediate node and make a charge on the bundle senders. Note that, since only the nodes on the successful delivery path are rewarded, each intermediate node can launch different kinds of attacks on this naive system. In the following sections, we progressively determine what SMART needs to prevent the various attacks.

1) Preventing Layer Injection or Nodular Tontine Attack: Layer injection and nodular tontine attacks are two ways to cheat the SMART scheme. In a layer injection attack, several nodes may collude with each other to cheat extra credits. We assume that the total number of nodes along the successful delivery paths is m, and the source node is going to reward these m nodes with α credits. Each node is to receive α/m credits. However, if a malicious node colludes with other n nodes to launch a layer injection attack, the colluding group will receive $\alpha * ((n+1)/(n+m) - 1/m)$ extra credits. On the other hand, in a nodular tontine attack, an intermediate node tries to obtain extra credits by removing the endorsed layers generated by previous intermediate nodes. When a misbehaving node removes n layers from the original layered coin, it can make an extra profit of $\alpha * (1/(m-n) - 1/m)$.

The main reason behind layer injection and nodular tontine attacks is that the naive multilayer incentive scheme lacks any integrity protection mechanism to prevent the misbehaving nodes from arbitrarily injecting or removing layers. To thwart these attacks and ensure the security of layered coins, we introduce a *layer concatenation* technique, which tries to concatenate different layers with each other by injecting the generator information of the next layer into the previous layer [28], [29]. The basic idea of layer concatenation can be seen in Fig. 2. Starting from the source node, each node stores identification information about the next forwarding node set SET, which includes all the next-hop forwarders, in its layer. For example, in Fig. 2, the identity of the first intermediate node $\mathcal{N}_1$ is embedded in the base layer. This design disallows any subsequent forwarding nodes from removing endorsed layer I and its generator $\mathcal{N}_1$ from the layered coin since any attacker

[1] A tontine, named after Lorenzo Tonti, who was a Neapolitan banker who started it in 1653, is referred to as a financial system in which each investor equally contributes to a fund and, in return, receives dividends from the capital invested; as each person dies, his share is divided among all the others until only one is remaining, reaping all the benefits.

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

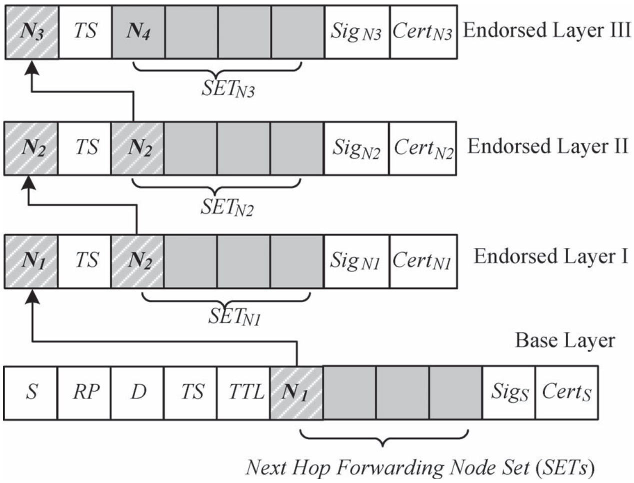


Fig. 2. Example of a layered coin for a single forwarding path.

has to forge a new non-$\mathcal{N}_1$-included base layer to replace the current one, although this cannot be achieved without the private key of the bundle sender. Similarly, the second intermediate node $\mathcal{N}_2$ is also defined in the endorsed layer generated by $\mathcal{N}_1$. Such a process will continue until the last endorsed layer generated by the destination. It is obvious that, with this layer-concatenation technique, the different layers can form a linkable layer chain. Each following node can easily detect the layer injection or nodular tontine attacks by checking the linkability of this layer chain.

In Fig. 2, we can further describe the components of a layered coin. A layered coin is composed of a *base layer* and multiple *endorsed layers*. A base layer is composed of the following: S and $Cert_S$, which are the identity and the public key certificate of the source node, respectively; RP, which refers to the CoS requirements and the rewarding policy proposed by the source D, which is the identity of the destination node; TS and TTL, which refer to the bundle creation timestamp and the time-to-live information, respectively; the forwarding node set SET, which includes all the possible forwarding nodes in the next hop; and Sig, which is the signature generated by the source node to protect the authenticity and integrity of the aforementioned information. Similar to the base layer, an endorsed layer includes the node identity, TS, SET, and a supporting signature.

2) Motivating Nodes to Submit Coins: We consider a countermeasure to the third type of selfish actions. As we have discussed earlier in this paper, due to lack of end-to-end connections in DTNs, SMART requires that the intermediate nodes opportunistically submit layered coins for clearance. However, the last intermediate node, i.e., the node that determines if a full linkable layer chain can be established, may collude with the sender to attack this system. In particular, if the last intermediate node does not submit the layered coin to the VB and loses the α/m credit, the sender can save α credits. In particular, if the sender gives the last intermediate node a behind-the-scene compensation of $\alpha/m+\epsilon$, where $\epsilon>0$, the last node will be better off while the sender still enjoys a net gain of $\alpha*(1-1/m)-\epsilon$. However, the other nodes, except the sender and the last intermediate node, will receive nothing, which may lead to a serious fairness issue.

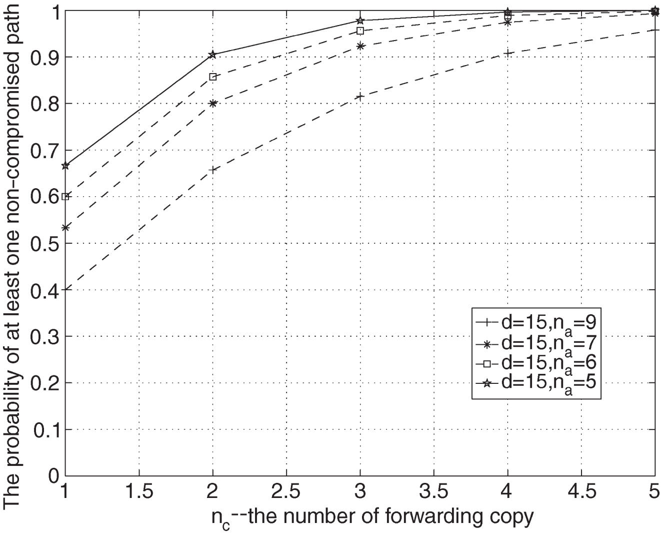


Fig. 3. Probability that there exists at least one noncompromised path under different n_c.

