ISSN: 2321-7782 (Online)
Volume 2, Issue 8, August 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Flood Attacks Mitigation in DTNs Using MD5 Algorithm

Poojitha Singu[1]
PG Full Time Student, CSE Department
Sree Vidyanikethan Engineering College(Autonomous)
A.Rangampet,Tirupathi,(A.P) – India

B.Gurunadha Rao[2]
Assistant Professor, CSE Department
Sree Vidyanikethan Engineering College(Autonomous)
A.Rangampet,Tirupathi,(A.P) – India

N.Bharatha Lakshmi[3]
PG Full Time Student, CSE Department
Sree Vidyanikethan Engineering College(Autonomous)
A.Rangampet,Tirupathi,(A.P) – India

Abstract: ***Disruption Tolerant Networks is network architecture which provides communication between two nodes in unstable or stressed environment areas. Due to limited network resources such as buffer space and bandwidth, these DTNs are easily vulnerable to flood attacks in attacker sends as many packets or replicas to overuse the limited network resources. To overcome these they employ Rate limiting to defend against flood attacks in DTNs. By using this rate limit each and every node over a particular packet so that if any node exceed rate limit then that node can be discarded. They propose a distributed scheme to identify attacker if they exceed rate limit. It is difficult to Identify / Track the address of the attacker node. For that the basic idea of detection is claim-carry-and check method. In this method each node itself counts the number of replicas or packets which was sent and claims the information to neighbour node, then receiving nodes carry the claims information when they move to other neighbouring node. And then cross-check these claim whether the carried node consistent or inconsistent. If inconsistent detected then discard that node and add to Blacklist. By using rate limit certificate we can find flood attacker when they exceed rate limit of that attacker node. To overcome this proposed uses MD5 Function. In which it will generates 32-bit hash key for the node who wants to send packets less than rate limit.32-bit hash key generation based on MD5 algorithm. Based on 32-bit hash keys, attackers who sends packet within the rate limit can also be easily identified.***

Keywords: Disruption Tolerant Network, claim-carry-and-check, flood attacks, Detection.

I. INTRODUCTION

Disruption Tolerant Networks is mainly used for data transfer between mobile nodes which carried by human beings vehicles etc. DTNs provides data transfer when mobile nodes are only intermittently connected, making them applicable for applications wherever no communication infrastructure is available such as military situation and rural areas. Due to this inconsistency, two nodes can transfer data when they enter into communication range of each other. Data is transferred via store-carry-forward method. This approach nodes store packets if they cannot find a next-hop node to deliver them to destinations. The each node first stores packets in its memory and then selectively transmits packets when it encounters other nodes based on various metrics including the last encounter time, the numbers of previous encounters, and the estimated packet delivery probability values to other nodes. Such metrics are derived from information provided by forwarding nodes themselves and it is hard to verify due to the network sparseness as well as the intermittent connectivity between nodes.

However DTN's has limitations such as low bandwidth and buffer space. Due to this they are liable to flood attacks. A flood attack is one in which the attackers send as many packet into the network and overuse the limited resources. Two types of flood attacks are packet flood attack and replica flood attack.

Also, mobile nodes may have limited buffer space. Due to the limitation 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 begin 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. There are many methods to prevent flood attacks, but none has been inducted for DTN's. A flood attack caused by outsider (unauthorized) can be prevented by authentication techniques. However it is not possible to prevent for attack caused by insiders (authorized). Thus, it is still an open problem is to address flood attacks in DTNs.

In this paper, they employ rate limiting to defend against flood attacks in DTNs. In our approach, each and every 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 telecommunication 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. 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. Each node itself counts the number of packets or replicas that it has sent out, and claims the count to other nodes; the receiving nodes carry the claims around when they move, exchange some claims when they contact, and cross-check if these claims are inconsistent. If an attacker floods more packets or replicas than its limit, it has to use the same count in more than one claim according to the pigeonhole principle,1 and this inconsistency may lead to detection. Based on this idea, we use different cryptographic constructions to detect packet flood and replica flood attacks.

