Resource-Misuse Attack Detection in Delay-Tolerant Networks

Vivek Natarajan
Department of Computer Science and Engineering
Pennsylvania State University
Email: vnataraj@cse.psu.edu

Yi Yang
Department of Electrical Engineering and Computer Science
Catholic University of America
Email: yangy@cua.edu

Sencun Zhu
Department of Computer Science and Engineering
Pennsylvania State University
Email: szhu@cse.psu.edu

Abstract—
In a Delay-Tolerant Network (DTN), data originating from a source node may be delivered to the destination node, despite the non-existence of end-to-end connectivity between them at all times. In an adversarial environment such as a battlefield, DTN nodes could be compromised to launch Denial-of-Service (DoS) attacks by generating excess data, to cause an overflow of the limited resources of the legitimate nodes, hence decreasing the network throughput. A node may also display selfish behavior by generating more data than allowed, to increase its throughput and to decrease the latency of its data packets. In this paper, we term such a DoS attack and selfish data generation behavior, a resource-misuse attack. We study two types of resource-misuse attacks, breadth attacks and depth attacks. Accordingly, we propose different schemes to detect these attacks. Trace-driven simulations using both a synthetic and a real-world trace show that our detection schemes have low average detection latency and additionally, probabilistic detection of the depth attack has low false positive and false negative rates.

Index Terms—Delay-Tolerant Network ; Denial-of-Service Attack ; Probabilistic Detection

I. INTRODUCTION

Delay-Tolerant Networks (DTNs) [1], [2], [3] are intermittently connected mobile networks that facilitate data communication under the condition of sparsely distributed nodes and no infrastructure. Thus, DTNs are essential in many military, space, rural and other such scenarios with sporadic connectivity. UMass DieselNet [2] is an example of a typical DTN. It consists of 30 buses fitted with wireless transmitters and receivers, that communicate via the 802.11 protocol. A connection event occurs and data is exchanged between two buses, when they are within the wireless range of each other.

DTNs are characterized by their lack of instantaneous end-to-end paths. In a DTN, generated data at the source is routed towards the destination opportunistically. The frequency of direct contact opportunities between nodes in a DTN is low, due to the sparse nature of the network. Hence, every contact opportunity is a critical resource to transfer data.

To increase the delivery probability of a data packet, a source node could generate multiple copies of it and transfer them to other nodes during contact opportunities, in the hope that at least one of those nodes or itself comes in contact with the destination node and delivers the packet. Several such replicative routing algorithms [4], [5] set an upper limit on the number of copies of a packet in the DTN, to ensure fair usage of the limited network resources.

Indeed, under the cover of replicative routing, an attack that causes a misuse of the scarce DTN resources, is inconspicuous but destructive. We term such an attack a resource-misuse attack in this paper. A node could either disrupt the normal functionality of a DTN by flooding with a large quantity of useless data (a denial-of-service attack launched by a malicious node) or generate more data than allowed by the network, to increase its throughput and to decrease the latency of its packets (disallowed data generation by a selfish node [6]). Resource-misuse attacks lead to the use of rare contact opportunities between nodes to transfer disallowed data, thereby misusing the limited network bandwidth of the legitimate nodes and also causing eviction of regular data packets from the buffers of the legitimate nodes. The network throughput of the DTN decreases considerably due to the effect of resource-misuse attacks. For ease of presentation, we call both malicious and selfish nodes compromised nodes in this paper.

A node launching a resource-misuse attack could either inject several different data packets, or inject several copies of the same data packet in the network. We call the former a breadth attack since the node generates different packets, and the latter a depth attack since the node generates multiple copies of the same packet. A node may launch a combination of these two attacks or even collude with other compromised nodes, to make the attack more effective.

