International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue I, December 2014
ISSN: 2394 - 6598

HANDLING FLOOD ATTACKS IN DISRUPTION TOLERANT NETWORKS BASED ON CLAIM VERIFICATION

K.PRIYA,
PG SCHOLAR,KALASALINGAM INSTITUTE OF TECHNOLOGY,
KRISHNAN KOVIL,VIRUDHUNAGAR,TAMILNADU.
rajeramcse@gmail.com

Abstract
Disruption Tolerant Networks(DTNs) consist of mobile nodes carried by human beings.DTNs utilize the mobility of nodes and the opportunistic contacts among nodes for data communication.DTNs are vulnerable to flood attacks due to limintations in bandwidth and buffer space.Flooded packets and replicas can waste the bandwidth and buffer resources and degrade the network service provided to good nodes.By using rate limiting we employ defend against flood attack in DTN.If a node violates its rate limits it will be detected and its traffic will be filtered.Claim-carry-and-check adopts that each node counts the number of packets and sent the count to other nodes.DTN are mobility of nodes which consist of the limited buffer space, & also it is Vulnerable to flood attacks.The receiving nodes use the claims and cross-check to find the inconsistant when the nodes are contact.The detection probability can be flexibly adjusted by system parameters that controls the amount of claims exchanged in contact.

Index Terms—DTN, security, flood attack, detection

1. INTRODUCTION

DTNs enable data transfer when mobile nodes are only intermittently connected, making them appropriate for applications where no communication infrastructure is available such as military scenarios and rural areas. Due to lack of consistent connectivity, two nodes can only exchange data when they move into the transmission range of each other .DTNs employ such contact opportunity for data forwarding with "store-carry-and-forward.Since the contacts between nodes are opportunistic and the duration of a contact may be short because of mobility, the usable bandwidth which is only available during the opportunistic contacts is a limited resource. Also, mobile nodes may have limited buffer space.

Due to limitations in bandwidth and buffer space,DTNs are vulnerable to flood attacks.In flood attacks, maliciously or selfishly motivated attackers inject as many packets as possible into the network,or instead of injecting different packets the attackers forward replicas of the same packet to as many nodes as possible. For convenience, we call the two types of attack packet flood attack and replica flood attack, respectively. Flooded packets and replicas can waste the precious bandwidth and buffer resources, prevent benign packets from being forwarded and thus degrade the network service provided to good nodes. Moreover, mobile nodes spend much energy on transmitting/receiving flooded packets and replicas which may shorten their battery life. Therefore, it is urgent to secure DTNs against flood attacks.Although many schemes have been proposed to defend against flood attacks on the Internet and in wireless sensor networks , they assume persistent connectivity and cannot be directly applied to DTNs that have intermittent connectivity. We noted that the packets flooded by outsider attackers can be easily filtered with authentication techniques. However, authentication alone does not work when insider attackers flood packets and replicas with valid signatures. Thus, it is still an open problem is to address flood attacks in DTNs.In this paper, we employ rate limiting to defend against flood attacks in DTNs.

In our approach, each node has a limit over the number of packets that it, as a source node, can send to the network in each time interval. Each node also has a limit over the number of replicas that it can generate for each packet (i.e., the number of nodes that it can forward each packet to). The two limits are used to mitigate packet flood and replica flood attacks, respectively. If a node violates its rate limits, it will be detected and its data traffic will be filtered. In this way, the amount of flooded traffic can be controlled. Our main contribution is a technique to detect if a node has violated its rate limits. Although it is easy to detect the violation of rate limit on the Internet and in telecommuni- cation networks where the egress router and base station can account each user’s traffic, it is challenging in DTNs due to lack of communication infrastructure and consistent connectivity.The detection probability can be flexibly adjusted by system parameters that control the amount of claims exchanged in a contact. We provide a lower and upper bound of detection probability and investigate the problem of parameter selection to maximize detection probability under a certain amount of exchanged claims. The effectiveness and efficiency of our scheme are evaluated with extensive trace-driven simulations.