To prevent this cheating action, we propose two strategies to discourage the bundle sender from colluding with the last forwarding node. The first strategy is a charge-model-based solution [24]. For every forwarding request, SMART requires that the VB charges the sender an extra amount of credit α, even if the last intermediate node does not submit the layered coins for clearance. This extra charge is reasonable since, although it seems that no successful delivery path exists, this data forwarding still incurs forwarding costs to all the forwarding nodes involved. This extra charge goes to the VB, which either keeps it or returns the credit back to the involved forwarding nodes uniformly. Given such extra charges, even a colluding group cannot benefit from this cheating action.

SMART can also reduce the risk of the submission refusal attack by using multicopy forwarding. We assume that the source node colludes with n_a forwarding nodes to launch a submission refusal attack. Let n_c denote the number of copies transmitted for each message, and let d refer to the average number of one-hop neighbors of a DTN node. To maximize the attacking effect, we consider that all of the colluding nodes are located in the destination's transmission range. Given this setting, the probability of successfully launching a submission refusal attack can be defined as the probability that every successful delivery path is controlled by the colluding nodes. In other words, the probability of successfully defending a submission refusal attack (or the SR rate) is the probability that at least one forwarding path involves no colluding nodes. SR can be computed with the following equation:

$$FI=\begin{cases}1-\prod_{i=1}^{n_c}\frac{n_a-i+1}{d-i+1}, & \text{if } n_c\le n_a\\ 1, & \text{if } n_c>n_a.\end{cases}\qquad(1)$$

Fig. 3 shows the SR under different n_c, d, and n_a values. It is observed that SR very quickly increases when n_c increases. For example, when $d=15$, $n_a=9$, and $n_c=5$, SR is approximately 95.8%. Therefore, depending on the level of required security and the potential number of colluding nodes in the

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

network, the OSM can find an optimal n_c that achieves a good balance between security and efficiency.

C. SMART Scheme

In this section, we present the details of the SMART scheme, which includes "System Initialization," "Bundle Generation," "Bundle Forwarding," and "Charging and Rewarding" steps.

1) System Initialization: The OSM adopts bilinear pairing system parameters $(q, \mathbb{G}, \mathbb{G}_T, \hat{e}, P)$ as the system parameters. In addition, two hash functions are formed, i.e., $H : \{0,1\}^* \rightarrow \{0,1\}^*$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}$. The system parameters $(q, \mathbb{G}, \mathbb{G}_T, \hat{e}, P, H, H_2)$ will be preloaded in every DTN node. Each node $\mathcal{N}$ randomly chooses $sk_{\mathcal{N}} \in \mathbb{Z}_q^*$ as its private key, which corresponds to the public key expressed as $PK_{\mathcal{N}} = sk_{\mathcal{N}}P$. Then, it contacts the OSM to obtain its corresponding public key certificate.

2) Bundle Generation: When a bundle sender $\mathcal{S}$ is going to send a bundle B to the destination $\mathcal{D}$, after determining the next-hop forwarding node set $SET_{\mathcal{S}}$, $\mathcal{S}$ signs on the bundles with its private keys $sk_{\mathcal{S}}$ by computing $Sig_{\mathcal{S}} \leftarrow sk_{\mathcal{S}}H_2(B\|\mathcal{S}\|RP\|D\|TS\|TTL\|SET_{\mathcal{S}})$. Here, we use the Boneh–Lynn–Shacham signature [30] as the underlying building block to generate the supporting signature. Thus, $\mathcal{S}$ obtains the base layer $B\_layer = (S, RP, D, TS, TTL, SET_s, Sig_{\mathcal{S}}, CertS)$. Then, $\mathcal{S}$ forwards the bundle and the base layer to the next forwarding nodes as follows:

$$\mathcal{S} \rightarrow SET_{\mathcal{S}} : B, B\_Layer.$$

Notice that, in a multicopy opportunistic data-forwarding algorithm, a bundle may be forwarded along with multiple paths. Each forwarding path may form its layered coin, although the generated coins share the same base layer. Without loss of generality, in the following section, we take a single forwarding path $\mathcal{S} \rightarrow \mathcal{N}_1 \rightarrow \mathcal{N}_2 \rightarrow \cdots \mathcal{N}_i \cdots \rightarrow \mathcal{N}_m \rightarrow \mathcal{D}$ as an example to show the details of the basic SMART scheme, where $\mathcal{N}_m$ represents the last intermediate node.

3) Bundle Forwarding: When an intermediate node $\mathcal{N}_i$ receives the bundle and the layered coin, which includes a base layer and multiple endorsed layers, it performs several steps to authenticate the layered coin.

1) Check if the bundle is in their lifetime.
2) Check the linkability of the layer chains.
3) Verify the sender's certificate and check the supporting signature of the base layer by verifying if $\hat{e}(P, sig_s) = \hat{e}(PK_{\mathcal{S}}, H_2(B\|\mathcal{S}\|RP\|D\|TS\|TTL\|SET_{\mathcal{S}}))$ holds.
4) Verify the intermediate nodes' certificates and check the endorsed layers one by one.

After performing the aforementioned verifications and determining the next-hop forwarding node set $SET_{\mathcal{N}_i}$, $\mathcal{N}_i$ creates an additional endorsed layer by computing $Sig_{\mathcal{N}_i} \leftarrow sk_{\mathcal{N}_i}H_2(B\|B\_Layer\|\mathcal{N}_i\|TS\|SET_{\mathcal{N}_i})$ and thus obtains the ith endorsed layer $E\_Layer_i = (\mathcal{N}_i, TS, SET_{\mathcal{N}_i}, Sig_{\mathcal{N}_i}, Cert\mathcal{N}_i)$. Then, $\mathcal{N}_i$ forwards the bundle and the layered coin to the next forwarding node set as follows:

$$\mathcal{N}_i \rightarrow SET_{\mathcal{N}_i} : B, B\_Layer, E\_Layer_1, \ldots, E\_Layer_i.$$

The verification of the supporting signature of the ith endorsed layer is performed by computing if $\hat{e}(P, Sig_{\mathcal{N}_i}) = \hat{e}(\mathcal{N}_i, H_2(B\|B\_Layer\|\mathcal{N}_i\|TS\|SET_{\mathcal{N}_i})$ holds.