Defense against resource-misuse attacks in a DTN is still an open problem. Although several schemes have been proposed to defend against denial-of-service attacks in the Internet [7] and in wireless sensor networks [8], they assume persistent connectivity and cannot be directly applied to a DTN that only has intermittent connectivity. Most of the existing work in DTNs concentrate on routing and data dissemination (e.g., [2], [4], [5]) without consideration of security issues. Rate-limiting techniques for other networks [9], [10] are also not applicable to address resource-misuse attacks in a DTN, since they require extra devices and configuration, which may not be available in a DTN. Even if a rate-limiting technique is

*This work was supported in part by ARL CTA and NSF CAREER 0643906.
applicable in a DTN, the schemes to detect resource-misuse attacks that we propose in this paper are useful to estimate the threshold rate that the rate-limiting technique should use.

In this paper, we propose a scheme to detect resource-misuse attacks, in which the gateway of the DTN monitors the activities of nodes and detects an attack if there is deviation from expected behavior. However, the detection latency of this scheme could be higher than that tolerated by the network, since nodes are sparsely distributed in a DTN and come in contact with the gateway infrequently. The case for the depth attack is even worse, since the gateway needs to meet a sufficient number of nodes to make a decision. To reduce the detection latency, we augment the detection scheme for the depth attack with probabilistic detection. Trace-driven simulations using both a synthetic and a real-world trace show that our detection schemes have low average detection latency and additionally, probabilistic detection of the depth attack has low false positive and false negative rates.

II. SYSTEM MODEL

A. Network Model

The Delay-Tolerant Network (DTN) consists of a certain number $N$ of resource (e.g., bandwidth and storage) limited nodes spread out sparsely in the network area. The nodes are mobile and come in contact with each other infrequently. We assume loose time synchronization among the nodes, at an acceptable granularity level. In addition to the nodes, there exists a powerful gateway, with unlimited resources, at a particular location in the DTN.

1) Gateway: Nodes come in the vicinity of the gateway when they need to upload data to or download data from the outside world, such as the Internet. For monitoring or surveillance purpose, when a node comes in contact with the gateway, it securely reports a snapshot of its buffer to the gateway.

2) Routing: Data is generated by a source node for a destination node in the form of a packet. A packet consists of the header and the payload. The header contains the packet identifier, the source node identifier, the destination node identifier and the packet generation time.

Multiple copies of a packet exist in the network if a replicative routing algorithm is used, e.g., Source Spray and Wait [4]. Note that a source node benefits from generating multiple copies of a packet and transferring them to nodes, since the delivery probability of the packet to the destination node increases and if the packet is delivered, its delivery latency may decrease. In Source Spray and Wait, a source node generates $L$ initial copies of a packet. When a node $A$ comes in contact with a node $B$, for every packet that $A$ has at least one copy of, it checks if $B$ has any copies of the packet. If it does not, it transfers one copy of the packet to $B$. If $A$ has exactly one copy of the packet, it transfers the copy to $B$ only if $B$ is the destination node of the packet. Note that if $A$ transfers a copy, the number of copies of the packet in $A$’s buffer decreases by one after the transfer. There are at most two hops (or at most one intermediate node) from the source node to the destination node in this routing algorithm.

The upper bound $L$ is set for the number of copies of a packet since the resources of a DTN are limited. Assuming that time is divided into a set of intervals, an upper bound $M$ could also be set by the DTN as the maximum number of packets generated by a node in a time interval. Rate-limiting in this manner is required since otherwise, several nodes may generate a large number of packets in the same time interval, causing congestion in the DTN.

Our detection schemes are designed to work with Source Spray and Wait in this paper. However, they are not restricted to a specific routing algorithm, and could be suitably modified to work with other routing algorithms as well.

3) Mobility: In this paper, we use both a synthetic (random walk model [11], [12]) and a real-world (iMote Infocom 2005 trace [3], [13], [14]) mobility model. In the random walk model, $N$ nodes with a transmission range of $K$ perform independent random walks on a $\sqrt{\alpha} \times \sqrt{\alpha}$ two-dimensional torus, where $\alpha$ is the area of the torus. Each node starts at a random location on the surface of the torus. At every clock tick, each node moves one location (to the left, to the right, up or down) at random. When two nodes come in the vicinity of each other (determined by the transmission range $K$, e.g., if $K=1$, they are in the vicinity of each other if they are at most one location away), they exchange packets based on the routing algorithm. The iMote Infocom 2005 trace was collected as a result of a human mobility experiment at the Infocom 2005 conference site.