2.MOTIVATION

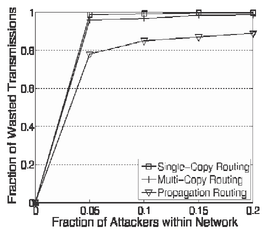
2.1 The Potential Prevalence of Flood Attacks

Many nodes may launch flood attacks for malicious or selfish purposes. Malicious nodes, which can be the nodes deliberately deployed by the adversary or subverted by the adversary via mobile phone worms , launch attacks to congest the network and waste the resources of other nodes.Selfish nodes may also exploit flood attacks to increase their communication throughput. In DTNs, a single packet usually can only be delivered to the destination with a probability smaller than 1 due to the opportunistic con- nectivity. If a selfish node floods many replicas of its own packet, it can increase the likelihood of its packet being delivered, since the delivery of any replica means successful delivery of packets.

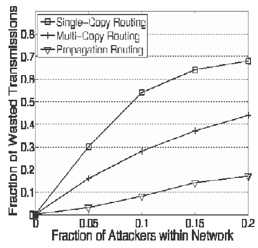
2.2 The Effect of Flood Attacks

To study the effect of flood attacks on DTN routing and motivate our work, we run simulations on the MIT Reality trace .We consider three general routing strategies in DTNs.

1) Single-copy routing : after forwarding a packet out, a node deletes its own copy of the packet. Thus, each packet only has one copy in the network. 2) Multicopy routing : the source node of a packet sprays a certain number of copies of the packet to other nodes and each copy is individually routed using the single-copy strategy. The maximum number of copies that each packet can have is fixed. 3) Propagation routing : when a node finds it appropriate to forward a packet to another encountered node, it replicates that packet to the encountered node and keeps its own copy. There is no preset limit over the number of copies a packet can have. In our simulations, SimBet , Spray-and- Focus and Propagation are used as representatives of the three routing strategies, respectively. In Propagation, a node replicates a packet to another encountered node if the latter has more frequent contacts with the destination of the packet. Fig. 1 shows the effect of flood attacks on wasted transmission



(a) Packet Flood Attack



(b) Replica Flood Attack

Fig. 1. The effect of flood attacks on the fraction of wasted transmission. Attackers are randomly deployed.

3 OVERVIEW

3.1 Problem Definition

3.1.1 Defense against Packet Flood Attacks

We consider a scenario where each node has a rate limit L on the number of unique packets that it as a source can generate and send into the network within each time interval T. The time intervals start from time 0, T, 2T, etc. The packets generated within the rate limit are deemed legitimate, but the packets generated beyond the limit are deemed flooded by this node. To defend against packet flood attacks, our goal is to detect if a node as a source has generated and sent more unique packets into the network than its rate limit L per time interval.

3.1.2 Defense against Replica Flood Attacks

The defense against replica flood considers single-copy and multicopy routing protocols. These protocols require that, for each packet that a node buffers no matter if this packet has been generated by the node or forwarded to it, there is a limit l on the number of times that the node can forward this packet to other nodes. The values of l may be different for different buffered packets. Our goal is to detect if a node has violated the routing protocol and forwarded a packet more times than its limit l for the packet.

3.1.3 Setting the Rate limit L

One Possible method is to set L in a request-approve style. When a user joins the network, she requests for a rate limit from a trusted authority which acts as the network operator. In the request, this user specifies an appropriate value of L based on prediction of her traffic demand. If the trusted authority approves this request, it issues a rate limit certificate to this user, which can be used by the user to prove to other nodes the legitimacy of her rate limit. To prevent users from requesting large rate limits, a user pays an appropriate amount of money or virtual currency for her rate limit.

3.2 Models and Assumptions

3.2.1 Network Model

In DTNs, since contact durations may be short, a large data item is usually split into smaller packets (or fragments) to facilitate data transfer. The packet becomes meaningless after its lifetime ends and will be discarded.We assume that every packet generated by nodes is unique. This can be implemented by including the source node ID and a locally unique sequence number, which is assigned by the source for this packet, in the packet header.

