EFFECTIVE SHIELDING OF STRESSED NETWORK FROM FLOOD ATTACKS

Gowtham Kovvuri[1] and Venkata Kiran Kuna[2]

1. M.Tech Scholar Department of CSE, Kaushik College of Engineering, Visakhapatnam, AP, India.
2. Assistant Professor Department of CSE, Kaushik College of Engineering, Visakhapatnam, AP, India

Abstract: Disruption Tolerant Networks (DTNs) bring into effective action of the mobility of nodes and the conciliatory contacts among nodes for relevant data communications. Due to the limitation in network resources such as contact opportunity and buffer space 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. Rate limiting was proposed 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. Here detection adopted Comply - 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. Using Rate limit certificate only the flood attacker who exceeds the rate limit was identified. To overcome this proposed approach uses key. Key will be generated for the node who wishes to send packets less than the rate limit. Based on AES algorithm the key Generates, based on keys & attackers it is easy to identify who sends packet within the rate limit.

Keywords: *DTN, security, flood attack, detection*

I. Introduction

Disruption Tolerant Networks (DTNs) Sensor networks hold the promise of facilitating large-scale, real-time data processing in complex environments, helping to protect and monitor military, environmental, safety-critical, or domestic infrastructures and resources, Denial-of-service attacks against such networks, however, may permit real world damage to public health and safety. DTN consist of mobile nodes carried by human beings, vehicles, 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 (which is called a contact between them).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. 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 the limitation in bandwidth and buffer space, DTNs is 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 attacker’s 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. Even with veniality and integrity, the WSN is not achieving its objectives if the services provided by it are not available to authorized users when they need it. In networks with such scarce resources, their improper con-assumption or destruction is a big concern. In addition to being a security problem, an inability of the network to perform its task may be a safety hazard, depending on the system being monitored or controlled. Although many schemes have been proposed to defend against flood attacks on the Intern et and in wireless sensor networks, they assume persistent connectivity and cannot be directly applied to DTNs that have intermittent connectivity. In DTN Rate limiting was employed to defend against flood attacks in DTNs. In this 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 any 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.

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. We provide a lower and upper bound of detection probability and investigate the problem of parameter selection to maximize detection probability under acertain amount of exchanged claims. The effectiveness and efficiency of our scheme are evaluated with extensive trace-driven simulations, These examples demonstrate that consideration of security at design time is the best way to ensure successful network deployment.

Motivation: A Domain Name Server (DNS) Amplification attack is a popular form of Distributed Denial of Service (DDoS), in which attackers use publically accessible open DNS servers to flood a target system with DNS response traffic. The primary technique consists of an attacker sending a DNS name lookup request to an open DNS server with the source address spoofed to be the target’s address. When the DNS server sends the DNS record response, it is sent instead to the target. Attackers will typically submit a request for as much zone information as possible to maximize the amplification effect. In most attacks of this type observed by US-CERT, the spoofed queries sent by the attacker are of the type, “ANY,” which returns all known information about a DNS zone in a single request. Because the size of the response is considerably larger than the request, the attacker is able to increase the amount of traffic directed at the victim. By leveraging a bonnet to produce a large number of spoofed DNS queries, an attacker can create an immense amount of traffic with little effort. Additionally, because the responses are legitimate data coming from valid servers, it is extremely difficult to prevent these types of attacks. While the attacks are difficult to stop, network operators can apply several possible mitigation strategies. While the most common form of this attack that US-CERT has observed involves DNS servers configured to allow unrestricted recursive resolution for any client on the Internet, attacks can also involve authoritative name servers that do not provide recursive resolution. The attack method is similar to open recursive resolvers, but is more difficult to mitigate since even a server configured with best practices can still be used in an attack. In the case of authoritative servers, mitigation should focus on using Response Rate Limiting to restrict the amount of traffic.

In a typical recursive DNS query, a client sends a query request to a local DNS server requesting the resolution of a name or the reverse resolution of an IP address.