Similar steps are also be taken by each intermediate node before the bundles reach the destination $\mathcal{D}$. When the destination receives the bundles, it may also check the bundles' lifetime, senders and forwarders' certificates, and the layered coins one by one. If the verification passes, it may generate a special endorsed layer as the receipt, i.e., $Sig_{\mathcal{D}} \leftarrow sk_{\mathcal{D}}H_2(B\|B\_Layer\|\mathcal{D}\|TS)$. Thus, it obtains the endorsed layer $E\_Layer_{\mathcal{D}} = (\mathcal{D}, TS, Sig_{\mathcal{D}})$. Then, $\mathcal{D}$ sends it to $\mathcal{N}_m$ as follows:

$$\mathcal{D} \rightarrow \mathcal{N}_m : B, E\_Layer_{\mathcal{D}}.$$

Thus, the last intermediate node obtains a complete layered coin $B, B\_Layer, E\_Layer_1, \ldots, E\_Layer_i, \ldots, E\_Layer_m, E\_Layer_D$, which will be submitted to the VB for clearance in the future.

4) Charging and Rewarding: After a batch of a given size of layered coins is gathered, the last intermediate node may connect to the VB and submit the collected layered coins for clearance. After receiving the submitted layered coins, the VB first checks the certificates of each node in the forwarding path and then verifies the legitimacy of the layered coins. The VB also checks if these layered coins have been deposited before by inquiring the sender's previous record. If all verifications pass, a predefined amount of credit will be shared by all of the forwarders under a particular predefined rewarding policy.

Credit calculation should take bundle fragmentation into consideration. In DTNs, when a message is large, it may not be possible to send the entire message at once. One possible solution is to split the message into smaller pieces, wherein each piece becomes its own bundle or "fragment bundle," and send some pieces of a large message through the current link and the rest of the message through another link later to make the best use of limited resources. Bundle fragmentation is regarded as a unique characteristic of DTN forwarding [31], and a recent study has shown that the fragment size may follow a certain distribution in practice [32]. As the general discussion on credit calculation, we assume that there are n intermediate nodes participating in a successful bundle forwarding process, and each node $\mathcal{N}_i | 1 \leq i \leq n$ forwards δ_i percentage of fragments, where $0 < \delta_i \leq 1$. Then, the node $\mathcal{N}_i$ will receive $Cred_{\mathcal{N}_i} = \alpha * \delta_i / \Sigma_{j=1}^{n} \delta_j$ credits, where α is the total number of credits provided by the bundle sender.

D. Efficiency Enhancement

In this section, we propose two methods to further improve the computation and transmission efficiency of the SMART scheme.

1) Reducing the Transmission and Computation Overhead With an Aggregate Signature: Signature transmission and verification contribute to most of the transmission and computation overhead incurred by SMART transmission and verification. Therefore, reducing the signature size and increasing the verification efficiency is a major concern in the practical deployment

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

of the SMART scheme. Here, we take the advantage of an aggregate signature to reduce the transmission and verification cost.

An aggregate signature is a digital signature that supports aggregation of n distinct signatures issued by n distinct signers to a single short signature [30]. This single signature (and the n original messages) will convince the verifier that the n signers indeed sign the n original messages. With an aggregate signature, it is possible for the intermediate nodes to aggregate the received layered coins into a short signature.

Step 1) **Layered coin aggregation.** Let an intermediate node $\mathcal{N}_m$ receive a layered coin that is constituted with a base layer $B\_layer = (S, RP, D, TS, TTL, SET_s, Sig_{\mathcal{S}}, CertS)$ and multiple endorsed layer $E\_Layer_i = (\mathcal{N}_i, TS, SET_{\mathcal{N}_i}, Sig_{\mathcal{N}_i}, Cert\mathcal{N}_i) | 1 \leq i \leq m-1$, where $\mathcal{S} \rightarrow \mathcal{N}_1 \cdots \rightarrow \mathcal{N}_i \cdots \rightarrow \mathcal{N}_m$ is the current forwarding path. For the simplicity of presentation, we assume that $M_0 = B \| \mathcal{S} \| RP \| D \| TS \| TTL \| SET_{\mathcal{S}}$ and $M_i = B \| B\_Layer \| \mathcal{N}_i \| TS \| SET_{\mathcal{N}_i}$, where $1 \leq i \leq m-1$. Thus, the layered coin signatures can be represented as $Sig_{\mathcal{S}} \leftarrow sk_{\mathcal{S}} H_2(M_0)$ and $\{Sig_{\mathcal{N}_i} \leftarrow sk_{\mathcal{N}_i} H_2(M_i) | 1 \leq i \leq m-1\}$. To aggregate the layered coin, node $\mathcal{N}_m$ can compute and obtain the aggregate signature $Sig_{\text{agg}} \leftarrow Sig_{\mathcal{S}} \prod_{i=1}^{m-1} Sig_{\mathcal{N}_i}$. In the subsequent bundle-forwarding process, node $\mathcal{N}_m$ could transmit the aggregate signature Sig_{agg} rather than transmit the signatures one by one. Therefore, the transmission overhead can be reduced.

Step 2) **Layered coin batch verification.** Given the aggregate signature Sig_{agg}, the message M_0 and $\{M_i | 1 \leq i \leq m-1\}$ on which it is based, and public keys $PK_{\mathcal{S}}$ and $\{PK_{\mathcal{N}_i} | 1 \leq i \leq m-1\}$, node $\mathcal{N}_m$ can verify the aggregate signature by checking if $\hat{e}(Sig_{\text{agg}}, P) = \hat{e}(PK_{\mathcal{S}}, H_2(M_0)) \prod_{i=1}^{m-1} \hat{e}(PK_{\mathcal{N}_i}, H_2(M_i))$.

It is observed that the computation cost that the intermediate node spends on verifying m signatures is reduced from $2m$ pairing operations to $m+1$ pairing operations, where the pairing operation is the most computational expensive operation in the SMART scheme. Thus, this batch verification can dramatically reduce the verification delay, particularly when verifying a large number of layered coins.