Because the contacts in DTNs are opportunistic in nature, our approach provides probabilistic detection. The more traffic an attacker floods, the more likely it will be detected. The detection probability can be flexibly adjusted by system parameters that control the amount of claims exchanged in a contact. In which it will generates 32-bit hash key for the node who wants to send packets less than rate limit.32-bit hash key generation based on MD5 algorithm. Based on 32-bit hash keys, attackers who sends packet within the rate limit can also be easily identified. The effectiveness and efficiency of our scheme are evaluated with extensive trace-driven simulations.

This paper is structured as follows. Section 2 motivates our work. Section 3 presents our models and basic ideas. Sections 4 and 5 present our scheme. Section 6 presents security and cost analysis. Section 7 presents simulation results. The last two sections present related work and conclusions, respectively.

II. EXISTING SYSTEM

In Existing System Store-and-forward method is used. Store-and-forward approach nodes store packets if they cannot find a next-hop node to deliver them to destinations. The each node first stores packets in its memory and then selectively transmits packets when it encounters other nodes based on various metrics including the last encounter time, the numbers of previous

encounters, and the estimated packet delivery probability values to other nodes. Such metrics are derived from information provided by forwarding nodes themselves and it is hard to verify due to the network sparseness as well as the intermittent connectivity between nodes.

The main contribution is a technique to detect if a node has violated its rate limits. Each node itself counts the number of packets or replicas that it has sent out, and claims the count to other nodes. The receiving nodes carry the claims around when they move, exchange some claims when they contact, and cross-check if these claims are inconsistent.

III. PROPOSED SYSTEM

In our proposed system, 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 Telecommunication 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. 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.

Each node itself counts the number of packets or replicas that it has sent out, and claims the count to other nodes; the receiving nodes carry the claims around when they move, exchange some claims when they contact, and cross-check if these claims are inconsistent. If an attacker floods more packets or replicas than its limit, it has to use the same count in more than one claim according to the pigeonhole principle and this inconsistency may lead to detection. 32-bit hash key generation based on MD5 algorithm. Based on 32-bit hash keys, attackers who sends packet within the rate limit can also be easily identified. The effectiveness and efficiency of our scheme are evaluated with extensive trace-driven simulations.

IV. BASIC IDEA

To observe the attackers who violate the rate limit L, we must count the amount of unique packets that every node as a source has generated and sent to the network among the present interval. However, since the node might send its packets to any node it contacts at any time and place, no alternative node can monitor all of its sending activities. To deal with this challenge, our plan is to let the node itself count the amount of unique packets that it has sent out, as a source node and claim the up-to-date packet count in every packet which is sent out. The node's rate limit certificate is additionally hooked up to the packet, such that other alternative nodes receiving the packet will learn its approved rate limit L. If an attacker is flooding a lot of packets than its rate limit, therefore it is a clear indicator of attack. If the claimed counts have been used before by the attacker in another claim, which is secured by the pigeonhole principle, and these two claims cause inconsistent. When the node received packets from the attacker, it carry those claims enclosed in those packets after they move around. Once two of the nodes contact, they check if there is any inconsistency between their collected claims. The attacker is detected once associate inconsistency is found.

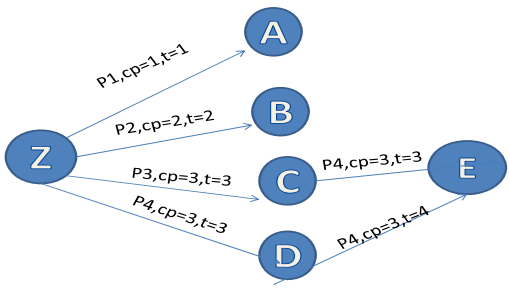

Fig 1. Packet Flood Detection

In the Fig. 1, Let's consider Z is as an attacker who injects 4 packets to nodes A, B, C, D. Where L is a Rate limit i.e. L = 3, cp is a packet count, t is a transmission count, If Z claims that the count value is 4 in p4, then that packet will be discarded, since rate limit included in the packet is 3,So Z dishonestly claims count to be 3, which is same as p3 . P3 packet is forwarded to E. When D and E contact, it acknowledges that two packets have the same count value. Therefore it detects that Z is an attacker and discards the packets and notify other nodes about the attacker.