Microsoft DNS Server

In the Microsoft DNS console tool [9]:

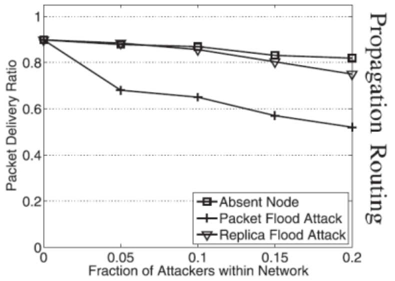
1. Right-click the DNS server and click Properties.
2. Click the Advanced tab.
3. In Server options, select the “Disable recursion” check box, and then click OK.

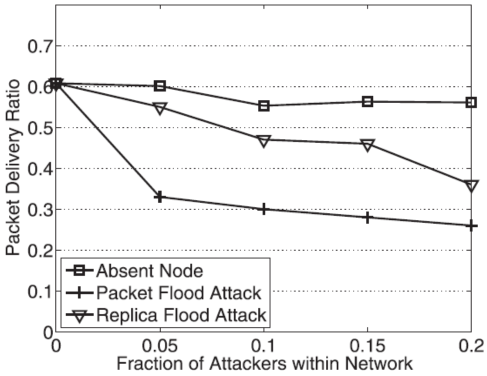
The Effect of Flood Attacks:

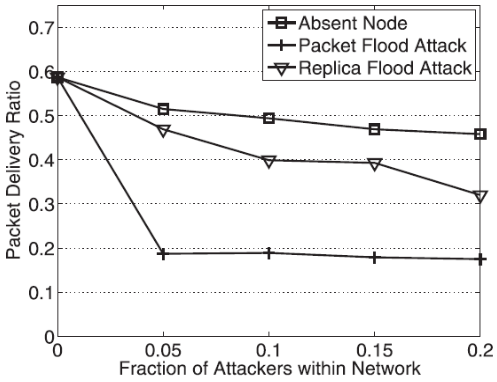
However, the choice to restrict a DTN to only authorized participants incurs an opportunity cost in the form of lost nodes that would have volunteered to participate had a simpler scheme been used. The question of whether to refuse all volunteer nodes depends on the level of threat posed by attackers and what percentage of the volunteers are honest. To demonstrate this phenomenon, we simulate the effects of adding 12 more nodes to a DTN of 18 existing authorized(and honest) nodes. The straight line in Figure 1 shows the performance of the network when only the 18 authorized nodes are available; if we increase the size of the network with 12 unauthorized but honest nodes, the average number of packets delivered per node improves by a factor of 7. As a larger proportion of the volunteers attack the network, performance degrades; however, because attacking DTNs is difficult for attackers, the network benefits from unrestricted use. We believe that for many non-military scenarios, it is un-likely that a network will attract such a large percentage of attacking nodes. The most widely deployed peer-to-peer scenarios do not see such denial-of-service statistics, including BitTorrent, SETI@home, and Tor [10]. Therefore, in this paper we suggest that successful DTNs will encourage participation and lack authentication restrictions. There are several other reasons to avoid authentication schemes for DTNs. Such mechanisms imply administrative registration and key distribution ahead of deployment however, DTNs can span hundreds of miles and many administrative domains, having a common or cooperative administrative authority for all users is unwieldy. Distributed

12 volunteers are added to a DTN of 18 authorized nodes; the straight line represents the performance of the network when only 18 honest nodes are available. The details of this simulation are included later in the paper and correspond to the greedy case in Figure 11, where attackers use knowledge of future events to plan their attacks. reputation schemes have been formally proven to be unworkable as well [8], and they are particularly problematic in a DTN where mobility leads to fleeting relationships with little chance for reputation building. For DTNs that do share an administrative authority, routing delays prevent querying of a public key infrastructure (PKI) supported by a central authority or distributed servers. Finally, all of these problems contribute to