2) Efficient Fragmentation Authentication With the Merkle Hash Tree: To support layered-coin-based fragment authentication in SMART, one possible way is to make each fragment self-authenticating by separately attaching a layered coin to the end of each fragment. However, this approach may lead to a more serious performance issue since the intermediate nodes have to spend more computational efforts on verifying an increasing number of signatures.

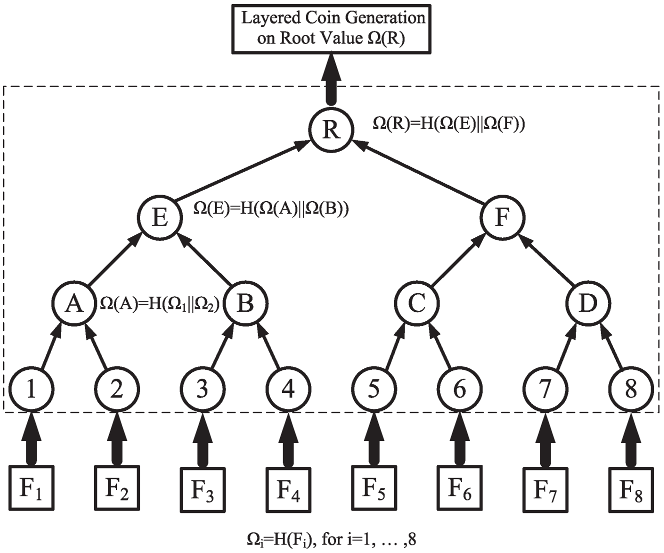


Fig. 4. Example of Merkle tree building.

The Merkle tree [33] (also called the binary hash tree) is a complete binary tree equipped with a function hash and an assignment Ω, which maps a set of nodes to a set of fixed-size strings. In a Merkle tree, the leaves of the tree contain the data, and the value of an internal tree node is the hash value of the concatenation of the values of its two children. Merkle tress have been applied in DTNs to realize efficient bundle authentication [35]. Here, we extend it to support efficient implementation of a credit-based incentive scheme or a Merkle-hash-tree-based SMART scheme (MKH-SMART).

Building a Merkle Tree: To build a Merkle tree for our problem, the sender constructs N leaves $\{\Omega_i = H(F_i) | i = 1, \ldots, m\}$, with each leaf corresponding to a fragment bundle, where $\{F_i | i = 1, \ldots, m\}$ refer to m fragments. The bundle sender then builds a complete Merkle tree with these leaves. The Ω value of each node is defined as

$$\Omega(V) = H\left(\Omega(V_{\text{left}}) \| \Omega(V_{\text{right}})\right))$$

where V denotes an internal tree node, and V_{left} and V_{right} denote V's two children. Fig. 4 shows an example of constructing such a Merkle tree. To add a credit-based incentive scheme to these bundles, the bundle sender only needs to generate a layered coin based on the root of the Merkle tree, which replaces the original bundle as the signed message.

Fragment Authentication With the Merkle-Tree-Based Incentive Scheme: To authenticate a particular fragment such as F_1, the intermediate node needs the set of hash values Ω_2, $\Omega(B)$, and $\Omega(D)$ and the base layer, which is a signature on the root $\Omega(E)$. The verifier can calculate each hash in the path from F_1 leaf node to the root node and finally check the validity of the layered coin. Note that, to verify m fragments, it only needs to perform one signature verification operation instead of verifying m signatures in total.

V. Performance Evaluation

In this section, we evaluate the performance of SMART from several aspects. Our evaluation begins with the cryptographic cost evaluation, which summarizes the computation and transmission cost incurred by the cryptographic operations in the SMART scheme. Then, by considering the cryptographic cost as the system parameter, we further demonstrate the effectiveness and efficiency of SMART in stimulating selfish nodes with extensive simulations. The evaluated schemes include the basic

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

TABLE I
SIZE OF EACH COMPONENT OF A LAYERED COIN (IN BYTES)

Base	Comp	$\mathcal{S}$	RP	D	TS		Total Size
	Size	4	10	4	4		
Layer	Comp	TTL	SET	Sig	Cert		66+4L
	Size	4	4L	20	20		
Endorsed	Comp	$\mathcal{N}_i$	TS	SET	Sig	Cert	Total Size
Layer	Size	4	4	4L	20	20	48+4L

TABLE II
CRYPTOGRAPHIC OPERATIONS' EXECUTION TIME

	Descriptions	Execution Time
T_{pmul}:	The time for one point multiplication in G	0.86 ms
T_{pair}:	The time for a pairing operation	4.14 ms

SMART, Agg-SMART, and MKH-SMART. Note that Agg-SMART and MKH-SMART can jointly be considered in the simulation as Optimized SMART.

A. Cryptographic Overhead Evaluation

1) Communication Overhead: One of the major advantages of SMART is the reduction in the transmission cost. It is observed that the communication cost of a layered coin is dominated by the size of supporting signatures generated by the intermediate nodes. To ensure the security of the protocol, the elements in $\mathbb{G}$ could be up to 160 bits. We summarize the approximated length of components of a layered coin in SMART, as shown in Table I. Note that L refers to the number of copies adopted in the bundle forwarding scheme. In the following performance analysis section, we take $L = 4$ as an example.

For m layered coins corresponding to m bundle fragments, each of which is accompanied with n endorsed coins, the total size of the layered coins (including both the base and endorsed layers) without aggregation should be $82m + 62mn$. However, in our Agg-SMART scheme, the total size can be reduced to $82m + (42n + 20)m$ by taking advantage of the aggregation signature. Under the same parameter, if every k fragment can be rebuilt with a Merkle hash tree, the total size of MKH-SMART can further be reduced to $82m/k + (42n + 20)m/k$. In other words, after adopting two optimization methods, the transmission overhead of the basic SMART scheme will be reduced from $82m + 62mn$ to $82m/k + (42n + 20)m/k$.

2) Computation Cost: The computation costs are measured by the most expensive pairing (Pair) and point multiplication (Pmul) operation. In the basic SMART scheme, a Pmul operation is involved for each base layer or endorsed layer generation, whereas two pairing operations are necessary for verification. To investigate the performance of the proposed SMART scheme, we first study the time for the Pmul operation and the Pair operation. We evaluate the delay of cryptographic operations on an Intel Pentium 4 3.0-GHz machine with 1-GB RAM, running on Fedora Core 4, based on the cryptographic library MIRACL [34], as shown in Table II.

