International Journal of Research Studies in Science, Engineering and Technology
Volume 1, Issue 9, December 2014, PP 149-154
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Defending Against Flood Attacks in Disruption Tolerant Networks

[1] Ms. L.Dhanam

Assistant Professor
Department of CSE,
SCAD Institute of Technology

[2] Ms. Anuradha Balasubramaniam

Department of ECE,
INFO Institute of Technology

Abstract: *Disruption Tolerant Networks (DTNs) utilize the mobility of nodes and the opportunistic contacts among nodes for data communications. Due to the limitation in network resources such as contact opportunity and buffer space, DTNs are vulnerable to flood attacks in which attackers send as many packets or packet replicas as possible to the network, in order to deplete or overuse the limited network resources. In this paper, we employ rate limiting to defend against flood attacks in DTNs, such that each node has a limit over the number of packets that it can generate in each time interval and a limit over the number of replicas that it can generate for each packet. We propose a distributed scheme to detect if a node has violated its rate limits. To address the challenge that it is difficult to count all the packets or replicas sent by a node due to lack of communication infrastructure, our detection adopts claim-carry-and-check: each node itself counts the number of packets or replicas that it has sent and claims the count to other nodes; the receiving nodes carry the claims when they move, and cross-check if their carried claims are inconsistent when they contact.*

Index Terms: *DTN, flood attacks, packet rate limiting, security*

1. INTRODUCTION

DTNs enable data transfer when mobile nodes are only intermittently connected, making them appropriate for applications where no communication infrastructure. Due to lack of consistent connectivity, two nodes can 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. 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. Due to limitation in bandwidth and buffer space, DTN is vulnerable to flood attacks. Simply we call the two types of attacks as packet flood attack and packet replica attack.

Our basic idea of detection is claim-carry-and-check. Each node itself checks the count of number of packets sent and the replicas of packets sent and claims the count to other nodes.

2. MOTIVATION

2.1 The Effect of Flood Attacks

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 (according to the routing algo-rithm) 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 [8], Spray-and-Focus [19] (three copies allowed for each packet) 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.

A replica flood attacker replicates the packets it has generated to every encountered node that does not have a copy. Each good node generates thirty packets on the 121st day of the Reality trace, and each attacker does the same in replica flood attacks. Each packet expires in 60 days. The buffer size of each node is 5 MB, bandwidth is 2 Mbps and packet size is 10 KB. Two metrics are used, The first metric is packet delivery ratio, which is defined as the fraction of packets delivered to their destinations out of all the unique packets generated. The second metric is the fraction of wasted transmissions (i.e., the transmissions made by good nodes for flooded packets). The higher fraction of wasted transmissions, the more network resources are wasted. We noticed that the effect of packet flood attacks on packet delivery ratio has been studied by Burgess et al. [22] using a different trace [4]. Their simulations show that packet flood attacks significantly reduce the packet delivery ratio of single-copy routing but do not affect propagation routing much. However, they do not study replica flood attacks and the effect of packet flood attacks on wasted transmissions.

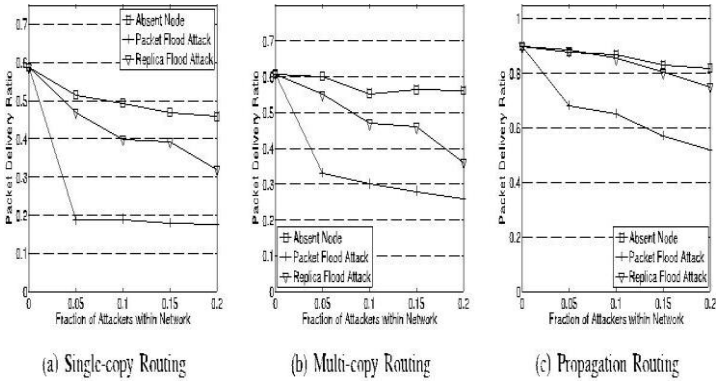


Fig. 1. *The effect of flood attacks on packet delivery ratio. In absent node, attackers are simply removed from the network. Attackers are selectively deployed to high-connectivity nodes.*

3. PROBLEM DEFINITION

3.1 Defence 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.

3.2 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. A node's limit l for a buffered packet is determined by the routing protocol. In multicopy routing, l ¼ L^0 (where L^0 is a parameter of routing) if the node is the source of the packet, and l ¼ 1 if the node is an intermediate hop (i.e., it received the packet from another node). In single-copy routing, l ¼ 1 no matter if the node is the source or an intermediate hop. Note that the two limits L and l do not depend on each other.