the difficulty of managing key revocation in atimely manner. In this paper, we evaluate the success of attacks on DTN routing, finding that such networks are difficult to attack even when unauthorized, malicious nodes are allowed to participate. In particular, the routing protocols have been designed with an expectation that nodes are often unavailable attacks are similar to network failures and the DTN implicitly routes around them. Moreover, the disconnected nature of the networks limits the effectiveness of attackers attempting flooding or dropping. The combination of these factors renders DTNs much less fragile than MANETs. This is not to say that a DTN's absolute performance is better than a MANET'srather that a DTN that is without access restrictions for unauthorized nodes degrades more gracefully under attack. One of the major themes in this paper is the two-fold benefit of epidemic-style packet dissemination in DTN routing: improved packet delivery rates and greater attack tolerance. We refer to any protocol that allows multiple copies of a given packet as replicative. In contrast, protocols that allow at most a single copy of each packet in the network at a time are called forwarding . Burgess et al. [5] showed that using the MaxProp protocol, replicative routing can perform well in terms of delivery rates. We show that MaxProp can also offer significant attack tolerance. Moreover, replicative routing is shown to be crucial to achieving this tolerance. Contributions. We describe numerous attacks that are possible against DTN routing protocols, including dropping packets, flooding nodes with useless data, falsifying routing tables, and counterfeiting message acknowledgments. We quantitatively demonstrate the impact of attacks and countermeasures using traces of movement and transfers from a deployed vehicle-based DTN named UMass Diesel Net [5] and using traces recorded by the Haggle project of a Bluetooth-based pedestrian DTN. Simulations run on these traces show evidence that replicative protocols like MaxProp [5] are more robust to attack than forwarding protocols. We evaluate two types of attackers, weak and strong , that represent endpoints of a spectrum of possible adversaries. A weak attacker lacks global knowledge of DTN topology and transfer opportunities and is forced to choose participants at random to attack. Such a strategy is not efficient at attacking DTNs: a network where 10% of participants are attackers still achieves over 90% of its unassailed delivery rate, and it achieves over 70% of its rate when 30% are attackers. On the other hand, we provide the strong attacker with knowledge of future events. Even with such knowledge, we prove that identifying the most damaging attack on a DTN is an NP-hard problem given a broad class of metrics. This result limits both a potential attacker and our own analysis. Accordingly, we adopt an attack heuristic that seeks to most lower the number of temporally connected pairs of nodes ina DTN. The strong attacker has more success: the network achieves 70% of its delivery rate when 10% of the network are attackers and only 50% of its delivery rate when 30% are attackers. While our simulation results are limited to the protocols that we evaluated, we believe many of our conclusions hold in general for the numerous DTN routing protocols that have been proposed. Moreover, our proofs of complexity and description of possible attacks are also widely applicable.







Problem Definition:

Existing System 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.

Disadvantages: It is easy for an adversary to compromise nodes within the network and launch insider attacks using the compromised nodes. They cannot address insider attacks launched by compromised nodes.Insider attacks can cause significant problems in networks.

- 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 theseclaims are inconsistent.

System Design: A. Proposed System In Our Proposed System to employ the 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 (the number of nodes that it can forward each packet to). 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

Advantages

Our basic idea of detection is claim-carry-and-check.

1. Each node itself counts the number of packets or replicas that it has sent out, and claims the count to other nodes.

2. The receiving nodes carry the claims around when they move, exchange some claims when they contact, and cross-check if these claims are inconsistent.

IV. Modules Description

Our Proposed work has the following modules. There is Listed Below

1. DTN Network Creation

2. Rate Limit Certification Creation.

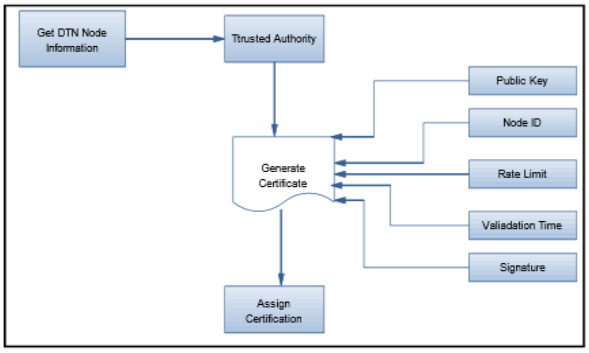
3. Claim Construction.
4. Inconsistency Analysis.
5. Metadata Exchanging Process.
6. Verification Process

DTN Network Creation

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, we assume that each packet has a lifetime. The packet becomes meaningless after its lifetime ends and will be discarded.