3.2.2 Adversary Model

There are a number of attackers in the network. An attacker can flood packets and/or replicas. When flooding packets, the attacker acts as a source node. It creates and injects more packets into the network than its rate limit L. Some attackers may collude and communicate via out-band channels.

3.2.3 Trust Model

KGC generates a private key for each node based on the node’s id, and publishes a small set of public security parameters to the node. With such a system, an attacker cannot forge a node id and private key pair. Each node has a rate limit certificate obtained from a trusted authority. The certificate includes the node’s ID, its approved rate limit L, the validation time of this certificate and the trusted authority’s signature.

3.3 Claim-Carry-and-Check

3.3.1 Packet Flood Detection

To detect the attackers that violate their rate limit L, we must count the number of unique packets that each node as a source has generated and sent to the network in the current interval. our idea is to let the node itself count the number of unique packets that it, as a source, has sent out, and claim the up-to-date packet count in each packet sent out. The node's rate limit certificate is also attached to the packet, such that other nodes receiving the packet can learn its authorized rate limit L. If an attacker is flooding more packets than its rate limit, it has to dishonestly claim a count smaller than the real value in the flooded packet, since the real value is larger than its rate limit and thus a clear indicator of attack.

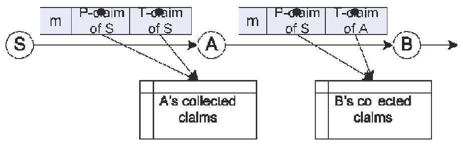


Fig. 2. The conceptual structure of a packet and the changes made at each hop of the forwarding path.

3.3.2 Replica Flood Detection

Claim-carry-and-check can also be used to detect the attacker that forwards a buffered packet more times than its limit l. Specifically, when the source node of a packet or an intermediate hop transmits the packet to its next hop, it claims a transmission count which means the number of times it has transmitted this packet (including the current transmission). Based on if the node is the source or an intermediate node and which routing protocol is used, the next hop can know the node's limit l for the packet, and ensure that the claimed count is within the correct range 1; l . Thus, if an attacker wants to transmit the packet more than l times, it must claim a false count which has been used before. Similarly as in packet flood attacks, the attacker can be detected.

4 OUR SCHEME

4.1 Claim Construction

Two pieces of metadata are added to each packet (see Fig. 2), Packet Count Claim (P-claim) and Transmission Count Claim (T-claim). P-claim and T-claim are used to detect packet flood and replica flood attacks, respectively.P-claim is added by the source and transmitted to later hops along with the packet. T-claim is generated and processed hop-by-hop. Specifically, the source generates a T-claim and appends it to the packet. When the first hop receives this packet, it peels off the T-claim; when it forwards the packet out, it appends a new T-claim to the packet. This process continues in later hops. Each hop keeps the P-claim of the source and the T-claim of its previous hop to detect attacks.

4.1.1 P-Claim

When a source node S sends a new packet m (which has been generated by S and not sent out before) to a contacted node, it generates a P-claim as follows:

P-claim: S; c_p; t; Hðmþ; SIG$_S$ ðHðH ðmþjSjc$_p$jtþþ: ð1þ

Here, t is the current time. c_p (c_p 2 ½1; L) is the packet count of S, which means that this is the c^{th} new packet S has created and sent to the network in the current time interval. S increases c_p by one after sending m out.The P-claim is attached to packet m as a header field, and will forwarded along with packet to later hops.

4.1.2 T-Claim

When node A transmits a packet m to node B, it appends a T-claim to m. The T-claim includes A's current transmission count c_t for m (i.e., the number of times it has transmitted m out) and the current time t. The T-claim is

T-claim: A; B; H ðmþ; c_t; t; SIG$_A$ ðHðAjBjH ðmþjc$_t$jtþþ: ð2þ

B checks if c_t is in the correct range based on if A is the source of m. If c_t has a valid value,B stores this T-claim.In single-copy and multicopy routing, after forwarding m for enough times, A deletes its own copy of m and will not forward m again.