B. Security Model

We assume the existence of a public-key security mechanism in the DTN. Key establishment and management in the DTN is beyond the scope of this paper. The source node of a packet attaches a digital signature when it transfers a copy of the packet to another node, for node authentication and data integrity. Similarly, a node that reports the packets in its buffer to the gateway attaches a digital signature. A source node may encrypt the payload of a packet it generates with the public key of the destination node, to ensure confidentiality during routing.

However, an adversary could compromise a (small) fraction of nodes and obtain their keys, if they lack hardware support for tamper resistance. A compromised node could launch a resource-misuse attack independently, or could collude with other compromised nodes. We assume that the gateway is a trusted entity.

III. PROPOSED SCHEMES

A. Overview

We propose schemes to detect resource-misuse attacks that cause misuse of scarce resources of a DTN, such as frequency of contact opportunities between nodes, node network bandwidth and node buffer size. The gateway, situated at a particular location in the DTN, is responsible for the detection. To detect resource-misuse attacks, the gateway maintains data
structures to store the count values of packets generated by a node in a time interval, and of copies of a packet generated by a node, that are updated during a contact opportunity with a node. A resource-misuse attack could be detected by the gateway after it meets a certain number of nodes and collects sufficient data.

In the remainder of the paper, we assume that a compromised node does not report the packets in its buffer used in resource-misuse attacks launched by itself or by other compromised nodes it colludes with, to avoid or delay the detection of those attacks.

B. Data Structures

To detect breadth attacks, the gateway maintains a count of the number of packets \( m \) reported to it, generated by a node in a time interval, for each node and each time interval. The gateway also maintains the identifiers of all packets that constitute the value of \( m \).

To detect depth attacks, for each packet, the gateway maintains a count of the number of copies \( l \) of the packet reported to it and the identifiers of all nodes involved in routing each copy, starting with the source node of the packet and ending with the node that reported the copy of the packet (for redundancy avoidance, see Section III-E). For each packet, the gateway also maintains a count of the number of all nodes \( n \) that have come in its contact starting from the time when the first copy of the packet was reported to it and the identifiers of all nodes that constitute the value of \( n \) (for probabilistic detection, see Section III-D).

C. Deterministic Detection

A breadth attack is detected when the number of packets \( m \) reported to the gateway, generated by a node in a time interval, exceeds a system threshold \( M \). Note that a compromised node may transfer (a single copy of) different packets it generated in the same time interval, during a single contact opportunity with another node. If the gateway meets a certain number of such nodes that receive different packets from a compromised node, it may observe more than \( M \) packets generated by the compromised node in a time interval, and deterministically detect a breadth attack. We assume that a source node launching breadth attacks does not transfer copies of more than \( M \) packets it generated in the same time interval during a single contact opportunity, to avoid easy detection by the receiving node.

A depth attack is detected when the number of copies \( l \) of a packet reported to the gateway exceeds a system threshold \( L \). Since any source node generates at most \( L \) initial copies of a packet, at any time it is expected that there are at most \( L \) nodes with at least one copy of a packet. Hence, if the gateway meets \((L+1)\) nodes with a copy of a packet, it deterministically detects a depth attack. Note that it may take longer for a depth attack to be detected than a breadth attack. This is so since the gateway needs to meet \((L+1)\) nodes to detect a depth attack but may need to meet only a few nodes to detect a breadth attack.

We next augment the detection scheme for a depth attack with probabilistic detection, to reduce its detection latency.

D. Probabilistic Detection of a Depth Attack

At any time, the gateway could estimate if the total number of copies of a packet in the DTN exceeds the system threshold \( L \) based on the number of copies \( l \) of the packet reported to it and the number of all nodes \( n \) that have come in its contact starting from the time when the first copy of the packet was reported to it. Note that the value of \( n \) includes all nodes, irrespective of whether they reported the packet or not.