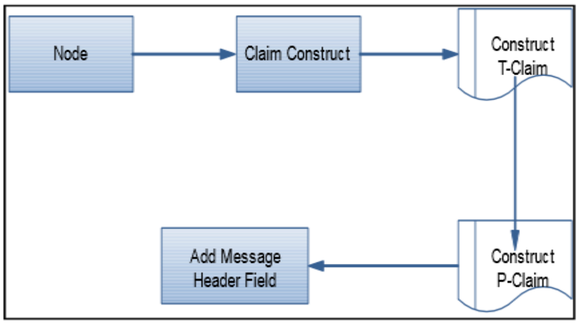
B. Rate limits Certification Reaction

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’ssignature. The rate limit certificate can be merged into the public key certificate or stand alone.



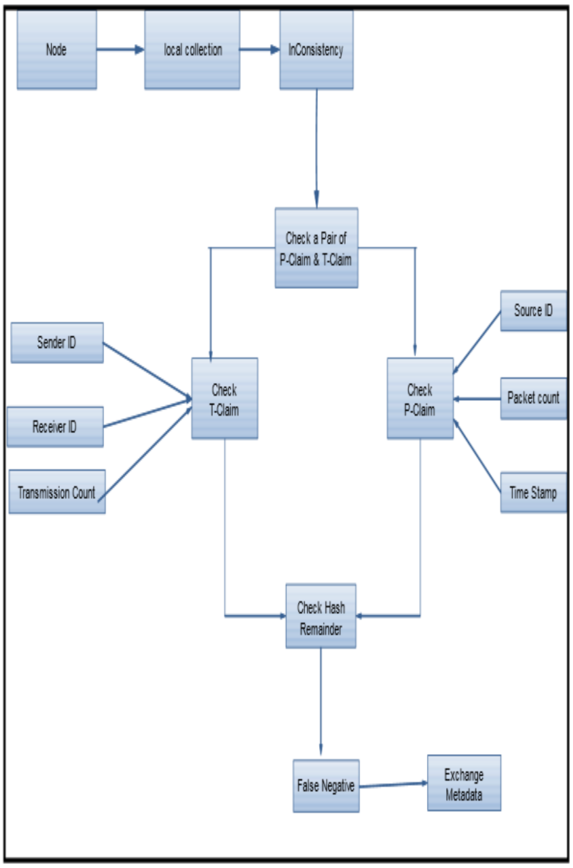
Claim Construction

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.



Inconsistency Analysis

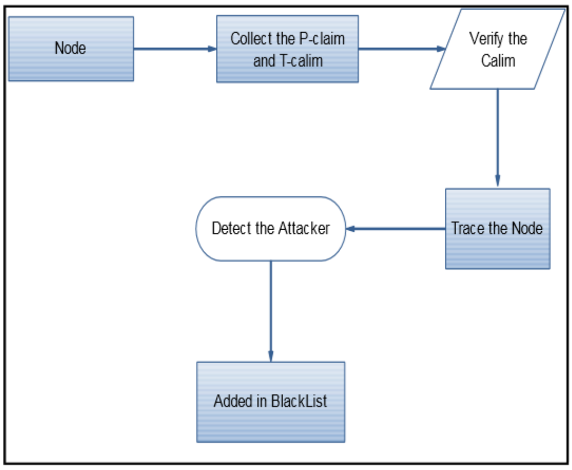
The inconsistency check based on compact P-claims does not cause false positive, since a good node never reuses any count value in different packets generated in the same interval. The inconsistency check may cause false negative if the two inconsistent P-claims have the same hash remainder



Exchanging Process

Good metadata management is essential for the efficient operation of statistical business processes. Metadata are present in every phase. The key challenge is to ensure that these metadata are captured as early as possible, and stored and transferred from phase to phase with their associated data. Metadata management strategy and systems are vital. When two nodes contact they exchange their collected P-claims and T-claims to detect flood attacks. 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.