4.2 Inconsistency Caused by Attack

In a dishonest P-claim, an attacker uses a smaller packet count than the real value. (We do not consider the case where the attacker uses a larger packet count than the real value, since it makes no sense for the attacker.) However, this packet count must have been used in another P-claim generated earlier. This causes an inconsistency called count reuse, which means the use of the same count in two different P-claims generated by the same node.

4.3 Protocol

Suppose two nodes contact and they have a number of packets to forward to each other. Then our protocol is sketched in Algorithm 1.

Algorithm 1. The protocol run by each node in a contact

```
1: Metadata (P-claim and T-claim)
   exchange and attack detection
2: if Have packets to send then
3: For each new packet, generate a P-claim;
4:  For all packets, generate their T-claims
      and sign them with a hash tree;
5:    Send every packet with the P-claim
      and T-claim attached;
6: end if
7: if Receive a packet then
8:    if Signature verification fails or the count
      value in its P-claim or T-claim is invalid
      then
9:         Discard this packet;
10:     end if
11:     Check the P-claim against those locally
        collected and generated in the same time
        interval to detect inconsistency;
12:     Check the T-claim against those locally
        collected for inconsistency;
13:     if Inconsistency is detected then
14:         Tag the signer of the P-claim (T-
            claim, respec- tively) as an attacker
            and add it into a blacklist;
15:         Disseminate an alarm against the
            attacker to the network;
16:     else
17:          Store the new P-claim (T-claim,
respectively);
18:     end if
19: end if
```

4.4 Local Data Structures

Each node collects P-claims and T-claims from the packets that it has received and stores them locally to detect flood attacks. Let us look at a received packet m and the P-claim and T-claim included in this packet. Initially, this pair of P-claim and T-claim are stored in full with all the components shown in Formulas 1 and 2.

4.4.1 Compact P-Claim Storage

Suppose node W stores a P-claim C_p ¼ fS; c_p ; t; H ðmÞ; SIG_S g. It compacts the P-claim as follows: using the timestamp t, W gets the index i of the time interval that t belongs to.The compacted P-claim is

$$S;\ i;\ c_p;\ \hat{H}_8\ ;$$

where $\hat{H}_8$ is a 8-bit string called hash remainder.

4.4.2 Compact T-Claim Storage

Suppose node W stores a T-claim ℂT ¼ fR; W ; H ðmÞ; ct ; t; SIGR g issued by node R. The signature is discarded since it has been verified. W does not need to store its own ID and t is not useful for inconsistency check. Then the compacted T- claim is

R; ct ; H̃32 ;

where H̃32 is a 32-bit hash remainder defined similarly as H̃8 . Suppose W has collected n T-claims generated by R. Then the compact structure of these T-claims is

CR ¼ R; locators; ½H̃321 ; ct1 ;. . . ;
[½H̃32n ; ctn :]

The locators are randomly and independently generated by W for R, and are shared by all the T-claims issued by R.

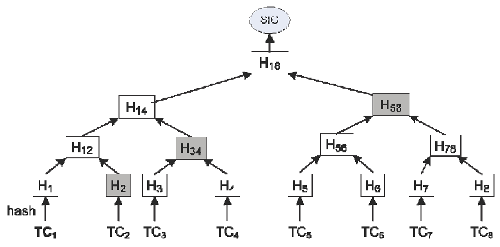


Fig. 3. The Merkle hash tree constructed upon eight T-claims TC1....TC8.

4.5 Efficient T-Claim Authentication

The T-claims of all the packets transmitted in a contact should be signed by the transmitting node. Since the contact may end at any unpredictable time, each received T-claim must be individually authenticated. A naive approach is to protect each T-claim with a separate public-key signature, but it has high computation cost in signature generation and verification.Our scheme uses Merkle hash tree to amortize the computation cost of public-key-based signature on all the T-claims that the node sends out in a contact.