Based on Source Spray and Wait routing, after the source node of a packet has spread its copies, there are \( L \) nodes with a copy and \((N-L)\) nodes without a copy of the packet in the network. In the random walk model, the expected time taken for a node to come in contact with the gateway is independent of its location in the two-dimensional torus \([11], [12]\). Hence, if the nodes follow the random walk model, the copies of the packet are expected to be uniformly randomly distributed among the nodes that the gateway meets and the probability \( p(l) \) of \( l \) copies of a packet reported to the gateway after it meets \( n \) nodes is calculated as follows:

\[
p(l) = \frac{(l) \cdot (N-L)^{n-l}}{N^n}
\]

(1)

The distribution of \( l \) is the hypergeometric distribution. Intuitively, the values of \( l \) and \( n \) at the gateway are expected to be relatively closer to each other for a packet that is used in a depth attack than for a packet that is not. This is due to the fact that the number of nodes that obtain copies of a packet used in a depth attack may be higher. Whenever the value of \( l \) for a packet changes, it is possible for the gateway to estimate if its values of \( l \) and \( n \) for the packet are close to each other by determining if \( p(l) < p \), for its values of \( l \) and \( n \) for the packet, where \( p \) is a threshold probability. If that is the case, the gateway probabilistically detects the packet to be used in a depth attack.\(^1\)

If \( p = 0 \), attacks are only deterministically detected. As the value of \( p \) increases, attacks are either probabilistically or deterministically detected and hence, the detection latency and the false negative rates decrease. However, the false positive

\(^1\)In particular, the plot of \( p(l) \) for the value of \( l \) for a packet at the gateway versus \( n \) is symmetric, as shown in Figure 1, and two sets of values of \( n \) (\( I_1 \) and \( I_2 \)) satisfy the inequality \( p(l) < p \). The gateway estimates if its values of \( l \) and \( n \) for the packet are close to each other by determining if its value of \( n \) for the packet belongs to \( I_1 \).
rate increases. The threshold probability \( p \) is tuned such that the detection latency for depth attacks is as low as possible, while the false positive and false negative rates are tolerable.

Algorithm 1 Detection of Depth Attacks

```plaintext
procedure DepthAttackDetection(A)
1: for each packet in the memory of the gateway do
2:   if \( A \notin \text{the set of nodes that constitute the value of } n \) then
3:      \( n \leftarrow n+1 \);
4:      add \( A \) to the set of nodes that constitute the value of \( n \);
5:   end if
6: end for
7: for each packet in the buffer of \( A \) do
8:   if \( l = 0 \) then
9:      \( n \leftarrow 1 \);
10:     add \( A \) to the set of nodes that constitute the value of \( n \);
11:    end if
12:   if \( A \) has not reported the packet to the gateway then
13:      if \( A \) is the destination node of the packet and the node that
14:         transferred a copy of the packet to \( A \) has not reported the packet to
15:         the gateway then
16:         \( flag \leftarrow 1 \);
17:      end if
18:   end if
19:   if \( flag = 1 \) then
20:      store that \( A \) has reported the packet to the gateway;
21:      \( l \leftarrow l+1 \);
22:      store the set of nodes involved in routing the copy of the packet;
23:      if \( l > L \) then
24:         detect that the packet was used in a depth attack; \{Deterministic
25:         Detection\}
26:      else if \( p(l) < pr \) then
27:         detect that the packet was used in a depth attack; \{Probabilistic
28:         Detection\}
29:      end if
30:   end if
31: end for
32: return;
```

E. Avoiding Redundancy in Detection of a Depth Attack

Redundant counting of copies of a packet in the detection scheme for the depth attack could occur in the following ways (these cases are illustrated in Figure 2):

1) Consider the following sequence of events:
   - The gateway meets the source node of a packet, with multiple copies, and increments \( l \) for the packet by the number of copies.
   - The source node comes in contact with a node \( A \) and transfers a copy.
   - The gateway meets \( A \) and increments \( l \) by one.