V. MODELS

- Network Model
- Adversary Model
- Trust Model

Network Model:

In DTNs, since contact durations will be short, a large data item is always split into smaller packets to facilitate data transfer. For simplicity, we have a tendency to assume that all packets have the similar predefined size. Though in DTNs the allowed delay of packet delivery is typically long, it is still impractical to permit unlimited delays. Thus, we assume that every packet has a lifetime. The packet becomes unimportant when its lifetime ends and will be is carded. We assume that every packet generated by nodes is unique. This can be implemented by supplying the source node ID and a domestically distinctive sequence range that is assigned by the source for this packet, in the packet header. We also assume that time is loosely synchronic, specified any two nodes are within the same time slot at any time. Since the intercontact time in DTNs is often at the dimensions of minutes or hours, the time slot can be at the dimensions of one minute. Such loose time synchronization is not difficult to achieve.

Adversary Model:

There are a variety of attackers within the network. An attacker can flood packets or replicas. On flooding packets, the attacker behaves as a source node. It creates and injects a lot of packets into the network than its rate limit L. When flooding replicas, the attacker sends its buffered packets (which can be generated by itself or received from other nodes) more than its limit l for other nodes. The attackers can also be insiders with valid cryptographic keys. Some attackers may also collude and communicate via outband channels.

Trust Model:

We assume that a public-key cryptography system is accessible. For example, Identity-Based Cryptography (IBC) [9] has been shown to be practical for DTNs [11]. In IBC, only an offline Key Generation Center (KGC) is needed. KGC generates a non-public key for every node. Except the KGC, no party will generate the non-public key for a node id. With this type of system, an attacker can't forge a node identification and non-public key pair. Also, attackers do not know the non-public of a honest node (not attacker). Every node has a rate limit certificate obtained from a trustworthy authority. The certificate includes the node's ID, its secure rate limit L, the life time of this certificate and the trustworthy authority's signature. The rate limit certificate can be united into the general public key certificate or stand alone.

VI. CLAIM CONSTRUCTION

Two pieces of metadata units are added to every packet

1. Packet Count Claim (P-claim) and
2. Transmission Count Claim (T-claim)

P-claim is added by the source node and transmitted to later hops together with the packet. T-claim is generated by every node and processed hop-by-hop. Specifically, the source itself generates a T-claim and appends it to the packet. Once the first hop receives this packet, it peels off the received T-claim; once it forwards the packet out, it appends a brand new T-claim to the packet. This technique continues in later hops. Every hop keeps the P-claim of the source and also the T-claim of its previous hop to notice attacks.

P-Claim: When a source node S sends a brand new packet m (which is generated by S and not sent out before) to a contacted node, it generates a P-claim. The P-claim is hooked up to packet m as a header field, and will be forwarded together with the packet to later hops. Once the contacted node receives this packet, it verifies the signature within the P-claim, and checks the worth of cp(packet count). If cp is larger than L, it discards this packet; otherwise, it stores this packet and also the P-claim.

T-Claim: When node A transmits a packet m to node B, it attaches a T-claim to m. The T-claim consists of A's current transmission count ct for m (i.e., the number of times it has transmitted m out t to n number of nodes) and also the current time t.

VII. Inconsistency Caused by Attack

In a dishonest P-claim, an attacker uses a smaller packet count than the real value. (We don't take into account the case wherever the attacker uses a much bigger packet count than the real value, since it doesn't make any sense for the attacker.) However, this packet count has been used in another P-claim generated earlier. This causes an inconsistency referred to as count reuse, which implies the utilization of the identical count in two completely different P-claims generated by the identical node.

