IJCAT International Journal of Computing and Technology, Volume 1, Issue 2, March 2014
ISSN : 2348 - 6090
www.IJCAT.org

A Distributed Method to Protect Disruption Tolerant Networks from the Effects of Packet Overflow

[1]Shwetha T J, [2]Resmi S

[1] Computer Science & Engineering, Atria Institute of Technology
Bangalore, Karnataka, India

[2] Computer Science & Engineering, Atria Institute of Technology
Bangalore, Karnataka, India

Abstract - Disruption Tolerant Networks (DTNs) are characterized by lack of connectivity, resulting in a lack of instantaneous end-to-end paths. They utilize mobility of nodes and opportunistic contacts among the nodes for data communications. Resources considered in this paper are contact opportunity and buffer space. Due to limitations of these in DTN, they are vulnerable to flood attacks caused by the overflow of packets. Attackers send as many packets or packet replicas as possible to the network to overuse the resources. This overflow reduces battery life of nodes, degrades network service provided to good nodes. In this paper, we employ rate limiting to defend against flood attacks, such that each node has a limit over the number of packets and replicas of each packet that it can generate in each time interval. We propose a distributed scheme to detect if a node has violated its rate limits. The detection adopts claim-carry-and-check.

Keywords **- DTN, Overflow, Defense, Claim, Pigeonhole.**

1. Introduction

Disruption Tolerant Networks (DTNs) [1] consists of mobile nodes carried by human beings [2], [3], vehicles [4], [5], etc. 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"; i.e., when a node receives some packets, it stores these packets in its buffer, carries them around until it contacts another node, and then forwards them. Due to the limitation in bandwidth and buffer space, DTNs are vulnerable to flood attacks caused by packet overflow. In flood attacks, maliciously or selfishly motivated attackers inject as many packets as possible into the network which is called packet flood attack, or the attackers forward replicas of the same packet to as many nodes as possible which is called replica flood attack. This overflow of packets can waste the precious bandwidth and buffer resources, degrade the network services provided to good nodes and can also shorten their battery life.

Although many schemes have been proposed to defend against this overflow on the Internet and in wireless sensor networks, they assume persistent connectivity and cannot be directly applied to DTNs that have intermittent connectivity. In DTNs, little work has been done on this, despite the many works on routing data dissemination, black hole attack wormhole attack, and selfish dropping behavior. We noted that the packets overflowed by outsider attackers (i.e., the attackers without valid cryptographic credentials) can be easily filtered with authentication techniques. However, authentication alone does not work when insider attackers with valid cryptographic credential flood packets and replicas with valid signatures. Thus, it is still an open problem to address flood attacks in DTNs. We employ rate limiting [6] 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 (the number of nodes that it can forward each packet to). The two limits are used to mitigate packet overflow attack and replica overflow 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. It is challenging to do this in DTNs due to lack of

communication infrastructure and consistent connectivity. Since a node moves around and may send data to any contacted node, it is very difficult to count the number of packets or replicas sent out by this node. Our basic idea of detection is claim-carry-and-check.

1.1 Motivation

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.

2. Overview

2.1 Problem Definition

Defense against Packet Overflow 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 overflow 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. A node's rate limit L does not depend on any specific routing protocol, but it can be determined by a service contract between the node and the network operator.

Defense against Replica Overflow Attacks: Defense against replica flood considers single-copy and multi-copy 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.

Setting the Rate Limit L: One possible method is to set L in a request-approve style. When a user joins the network, he/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 unreasonably large rate limits, a user pays an appropriate amount of money or virtual currency for her rate limit.

2.2 Basic Idea: Claim-Carry-Check

Packet Overflow Detection: To detect the attackers who 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 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, (which states that if A items are put into B pigeonholes with A >B, then at least one pigeonhole must contain more than one item) 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.

Replica Overflow 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.

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. 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 overflow attacks, the attacker can be detected.

3. Modules

Our Proposed work has the following modules.

3.1 DTN Network Creation

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 synchronised, such that any two nodes are in the same time slot at any time. Since inter contact time in DTN is usually at the scale of minutes or hours, the time slot can be at the scale of one minute. Such loose time synchronisation is not hard to achieve.