   Note that the gateway has considered the same copy twice in its count of \( l \). To avoid this redundancy, the gateway conservatively increments \( l \) by just one, even when it meets a node with multiple copies. If the gateway meets the source node of a packet with multiple copies, it increments \( l \) for the packet by one and waits for the source node to distribute those copies to other nodes, and then considers those copies in the count of \( l \) if it subsequently meets those nodes. Since we assume that a compromised source node launching a depth attack does not report the packet used in the attack to the gateway, conservatively incrementing \( l \) by one when the gateway meets a node with multiple copies of a packet does not increase the detection latency or decrease the chance of detection of depth attacks.

2) Consider the following sequence of events:
   - The gateway meets a node \( A \) with exactly one copy of a packet and increments \( l \) for the packet by one.
   - \( A \) comes in contact with the destination node of the packet and transfers its copy, following Source Spray and Wait routing.
   - The gateway meets the destination node and increments \( l \) by one.

   Note that once again, the gateway has considered the same copy twice in its count of \( l \). To avoid this redundancy, if the destination node of a packet reports it, the gateway increments \( l \) for the packet by one only if the node that transferred the copy of the packet to the destination node had not reported the packet earlier.

3) Consider the following sequence of events:
   - A node \( A \) with exactly one copy of a packet comes in contact with the destination node of the packet and transfers its copy, following Source Spray and Wait routing.
   - The gateway meets the destination node and increments \( l \) for the packet by one.
   - \( A \) comes in contact with the gateway and reports the packet, even though it should no longer have a copy in its buffer. The gateway increments \( l \) by one.

   Note that \( A \) is not supposed to retain its copy in its buffer after the transfer to the destination node. To avoid this
redundancy, when a node apart from the destination node of a packet reports the packet, the gateway checks if the destination node had reported the packet earlier. If not, it increments \( l \) for the packet by one. If the destination node had reported the packet earlier, the gateway checks if the node that transferred the copy of the packet to the destination node was \( A \). If yes, the gateway does not increment \( l \). If not, the gateway increments \( l \) by one.

The complete procedure to perform probabilistic and deterministic detection of depth attacks, including redundancy avoidance, when a node \( A \) comes in contact with the gateway, is shown in Algorithm 1.

**F. Security Analysis**

We have performed detailed security analysis of our detection schemes that is available here [15]. We now briefly present some of the results of our security analysis.

A source node is required to attach a digital signature in a copy of a packet it transfers, that also contains the identifier of the receiving node. A node launching a resource-misuse attack is unable to change the source node identifier in a packet used in the attack to that of some node other than itself, to escape identification by the gateway, if the attack is detected. This is so since the copies of the packet would be rejected by the receiving nodes when they verify the digital signature. Also, a set of colluding nodes are unable to create a false positive for detection of a resource-misuse attack, by each node in the set reporting a bogus or an eavesdropped packet and falsely claiming that it received a copy of the packet from (and generated by) some legitimate node, since the digital signature is verified by the gateway. Furthermore, a node in the set of colluding nodes is unable to report a packet multiple times to the gateway by using the identity of a different node in the set each time, to create a false positive, for the same reason.

A source node could generate more than \( M \) packets in a time interval and alter the generation times of the extra packets (other than \( M \) packets) to some times in some previous time intervals in which it had generated less than \( M \) packets, so that no more than \( M \) packets have their generation times set to times in a particular time interval. This results in the source node avoiding detection of any breadth attacks by the gateway. However, a copy of a packet whose generation time is set to a time in a previous time interval is likely to be dropped relatively soon from the buffer of the receiving node, based on the replacement algorithm, that could favor retention of recent copies over old ones for most of the routing algorithms. Furthermore, if at some time \( t \), \( \sigma \) time intervals have elapsed, the source node is unable to have generated more than a total of \( M \times \sigma \) packets in the \( \sigma \) time intervals, and also avoid detection of breadth attacks.