TABLE III
SIMULATION PARAMETERS

Parameter	Value Range
Duration	12 hrs
Number of nodes	250 nodes
Speed of nodes	10 km/h ∼ 50 km/h
Transmission coverage	300 m
Mobility Model	Map based mobility model
Message size	5 m
Fragmentation size	10 k ∼ 100 k
Message generation interval	5 s ∼ 45 s
Routing Protocol	Spray and Wait routing protocol
Number of forwarding copies	1 ∼ 32 copies

Here, we focus on the cost of verifying the operation in SMART since the verification operation will be operated at each hop. Based on the execution time results, we have the verification cost for the nth intermediate node in the basic SMART as $T_{\text{SCI}} = 2 * mn * T_{\text{pair}}$, where m and n refer to the number of fragments. In the Agg-SMART scheme, by using an aggregate signature and a batch verification technique, the verification cost can be reduced to $T_{\text{agg-SCI}} = m * (n + 1)(T_{\text{pair}} + T_{\text{pmul}})$. The verification cost can further be reduced in the MKH-SMART scheme. Given that every k fragment can be rebuilt with a Merkle hash tree, the total verification cost of MKH-SMART can further be reduced to $T_{\text{MKH-SCI}} = m/k * (n + 1)(T_{\text{pair}} + T_{\text{pmul}})$.

After determining the cryptographic overhead, in the following sections, we will evaluate the performance of SMART by implementing SMART and optimized SMART on a specific DTN routing protocol.

B. Simulation

In this section, we evaluate the performance of SMART by simulations.

1) Simulation Setup: We implement our SMART scheme on a public available DTN simulator, namely, the Opportunistic Networking Environment simulator [36], and evaluate its performance under a practical application scenario, i.e., vehicular DTNs. We run our simulation with 250 vehicles uniformly deployed in an area of 4000 × 4000 m. The average speed of vehicles varies from 10 to 50 km/h (or from 2.7 to 13.9 m/s), and the transmission coverage of cars is 300 m. The map adopted in this paper is extracted from a real city map, which makes the model realistic. Each vehicle is first randomly scattered on one position of the roads and moved toward another randomly selected position along the paths in the map. The details of our simulation parameters are summarized in Table III.

Based on these parameters, we implement our SMART scheme on top of a typical multicopy DTN routing protocol, namely, the Spray and Wait routing (SW) protocol, the effectiveness and efficiency of which has been demonstrated in [9]. Generally speaking, spray and wait is available in the normal (nonbinary) and the binary variants. In this simulation, we choose binary spray and wait (SWB) as a basic routing protocol. However, it is important to point out that the SMART scheme can also be applied to other routing schemes if we choose a corresponding forwarding copy number for each forwarding hop.

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

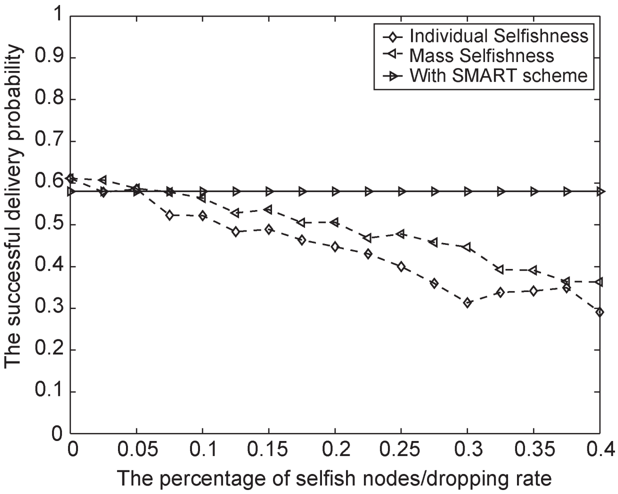


Fig. 5. Incentive effectiveness of SMART.

2) Incentive Effectiveness: We start our evaluation by observing the incentive effectiveness of the SMART scheme. We define two kinds of selfish scenarios: 1) individual selfishness and 2) mass selfishness. In an individual selfishness case, only a small number of selfish nodes may not be willing to forward packets for others, although it still expects others to forward packets on its behalf. On the other hand, mass selfishness can be defined that every node has the "intrinsic" selfishness nature so that it may probabilistically drop a certain percentage of messages instead of forwarding them. The incentive effectiveness can be measured by the message delivery probability, as shown in Fig. 5. For the mass selfishness case, when the packet dropping probability of each network node increases from 10% to 40%, the average successful delivery rate will drop from 56.39% to 36.31%. On the other hand, as for the individual selfishness, if 10% to 50% of network nodes are selfish nodes, the average successful delivery rate will dramatically decrease from 52.21% to 29.08%. This result demonstrates that the average network throughput could significantly degrade when the selfish nodes or selfish behaviors exist. However, with SMART in place, nodes are naturally motivated to participate in bundle forwarding to earn as many credits as possible. Although the successful delivery rate of SMART is slightly lower in the beginning due to the extra security overheads, the network throughput would remain relatively stable since SMART can successfully stimulate selfish nodes in packet forwarding. This demonstrates the incentive effectiveness of SMART. In the following section, we will discuss the other important metrics related to SMART-based DTN routing, namely, delivery ratio, overhead ratio, average latency, and number of forwarding copies.