4.6 Dealing with Different Rate Limits

Previously we have assumed that all nodes have the same rate limit L. When nodes have different rate limits, for our detection scheme to work properly, each intermediate node that receives a packet needs to know the rate limit L of the source of the packet, such that it can check if the packet count is in the correct range 1; 2; . . . ; L. To do so, when a source node sends out a packet, it attaches its rate limit certificate to the packet. The intermediate nodes receiving this packet can learn the node's authorized rate limit from the attached certificate.

5 ANALAYSIS

This section presents rigorous analysis over the security and cost of our scheme, and discusses the optimal parameter to maximize the effectiveness of flood attack detection under a certain amount of exchanged metadata per contact.

5.1 Detection Probability

The following analysis assumes uniform and independent contacts between nodes, i.e., at any time each node's next contacted node can be any other node with the same probability. This assumption holds for mobility models such as Random Waypoint (RWP) where the contacts between all node pairs can be modeled as i.i.d. Poisson processes. When analyzing the detection probability, we assume that each attacker acts alone.

5.1.1 The Basic Attack

First we consider a basic attack (see Fig. 4 a) in which an attacker S floods two sets of mutually inconsistent packets to two good nodes A and B, respectively. Each flooded packet received by A is inconsistent with one of the flooded packets received by B. In the contacts with A and B, S also forwards some normal, not flooded, packets to A and B to make the attack harder to detect. Let y denote the proportion of flooded packets among those sent by S. For simplicity, we assume y is the same in both contacts. Suppose A and B redirect the claims sampled in the contact with S to C and D, respectively.

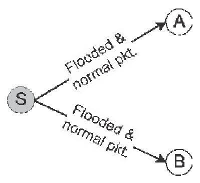


Fig 4.The Basic Attack

5.2 Cost Ananlysis

5.2.1 Communication

The communication cost mainly has two parts. One part is the P-claim and T-claim transmitted with each packet and the other part is the partial claims transmitted during metadata exchange.At most,42K P-claims and 4Z K T-claims are exchanged in each contact,with one half for sampled and the other half for redirected claims.

5.2.2 Computation

As to signature generation,a node generates one signature for each newly generated packet.It also generates one signature for all its T-claims as a whole sent in a contact.As to signature verification,a node verifies the signature of each received packet.It also verifies one signature for all the T-claims as a whole received in one contact.

5.2.3 Storage

Most P-claims and T-claims are compacted when the packets are forwarded.The Z sampled P-claims and T-claims are stored in full until the packets are forwarded or have been exchanged to K nodes.For each received packet,less than 20 bytes of compact claims are stored for time duration T.

5.3 Collusion Analysis

5.3.1 Packet Flood Attack

One attacker may send a packet with a dishonest packet
count to its colluder, which will forward the packet to the network. Certainly, the colluder will not exchange the dishonest P-claim with its contacted nodes. However, so long as the colluder forwards this packet to a good node,this good node has a chance to detect the dishonest claim as well as the attacker. Thus, the detection probability is not affected by this type of collusion.

5.3.2 Replica Flood Attack

When attackers collude, they can inject invalid replicas of a packet without being detected, but the number of flooded replicas is effectively limited in our scheme. More specifically,in our scheme for a unique packet all the M colluders as a whole can flood a total of M _ 1 invalid replicas without being detected. To the contrast, when there is no defense, a total of N _M invalid replicas can be injected by the colluders for each unique packet. Since the number of colluders is not very large, our scheme can still effectively mitigate the replica flood attack.

6.PERFORMANCE EVALUATIONS

6.1 Routing Algorithms and Metrics

We use the following routing protocols in evaluations:

Forward. A single-copy routing protocol where a packet is forwarded to a relay if the relay has more frequent contacts with the destination.

SimBet [8]. A single-copy routing protocol where a packet is forwarded to a relay if the relay has a higher simbet metric, which is calculated from two social measures (similarity and betweenness).