VIII. Algorithm

Algorithm**:** The protocol runs by each node in a contact

//Metadata (P-claim and T-claim) exchange and attack detection

1: if Have packets to send then

2: For each new packet, generate a P-claim;

3: For all packets, generate their T-claims and sign them with a hash tree;

4: Send every packet with the P-claim and T-claim attached;

5: end if

6: if Receive a packet then

7: if Signature verification fails or the count value in its P-claim or T-claim is invalid then

8: Discard this packet;

9: end if

10: Check the P-claim against those locally collected and generated in the same time interval to detect inconsistency;

11: Check the T-claim against those locally collected for inconsistency;

12: if Inconsistency is detected then

13: Tag the signer of the P-claim (T-claim, respectively) as an attacker and add it into a blacklist;

14: Disseminate an alarm against the attacker to the network;

15: else

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

17: end if

18: end if

IX. PERFORMANCE EVALUATIONS

The following are the performance evaluation metrics:

- Detection rate: The ration of attackers that are observed out of all the attackers.
- Detection delay: From the time the primary invalid packet is sent to the time the attacker is observed.
- Computation cost: The typical range of signature generations and analysis per contact.
- Communication cost: The amount of P-claim/T-claim pairs transmitted into the air, normalized by the amount of packets transmitted.
- Storage cost: The time-averaged kilobytes stored for P-claims and T-claims per every node.

X. EXPECTED RESULT

The followings are the results which will be analyzed in this system.

Communication cost: The communication cost mainly has two components. One component is that the P-claim and T-claim transmitted with every packet, and the alternative component is that the partial claims transmitted during metadata exchange. As to the latter, at the most P-claims and T-claims are exchanged in every contact, with one half for sampled and the other half for redirected claims.

Storage cost: Most P-claims and T-claims are compacted when the packets are forwarded. The sampled P-claims and T-claims are stored in full until the packets are forwarded or are exchanged to K nodes, whichever is later, and then compacted. For every received packet, less than 20 bytes of compact claims are stored for restricted time duration.

Collusion Analysis: One attacker might send a packet with a dishonest packet count to its colluder, which is able to forward the packet to the network. Certainly, the colluder won't exchange the dishonest P-claim with its contacted nodes. However, so as long as the colluder forwards this packet to a honest node, this honest node has a chance to observe the dishonest claim as well as the attacker.

XI. CONCLUSION

In this paper, we used rate limiting to mitigate flood attacks in DTNs, and design a scheme that exploits claim-carry-and-check to probabilistically identify the violation of rate limit in DTN environments. This scheme uses efficient constructions to keep the computation, communication and storage price low. Also, MD5 Algorithm which gives effective security while transmit data from hop to hop. This scheme works in a distributed manner, not relying on any online central authority or infrastructure, which well fits the environment of DTNs. The usage of the proposed is mainly in the case of military and rural areas scenario.

References

1. K. Fall "A Delay-Tolerant Network Architecture for Challenged Internets," proc.ACM SIGCOMM, 2003.
2. A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, P. Hui, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," proc.ACM SIGCOMM, 2005.
3. M. Motani, V. Srinivasan, and P. Nuggehalli, "People Net:Engineering a Wireless Virtual Social Network," Proc.MobiCom, 2005.
4. G.D. Bissias, M. Corner, J. Burgess, and B.N. Levine "Maxprop :Routing for Vehicle-based Disruption Tolerant Network,"Proc. IEEE INFOCON, pp 1-11, 2006.

5. Zhi-Jun Li, Shou-Xu Jiang “Planning the Mobility of Routing Ferries for Intermittently Connected Mobile Networks,” in ICST International Conference, 2011.
6. A.Afanasyev, P. Mahadevan, I.Moiseenko,E.Uzun,and L.Zhang, ”Internet flooding attack and countermeasures in Named Data Networking,” in IFIP Network Conference, 2013.
7. R. Bhatnagar and U. Shankar,"The proposal of hybrid intrusion detection for defence of sync flood attack in wireless sensor network," International Journal of Computer Science & Engineering Survey, vol. 3, pp. 31-38, 2012.
8. Barath Raghavan, Kashi Vishwanath, Sriram Ramabhadran, Kenneth Yocum, and Alex C. Snoeren, “Cloud Control with Distributed Rate Limiting,” SIGCOMM, 2007.
9. C. Gentry and A. Silverberg, “Hierarchical Id-Based Cryptography,” Proc. Int’l Conf. Theory and Application of Cryptography and Information Security EUROCRYPT, 2002.
10. A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, “Low cost Communication for Rural Internet Kiosks Using Mechanical Backhaul,” Proc. ACM Mobicom, 2006.