3) Scenario I—Impact of Traffic Load: To evaluate the practicality of the SMART scheme, we first examine the system performance under different sending frequencies by adjusting the message generation interval, which is initialized to 35 s and then gradually decreased to 5 s. Fig. 6 shows the system performance of the original SWB routing protocol with no incentive scheme, SWB with the SMART scheme, and SWB with the optimized SMART scheme. The network performance can be measured in terms of three metrics: 1) successful delivery rate; 2) overhead ratios; and 3) average latency. Fig. 6(a) shows the relationship between the successful delivery rate and the message sending frequency. It is clear that a higher message sending frequency would result in a lower delivery rate in different SWB scenarios due to the increased number

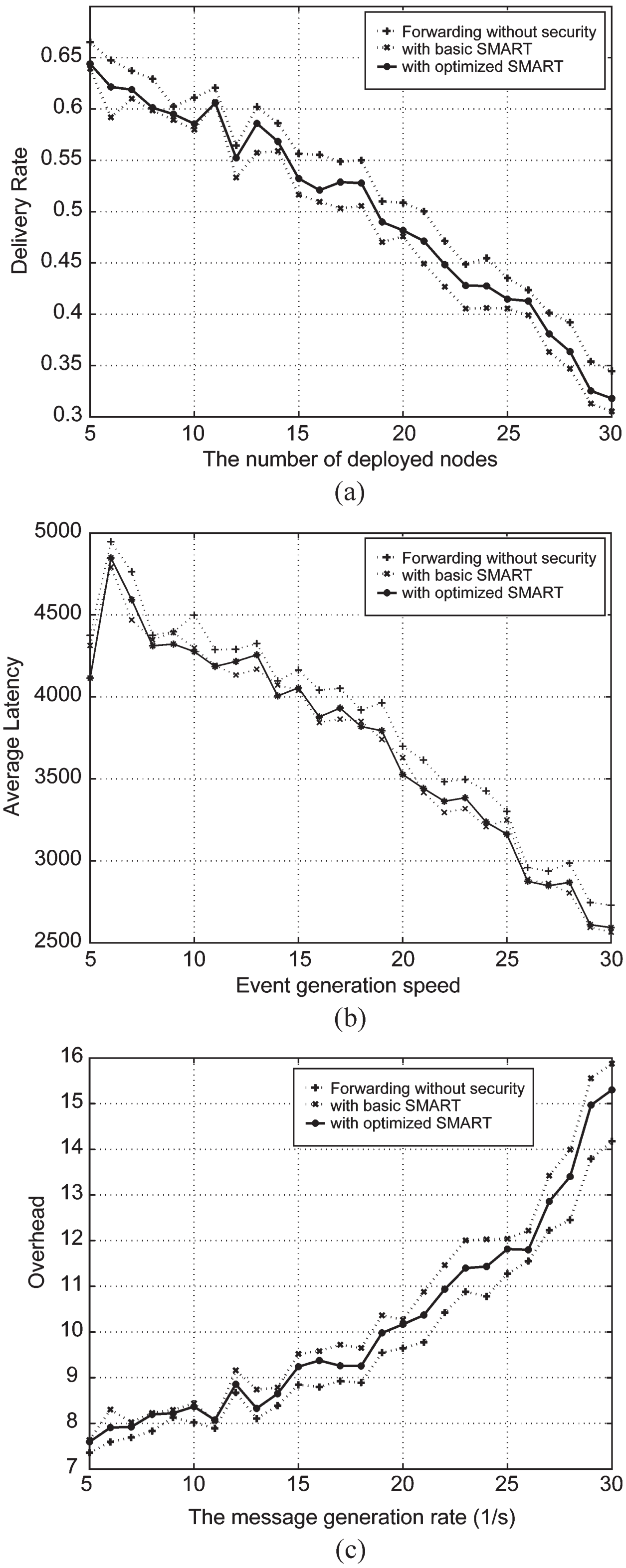


Fig. 6. Impact of network load on system performance. (a) Successful delivery ratio. (b) Average latency. (c) Overhead ratio.

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

of forwarding messages. However, we can also see that the performance of SWB with the SMART scheme and optimized SMART is very close to that of SWB without any security addons. For example, when a high message forwarding frequency is in place (e.g., the message generation interval is set to 5 s), the SMART scheme incurred a 13.3% decrease in the successful delivery rate, whereas the optimized SMART scheme only incurred an 8.3% decrease. Fig. 6(b) shows the average latency of different scenarios. It is observed that, after a small increase, the average latency will quickly decrease, and optimized SMART has a comparable performance with the SMART scheme, both of which are less than the no-security system. This is mainly caused by the dramatically decreased delivery rate. Fig. 6(c) demonstrates that SMART and optimized SMART only have a slightly larger overhead than the no-incentive SWB scheme. However, the increased overhead is not very significant, and thus, they have a similar overall performance.

4) Scenario II—Impact of Forwarding Copy Number: Multicopy data forwarding is a major characteristic of DTN data forwarding. In this section, we investigate the impact of number of copies on the system performance, and we also study how to find an optimal forwarding copy number with or without an incentive mechanism. In Fig. 7, the number of forwarding copies is initially set to 1 and then increased one by one. It is obvious that the delivery rate will increase very fast in the beginning and then decrease after a specific threshold [for example, 5 in Fig. 7(a)]. This shows that an optimal copy number exists to achieve the highest successful delivery rate. On the other hand, in Fig. 7(b) and (c), it is observed that average latency will decrease with the increased copies, whereas overhead will significantly increase along the forwarding copies. By jointly considering the system performance and ensuring a certain security level, the OSM can choose an optimal forwarding copy number to find a balance between security and system performance.