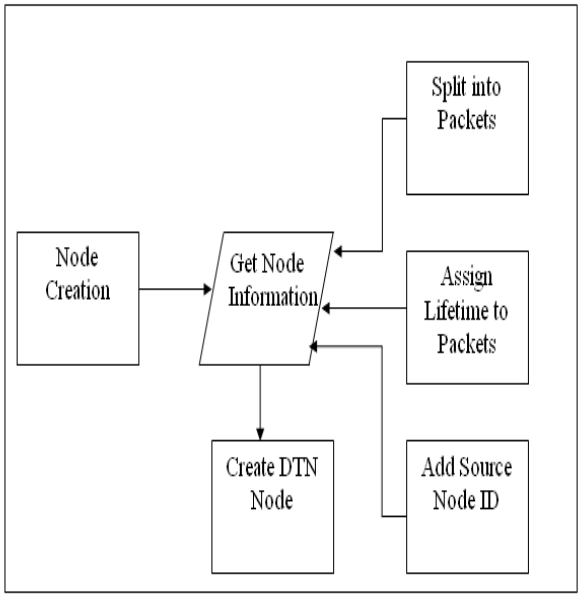

Fig. 1 DTN network creation

3.2 Rate Limit Certificate Creation

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. 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.

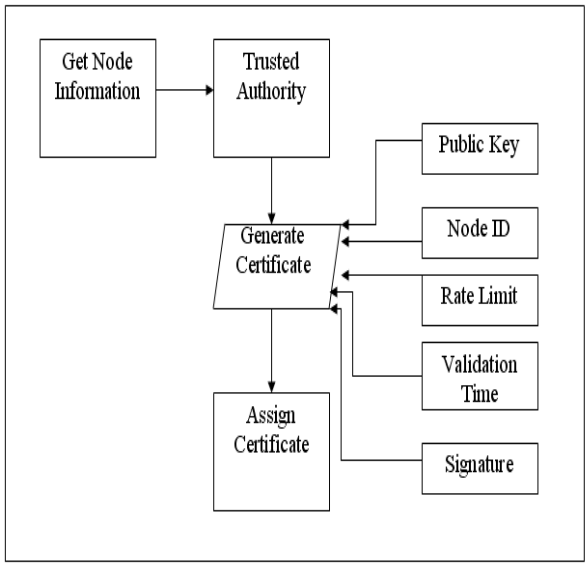

Fig. 2 Rate limit certificate creation

3.3 Claim Construction

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.

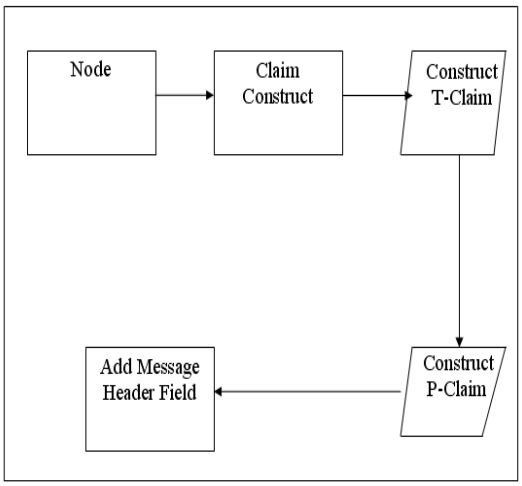

Fig. 3 Claim construction

3.4 Inconsistency Analysis

The inconsistency check based on compact P-claims does not cause false positive, since a good node never reuses any count value in different packets generated in the same interval. The inconsistency check may cause false negative if the two inconsistent P-claims have the same hash remainder. The inconsistency check based on compact T-claims does not cause extra false negative. False positive is possible but it can be kept low. We consider inconsistency check against compactly stored claims.