Spray-and-focus [19]. It is similar to Spray-and- Wait, but each packet copy is individually routed to the destination with Forward.

Propagation. A packet is replicated to a relay if the relay has more frequent contacts with the destination.

We use the following performance evaluation metrics:

Detection rate. The proportion of attackers that are detected out of all the attackers.

Detection delay. From the time the first invalid packet is sent to the time the attacker is detected.

Computation cost. The average number of signature generations and verifications per contact.

Communication cost. The number of P-claim/ T-claim pairs transmitted into the air, normalized by the number of packets transmitted.

Storage cost. The time-averaged kilobytes stored for P-claims and T-claims per node

6.2 Cost

To evaluate the cost of our scheme in a steady state (i.e., all attackers have been detected), no attackers are deployed in this group of simulations. The Reality trace is used. Packets are generated between the 61st and 120th day of the trace, and statistics are collected from the 91th day. By default, each node generates two packets per day, parameter (i.e., the time a claim is stored) is 30 days and K is 10. In a contact, a node may receive some packets but then immediately drop them due to buffer overflow. In such cases, the transmission of the claims attached to these packets is counted into the communication overhead, and the signature generations for these claims are counted into the computation overhead. Since the receiver does not buffer these packets, it does not store these claims or verify their signatures. We first evaluate the computation cost of our scheme

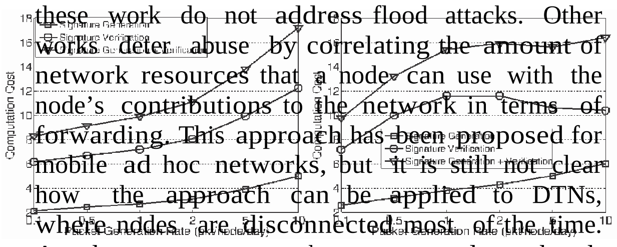


Fig 5.The Computation cost of our scheme

7 RELATED WORK

Our scheme bears some similarity with previous approaches that detect node clone attacks in sensor networks. Both rely on the identification of some kind of inconsistency to detect the attacker. However, their approaches assumes consistent connectivity between nodes which is unavailable in DTNs. Also, they do not handle the long delays of detection.

A few recent works also address security issues in DTNs. It also Address blackhole attack in which malicious nodes forge routing metrics to attract packets and drop all received packets. Their approach uses a primitive called encounter ticket to prove the existence of contacts and prevent the forgery of routing metrics, but encounter ticket cannot be used to address flood attacks. Li and Cao also proposed a distributed scheme to mitigate packet drop attacks, which works no matter if the attackers forge routing metrics or not. Ren et al. studied wormhole attacks in DTNs. Chen and Choon proposed a credit-based approach and Shevade et al. proposed a gaming-based approach to provide in- centives for packet forwarding. Privacy issues have also be addressed. However, these work do not address flood attacks. Other works deter abuse by correlating the amount of network resources that a node can use with the node's contributions to the network in terms of forwarding. This approach has been proposed for mobile ad hoc networks, but it is still not clear how the approach can be applied to DTNs, where nodes are disconnected most of the time. Another recent work proposed a batch authentication protocol for DTNs, which verifies multiple packet signatures in an aggregated way to save the computation cost. This work is complementary to ours,

8 CONCLUSION

In this paper, we employed rate limiting to mitigate flood attacks in DTNs, and proposed a scheme which exploits claim-carry-and-check to probabilistically detect the violation of rate limit in DTN environments. Our scheme uses efficient constructions to keep the computation, commu- nication and storage cost low. Also, we analyzed the lower bound and upper bound of detection probability. Extensive trace-driven simulations showed that our scheme is effective to detect flood attacks and it achieves such effectiveness in an efficient way. Our scheme works in a distributed manner, not relying on any online central authority or infrastructure, which well fits the environment of DTNs. Besides, it can tolerate a small number of attackers to collude.