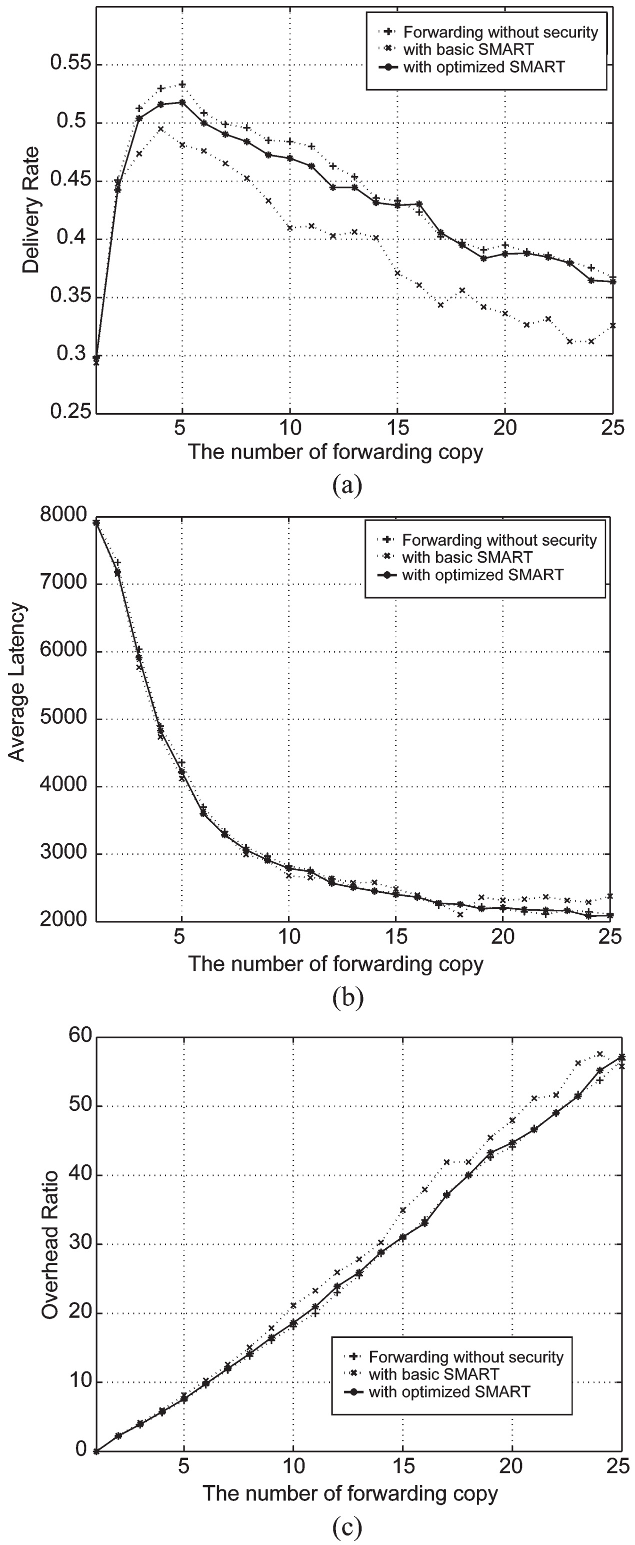
In summary, the simulation results demonstrate that SMART is indeed a viable lightweight solution that stimulates bundle forwarding in a DTN environment.

VI. Further Discussion

In previous sections, we have introduced the SMART scheme in detail, which can be used to stimulate routing and data forwarding in DTNs. In this section, we further discuss other challenges related to secure incentive design in DTNs.

A. Public Key Revocation in DTNs

Public key management is the foundation of any security protocol. For a secure incentive scheme, any misbehaving or malicious nodes will pay the penalty of having their public key certificates revoked. Even for those selfish nodes that run out of their credits, one possible punishment action is also revoking their certificates or reducing their CoS right by revising their certificates. However, public key revocation still represents a great challenge in DTNs. In a traditional Public Key Infrastructure, the most commonly adopted certificate revocation scheme is through a Certificate Revocation List (CRL), which is a list of revoked certificates stored in central repositories prepared by the Certificate Authorities. However, in DTNs, the nodes may suffer from delayed or frequent loss of connectivity to CRL servers [22]. In [23], the use of periodical public key updating is suggested to replace the traditional public key revocation, although in the real world, public key distribution is also a



(a)

(b)

(c)

Fig. 7. Impact of forwarding copy number on system performance. (a) Delivery ratio. (b) Average latency. (c) Overhead ratio.

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

challenging problem and may lead to a lot of extra management costs. Another possible way to address public key revocation in DTNs is by using cooperative CRL distribution [22], which needs further investigation to find an improved method.

B. Public Key Cryptography Versus IBC

Currently, we use traditional public-key-certificate-based cryptography as the basic cryptographic tool to realize our SMART scheme. One possible way to further improve the efficiency of SMART is using identity-based cryptography (IBC) to redesign the current public-key-certificate-based protocol. IBC is a relatively new cryptographic method and is also a powerful alternative to traditional certificate-based cryptography [27]. Its main idea is to make an entity's public key directly derivable from its publicly known identity information such as the e-mail address. Recently, there have been several research proposals that have suggested adoption of IBC to realize the efficient bundle authentication in DTNs [35]. However, it is straightforward to transform our public-key-certificate-based SMART scheme into an ID-based SMART scheme by adopting an ID-based signature scheme such as [37]. Therefore, adopting IBC will not affect the contribution of this paper.

VII. Conclusion

In this paper, we have proposed a SMART scheme to stimulate cooperation in packet forwarding for DTNs. We have also proposed two efficiency-optimization methods to reduce the transmission and computation overhead. The SMART scheme is compatible to diverse existing routing schemes and is expected to improve the system performance of DTNs, which suffer from selfishness. We have also demonstrated the efficiency and effectiveness of SMART through extensive simulations. Our future work includes a reputation-based incentive scheme or a secure incentive-compatible routing scheme for DTNs.

References

[1] A. Kate, G. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Proc. SecureComm*, Sep. 2007, pp. 504–513.

[2] H. Zhu, R. Lu, X. Lin, and X. Shen, "Security in service-oriented vehicular networks," *IEEE Wireless Commun., Special Issue on Service-Oriented Broadband Wireless Network Architecture*, vol. 16, no. 4, Aug. 2009, to be published.

[3] R. Lu, X. Lin, H. Zhu, P. H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE Infocom*, Phoenix, AZ, Apr. 14–18, 2008, pp. 1229–1237.

[4] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. Infocom*, 2006, pp. 1–12.

[5] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proc. MobiHoc*, 2008, pp. 241–250.

[6] A. Garyfalos and K. C. Almeroth, "Coupons: A multilevel incentive scheme for information dissemination in mobile networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 6, pp. 792–804, Jun. 2008.

[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proc. IEEE Infocom*, 2006, pp. 1–13.

[8] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications," *IEEE Netw.—Special Issue Wireless Sensor Netw.*, vol. 20, no. 3, pp. 12–18, May 2006.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy cast," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy cast," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[11] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. ACM SIGCOMM*, 2004, pp. 145–158.

[12] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, Tech. Rep. CS-200006, Apr. 2000.

[13] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Proc. SAPIR*, 2004, pp. 239–254.

[14] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. ACM Mobicom*, Boston, MA, Aug. 2000, pp. 255–265.

[15] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputation-based incentive scheme for ad hoc networks," in *Proc. WCNC*, Atlanta, GA, Mar. 2004, pp. 825–830.

[16] Y. Zhang and Y. Fang, "A fine-grained reputation system for reliable service selection in peer-to-peer networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 8, pp. 1134–1145, Aug. 2007.