3.3 Setting the Rate Limit L

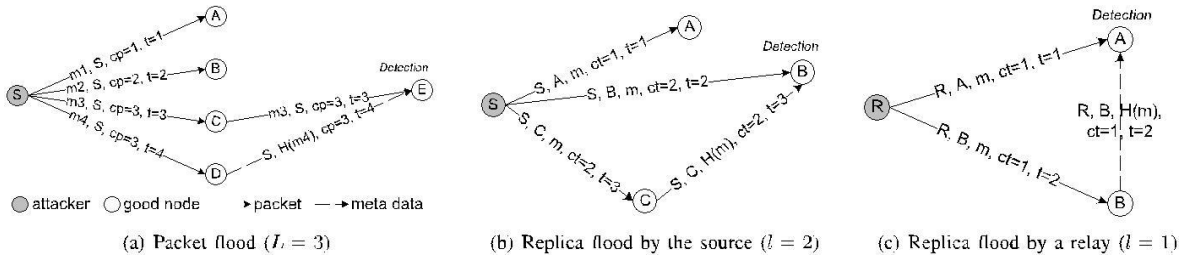
One possible method is to set L in a request-approve style. When a user joins the network, she requests for a rate limit from a trusted authority which acts as the network operator. In the request, this user specifies an appropriate value of L based on prediction of her traffic demand. If the trusted 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. The flexibility of rate limit leaves legitimate users' usage of the network unhindered. This process can be similar to signing a contract between a smartphone user and a 3G service provider: the user selects a data plan (e.g., 200 MB/month) and pays for it; she can upgrade or downgrade the plan when needed.

3.4 Models and Assumptions

3.4.1 Network Model

For simplicity, we assume that all packets have the same predefined size. Although in DTNs the allowed delay of packet delivery is usually long, it is still impractical to allow unlimited delays. Thus, we assume that each packet has a lifetime. 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.We also assume that time is loosely synchronized, such that any two nodes are in the same time slot at any time. Since the intercontact time in DTNs is usually at the scale of minutes or hours, the time slot can be at the scale of one minute. Such loose time synchronization is not hard to achieve.



(a) Packet flood ($l = 3$) (b) Replica flood by the source ($l = 2$) (c) Replica flood by a relay ($l = 1$)

Fig. 2. *The basic idea of flood attack detection. cp and ct are packet count and transmission count, respectively. The arrows mean the transmission of packet or metadata which happens when the two end nodes contact.*

3.5 Basic Idea: Claim-Carry-and-Check

3.5.1 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. However, since the node may send its packets to any node it contacts at any time and place, no other node can monitor all of its sending activities. To address this challenge, 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 (together with a little auxiliary information such as its ID and a timestamp) 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. The claimed count must have been used before by the attacker in another claim, which is guaranteed by the pigeonhole principle, and these two claims are inconsistent. The nodes which have received packets from the attacker carry the claims included in those packets when they move around. When two of them contact, they check if there is any inconsistency between their collected claims. The attacker is detected when an inconsistency is found. in Fig. 2a. S is an attacker that successively sends out four packets to A, B, C, and D, respectively. Since L ¼ 3, if S claims the true count 4 in the fourth packet m4, this packet will be discarded by D. Thus, S dishonestly claims the count to be 3, which has already been claimed in the third packet m3. m3 (including the claim) is further forwarded to node E.

3.5.2 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 ½1; 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. Examples are given in Figs. 2b and 2c.

4 OUR SCHEME

4.1 Claim Construction

Two pieces of metadata are added to each packet (see Fig. 4), 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 hasbeen 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,H(m),$SIG_{s,}$(H(H(m)|S|C_p|t))

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 ct 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(m),C_t,t,SIG_A,(H(A|B|H(m)|c_t|t))

4.2 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:

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, respectively)as an attacker and add it into a blacklist;

15:Disseminate an alarm against the attacker to thenetwork;

16:else

17:Store the new P-claim (T-claim, respectively);

18:end if

19:end if

5 COLLUSION ANALYSIS

5.1 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.

5.2 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-wait [15]: A multicopy protocol, where the source replicates a packet to L0 ¼ 3 relays and each relay directly delivers its copy to the destination when they contact.

Spray-and-focus [15]: 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.

7 CONCLUSIONS

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.

REFERENCES

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

[2] 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.

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

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

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

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

[7] 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.

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

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

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

[11] 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.

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

[13] 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.

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

[15] 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.