9.REFERENCES

[1] Qinghus Li,Wei Gao,Sencun Zhu & Guohong Cao "To Le or to Comply:Defending against Flood Attacks in Disruption Tolerant Networks" , 2013

[2] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," Proc. ACM SIGCOMM, pp. 27-34, 2003.

[3] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," Proc. ACM SIGCOMM, 2005.

[4] M. Motani, V. Srinivasan, and P. Nuggehalli, "PeopleNet: Engineering a Wireless Virtual Social Network," Proc. MobiCom,pp. 243-257, 2005.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for Vehicle-Based Disruption-Tolerant Networks," Proc. IEEE INFOCOM, 2006.

[6] S.J.T.U.Grid Computing Center, "Shanghai Taxi Trace Data,"http://wirelesslab.sjtu.edu.cn/, 2012.

[7] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, Internet Denial of Service: Attack and Defense Mechanisms. Prentice Hall, 2005.

[8] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," Proc. IEEE First Int'l Workshop Sensor Network Protocols and Applications, 2003.

[9] E. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs," Proc. MobiHoc, pp. 32-40,2007.

[10] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in Delay Tolerant Networks: A Social Network Perspective," Proc. ACM MobiHoc, 2009.

[11] F. Li, A. Srinivasan, and J. Wu, "Thwarting Blackhole Attacks in Distruption-Tolerant Networks Using Encounter Tickets," Proc. IEEE INFOCOM, 2009.

[12] Y. Ren, M.C. Chuah, J. Yang, and Y. Chen, "Detecting Wormhole Attacks in Delay Tolerant Networks," IEEE Wireless Comm. Magazine, vol. 17, no. 5, pp. 36-42, Oct. 2010.

[13] U. Shevade, H. Song, L. Qiu, and Y. Zhang, "Incentive-Aware Routing in DTNS," Proc. IEEE Int'l Conf. Network Protocols (ICNP '08), 2008.

[14] Q. Li and G. Cao, "Mitigating Routing Misbehavior in Disruption Tolerant Networks," IEEE Trans. Information Forensics and Security, vol. 7, no. 2, pp. 664-675, Apr. 2012

[15] B. Raghavan, K. Vishwanath, S. Ramabhadran, K. Yocum, and A.Snoeren, "Cloud Control with Distributed Rate Limiting," Proc. ACM SIGCOMM, 2007.

[16] F-SECURE, "F-Secure Malware Information Pages: Smsworm:- Symbos/Feak," http://www.f-secure.com/v-descs/smsworm symbos feak.shtml, 2012.

[17] N. Eagle and A. Pentland, "Reality Mining: Sensing ComplexSocial Systems,"

Personal and Ubiquitous Computing, vol. 10, no. 4, pp. 255-268, 2006.

[18] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," Proc. IEEE INFOCOM, 2010.

[19] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," IEEE/ACM Trans. Networking, vol. 16, no. 1, pp. 77-90, Feb. 2008.

[20] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," ACM SIGMOBILE Mobile Computing and Comm. Rev., vol. 7, no. 3, pp. 19-20, 2003.

[21] W. Gao and G. Cao, "On Exploiting Transient Contact Patterns for Data Forwarding in Delay Tolerant Networks," Proc. IEEE 18th Int'l Conf. Networks Protocols (ICNP), 2010.

[22] J. Burgess, G.D. Bissias, M. Corner, and B.N. Levine, "Surviving Attacks on Disruption-Tolerant Networks without Authentica- tion," Proc. ACM MobiHoc, 2007.

[23] S.C. Nelson, M. Bakht, and R. Kravets, "Encounter-Based Routing in Dtns," Proc. IEEE INFOCOM, pp. 846-854, 2009.

[24] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," Proc. ACM SIGCOMM, pp. 252-259, 2005.

[25] B. Chen and C. Choon, "Mobicent: A Credit-Based Incentive System for Disruption Tolerant Network," Proc. IEEE INFOCOM,2010.