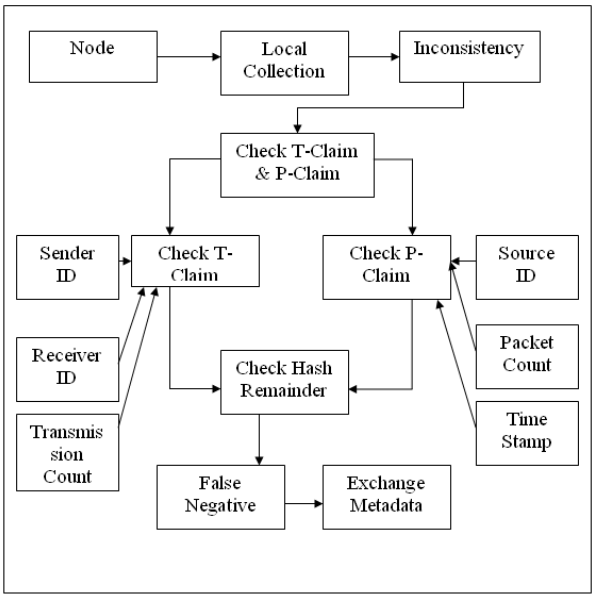

Fig. 4 Inconsistency analysis

3.5 Metadata Exchange Process

When two nodes contact they exchange their collected P-claims and T-claims to detect flood attacks. If all claims are exchanged, the communication cost will be too high. Thus, our scheme uses sampling techniques to keep the communication cost low. To increase the probability of attack detection, one node also stores a small portion of claims exchanged from its contacted node, and exchanges them to its own future contacts. This is called redirection. Each node maintains two separate sets of P-claims, T-claims, for metadata exchange, a sampled set which includes the P-claims sampled from the most recent contacts with K different nodes and a redirected set which includes the P-claims redirected from those contacts. Both sets include Z P-claims obtained in each of those contacts. When analysing detection probability, we assume that each attacker acts alone.

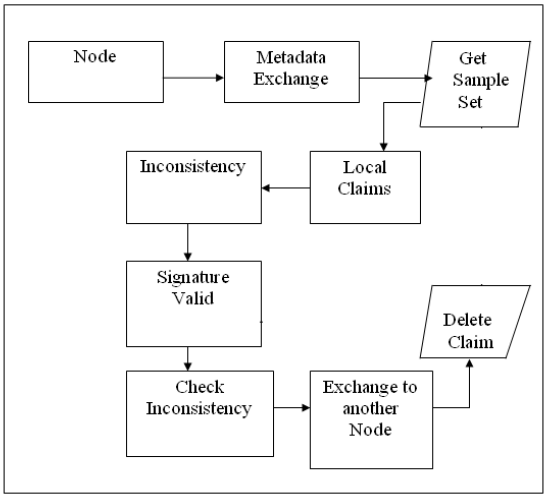

Fig. 5 Inconsistency analysis

3.6 Verification Process

To better detect flood attacks, the two nodes also exchange a small number of the recently collected P-claims and T-claims and check them for inconsistency. When a node detects inconsistency and finds out that sending node is an attacker, it adds the attacker into a blacklist and will not accept packets originated from or forwarded by the attacker.

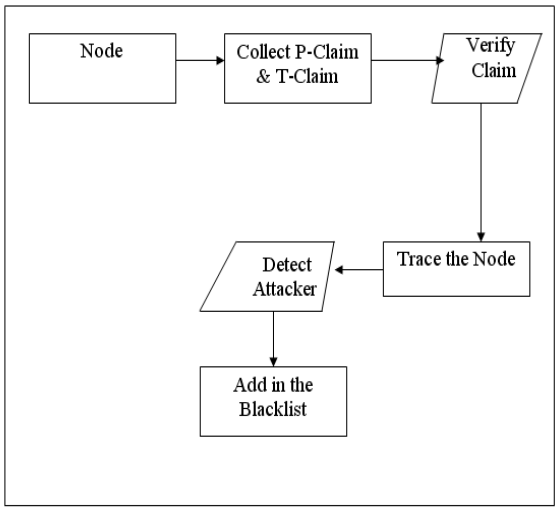

Fig. 6 Verification process

4. Cost Analysis

4.1 Metadata Exchange Process

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. As to the latter, at most 4ZK P-claims and 4ZK T-claims are exchanged in each contact, with one half for sampled and the other half for redirected claims.

4.2 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.

4.3 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, whichever is later, and then compacted.

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. 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. When there is no defence, 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. Routing Algorithms

We use the following routing protocols in evaluations:

. Forward is a single-copy routing protocol where a packet is forwarded to a relay if the relay has more frequent contacts with the destination.
. SimBet [7] is 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.
. Spray-and-wait [8] is a multi copy protocol, where the source replicates a packet to L'= 3 relays and each relay directly delivers its copy to the destination when they contact.
. Spray-and-focus [8] is a 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.

7. 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, communication and storage cost low. 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] Qinghua Li, Sencun Zhu "To lie or to Comply: Defending against Flood Attacks in Disruption Tolerant Networks" IEEE Transactions on Dependable and Secure Computing, Vol 10, No. 3, pp 168-182, 2013.

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

[8] 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.

Shwetha T J B.E in Information Science in the year 2006 , MTech (pursuing) Computer Science & Engineering in Atria Institute of Technology. Previously worked as Software Engineer in Pro Mind Technologies (2007-2008), then as lecturer in Brindavan College of Engineering (2009-2011).

Resmi S Mtech (CSE), currently working as Assistant Professor in Atria Institute of Technology.