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 an attacker, it adds the attacker into a blacklist and will not accept packets originated from or forwarded by the attacker

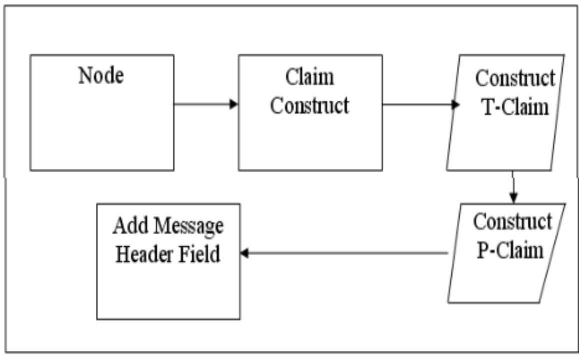
Rate Limit 'L' Existing Source

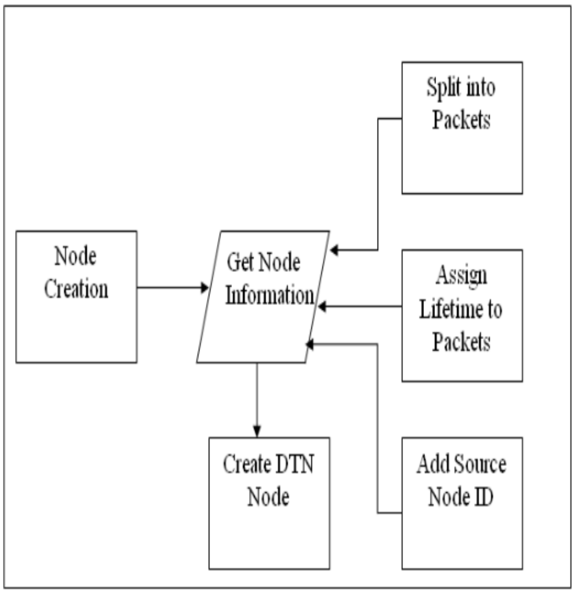
One possible method is to set L in a request-approve style. When a user joins the network, she requests for a rate limit from a trusted authority which acts as the network operator. In the request, this user specifies an appropriate value of L based on prediction of her traffic demand. If the trusted 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 unreasonably large rate limits, a user pays an appropriate amount of money or virtual currency (e.g., the credits that she earns by forwarding data for other users [25]) for her rate limit. When a user predicts an increase (decrease) of her demand, she can request for a higher (lower) rate limit. The request and approval of rate limit may be done offline. The flexibility of rate limit leaves legitimate users' usage of the network unhindered. This process can be similar to signing a contract between a smart phone 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.

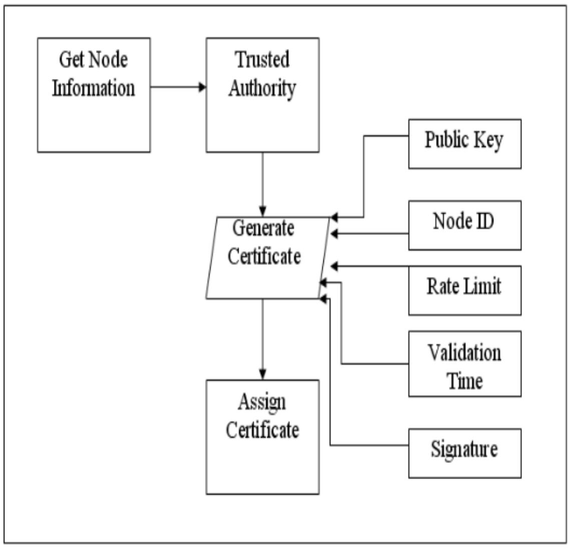
Modules Pattern Flow Chart

Our Proposed work has the following modules.

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.