[17] S. Buchegger and J. Le Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes-fairness in distributed ad-hoc networks," in *Proc. IEEE/ACM MobiHoc*, Lausanne, Switzerland, Jun. 2002, pp. 226–236.

[18] Y. Zhang, W. Lou, W. Liu, and Y. Fang, "A secure incentive protocol for mobile ad hoc networks," *Wirel. Netw.*, vol. 13, no. 5, pp. 569–582, Oct. 2007.

[19] S. B. Lee, G. Pan, J.-S. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. MobiHoc*, Sep. 2007, pp. 150–159.

[20] R. Lu, X. Lin, H. Zhu, C. Zhang, P. H. Ho, and X. Shen, "A novel fair incentive protocol for mobile ad hoc networks," in *Proc. IEEE WCNC*. Las Vegas, NV, Mar. 31–Apr. 3 2008, pp. 3237–3242.

[21] M. Felegyhazi, J.-P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 463–476, May 2006.

[22] X. Lin, R. Lu, C. Zhang, H. Zhu, P. H. Ho, and X. Shen, "Security in vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 88–95, Apr. 2008.

[23] A. Seth, U. Hengartner, and S. Keshav, "Practical security for disconnected nodes," in *Proc. NPSec*, Nov. 2005, pp. 31–36.

[24] S. Zhong, J. Chen, and Y. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE Infocom*, 2003, pp. 1987–1997.

[25] S. Zhong, L. Li, Y. Liu, and Y. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks," *Wirel. Netw.*, vol. 13, no. 6, pp. 799–816, Dec. 2007.

[26] S. Symington, S. Farrell, H. Weiss, and P. Lovell, *Bundle Security Protocol Specification*, Feb. 2008. draft-irtf-dtnrg-bundle-security-05.txt, work-in-progress.

[27] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," in *Proc. Crypto*. New York: Springer-Verlag, 2001, vol. 2139, *LNCS*, pp. 213–229.

[28] H. Zhu, X. Lin, M. Shi, P.-H. Ho, and X. Shen, "PPAB: A privacy preserving authentication and billing architecture for metropolitan area sharing networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2529–2543, Jun. 2009.

[29] H. Zhu, X. Lin, R. Lu, P. H. Ho, and X. Sheng, "SLAB: Secure localized authentication and billing scheme for wireless mesh networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 3858–3868, Oct. 2008.

[30] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, Sep. 2004.

[31] S. Farrell and V. Cahill, "DTN: An architectural retrospective," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 828–836, Jun. 2008.

[32] M. Pitkanen, M. Keranen, and J. Ott, "Message fragmentation in opportunistic DTNs," in *Proc. WoWMoM*, 2008, pp. 1–7.

[33] R. Merkle, "Protocols for public key cryptosystems," in *Proc. IEEE SP*, 1980, pp. 122–133.

[34] *Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL)*.

[35] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo, "Applicability of identity-based cryptography for disruption-tolerant networking," in *Proc. 1st Int. MobiOpp*, Jun. 2007, pp. 52–56.

[36] *The One Simulator*. [Online]. Available: http://www.netlab.tkk.fi/tutkimus/dtn/theone/

[37] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie–Hellman groups," in *Proc. PKC*, 2003, vol. 2567, pp. 18–30.

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.

Haojin Zhu (M'09) received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2002, the M.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009.

He is currently an Assistant Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include wireless network security and applied cryptography.

Mr. Zhu was a recipient of the Best Paper Award at the 2007 IEEE International Communications Conference: Computer and Communications Security Symposium and at the 2008 Third International Conference on Communications and Networking in China: Wireless Communication Symposium.

Xiaodong Lin (S'07–M'09) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998 and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008.

He is currently an Assistant Professor with the Faculty of Business and Information Technology, Institute of Technology, University of Ontario, Oshawa, ON. His research interests include wireless network security, applied cryptography, and anomaly-based intrusion detection.

Dr. Lin was a recipient of the Natural Sciences and Engineering Research Council of Canada Canada Graduate Scholarships Doctoral and the Best Paper Award at the 2007 IEEE International Communications Conference: Computer and Communications Security Symposium and at the 2008 Third International Conference on Communications and Networking in China: Wireless Communication Symposium.

Rongxing Lu (S'09) is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is currently a Research Assistant with the Broadband Communications Research Group, University of Waterloo. His research interests include wireless network security, applied cryptography, and trusted computing.

Dr. Lu was a recipient of the Best Paper Award at the 2007 IEEE International Communications Conference: Computer and Communications Security Symposium and at the 2008 Third International Conference on Communications and Networking in China: Wireless Communication Symposium.

Yanfei Fan received the B.Sc. degree in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2002 and the M.Sc. degree in computer science from Tsinghua University, Beijing, in 2005. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

His current research interests include wireless network security and network coding.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees from Rutgers University, Camden, NJ, in 1987 and 1990, respectively, all in electrical engineering.

He is a Professor, the University Research Chair, and the Associate Chair for Graduate Studies with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He serves as the Editor-in-Chief for *Peer-to-Peer Networking and Application* and as an Associate Editor for *Computer Networks*, *ACM/Wireless Networks*, and *Wireless Communications and Mobile Computing*. His research focuses on mobility and resource management in interconnected wireless/wireline networks, ultra-wideband wireless communications systems, wireless security, and ad hoc and sensor networks. He is a coauthor of three books and has published more than 300 papers and book chapters in wireless communications and networks, control, and filtering.

Dr. Shen is a Registered Professional Engineer in the Province of Ontario, Canada. He served as the Technical Program Committee Chair of the 2007 IEEE Global Communications Conference; a General Cochair of the 2007 International Conference on Communications and Networking in China and the 2006 Third International Conference on Heterogeneous Networking for Quality, Reliability, Security, and Robustness; and the Founding Chair of the IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He has also served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and *IEEE Communications Magazine*. He was the recipient of the Distinguished Performance Award from the Faculty of Engineering, University of Waterloo, in 2002; the Premier's Research Excellence Award from the Province of Ontario in 2003; and the Excellent Graduate Supervision Award and the Outstanding Performance Award from the University of Waterloo in 2006 and 2004, respectively.

Authorized licensed use limited to: University of Waterloo. Downloaded on November 21, 2009 at 02:10 from IEEE Xplore. Restrictions apply.