Rate Limit Certificate Creation

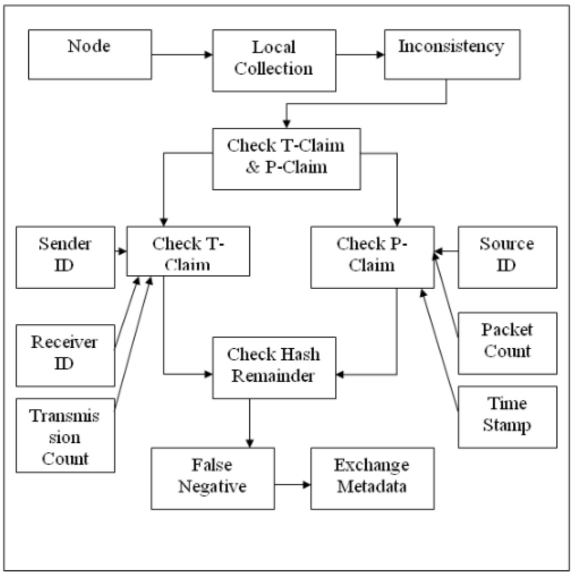
When a user joins the network, she requests for a rate limit from a trusted authority which acts as the network operator. In the request, this user specifies an appropriate value of L based on prediction of her traffic demand. If the trusted authority approves this request, it issues a rate limit certificate 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 A node stores the P-claims and T-claims collected from received data packets for a certain time denoted by and deletes them afterward. It deletes the claims redirected from other nodes immediately after it has exchanged them to K different nodes. 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.

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.

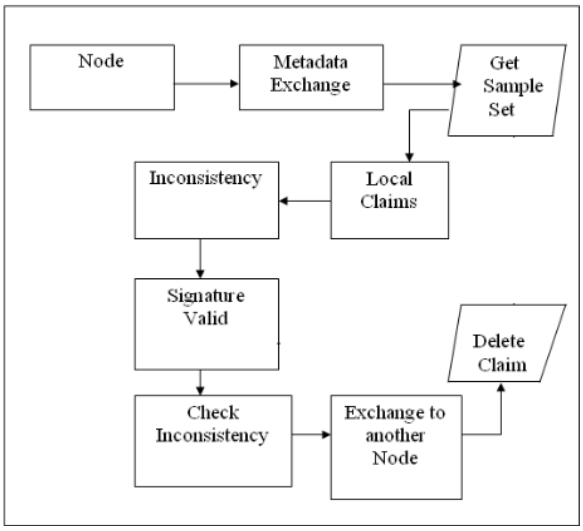
Inconsistency Analysis

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.



Metadata Exchange Process

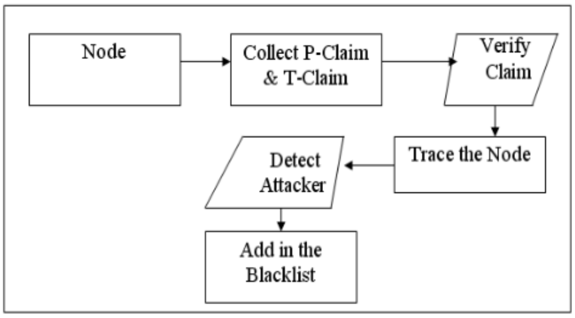
When two nodes contact they exchange their collected P-claims and T-claims to detect flood attacks. If all claims are exchanged, the communication cost will be too high. Thus, our scheme uses sampling techniques to keep the communication cost low. To increase the probability of attack detection, one 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 analyzing detection probability, we assume that each attacker acts alone.



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.

Cost Analysis



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.

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.

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.

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. 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.

2. 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.

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

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

8. A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Technical Report CS-200006, Duke Univ., 2000.

9. J. Burgess, B. Gallagher, D. Jensen, and B. Levine,“Maxprop: Routing for Vehicle-Based Disruption-Tolerant Networks,” Proc. IEEE INFOCOM, 2006.

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

11. E. Daly and M. Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs,” Proc. MobiHoc, pp. 32-40, 2007.

12. M. Motani, V. Srinivasan, and P. Nuggehalli, “PeopleNet: Engineering a Wireless Virtual Social Network,” Proc. MobiCom,pp. 243-257, 2005.