**IV. Analytical Results on Detection Time**

In this section, we calculate an upper bound on the expected value of the detection time for deterministic detection of a depth attack in the random walk model. Let \( N \) be the total number of nodes and \( A \) be a compromised node launching a depth attack by generating \((L+\delta)\) copies of a packet, instead of the \( L \) allowed copies. The detection time \( T \) for the attack is the difference between the time when it is detected and the time when it happened. The attack happens when \( A \) transfers the \((L+1)\)th copy to the \((L+1)\)th node without a copy.

In the random walk model, the expected time \( E \) for two nodes on the surface of a two-dimensional torus with area \( \alpha \) and transmission range \( K \) to meet each other is calculated as follows [11], [12]:

\[
E = \frac{2}{\pi} \left[ 0.344 \alpha \log_2 \left( \frac{2K+1}{K-\delta} \right) \right].
\]

Also, the expected time for a node to hit a particular location on the torus is twice the value of \( E \). Therefore, each meeting time or hitting time is an iid exponential random variable with average \( E \) and \( 2E \), respectively.

1) Non-Colluding Depth Attack: The upper bound on the expected value of \( T \) is the sum of the expected time \( E[T_1] \) for \( A \) to transfer the \((L+\delta)\)th copy after it transfers the \((L+1)\)th copy to a node without a copy that come in its contact and the expected time \( E[T_2] \) for \( (L+1) \) one more than the threshold) out of the \((L+\delta)\) nodes with a copy to come in contact with the gateway (i.e., \( E[T] \leq E[T_1]+E[T_2] \)). This is due to our assumption that at the time when the \((L+\delta)\)th copy is transferred and there are \((L+\delta)\) nodes, each with a copy, no node with a copy has come in contact with the gateway (i.e., there is no overlap between \( T_1 \) and \( T_2 \)).

After \( A \) transfers the \((i-1)\)th copy, there are \( i \) nodes (including \( A \)) with at least one copy and \((N-i)\) without a copy. The time \( T_i \) for \( A \) to transfer the \( i \)th copy after transferring the \((i-1)\)th copy, is the time for any of the \((N-i)\) nodes without a copy to come in its contact. According to [11], \( E[T_i] = \frac{E}{N-i} \). Hence, the expected time for \( A \) to transfer the \((L+\delta)\)th copy after transferring the \((L+1)\)th copy is \( E[T_1] = \sum_{i=L+2}^{L+\delta} E[T_i] = \sum_{i=L+2}^{L+\delta} \frac{E}{N-i} \).

Similarly, the time \( T_2 \) for the gateway to meet the \( i \)th node with a copy after meeting the \((i-1)\)th node with a copy is the time for any of the \( ((L+\delta)-(i-1)) \) nodes with a copy to come in its contact. According to [11], \( E[T_i] = \frac{E}{(L+\delta) -(i-1)} \). Thus, the expected time for \((L+1)\) out of the \((L+\delta)\) nodes with a copy to come in contact with the gateway is \( E[T_2] = \sum_{i=L+1}^{L+\delta-1} \frac{2E}{N-i} \).

Hence, the upper bound on \( E[T] \) is calculated as follows:

\[
E[T] \leq \sum_{i=L+2}^{L+\delta} \frac{E}{N-i} + \sum_{i=1}^{L+1} \frac{2E}{(L+\delta) -(i-1)} \quad (2)
\]

2) Colluding Depth Attack: Similar to the non-colluding depth attack case, to derive an upper bound on \( E[T] \), we assume that at the time when the \((L+\delta)\)th copy is transferred, no node with a copy has come in contact with the gateway. Additionally, we assume that the \((L+1)\)th copy is transferred to the \((L+1)\)th legitimate node to receive a copy. Note that if less than \((L+1)\) legitimate nodes receive copies of the packet, it is not possible for the attack to be deterministically detected. The upper bound on \( E[T] \) is calculated as follows:

\[
E[T] \leq \sum_{i=L+2}^{L+\delta} \frac{E}{N-i} + \sum_{i=1}^{L+1} \frac{2E}{(L+1) -(i-1)} \quad (3)
\]
V. PERFORMANCE EVALUATION

A. Experiment Setup

We use both a synthetic and a real-world trace for our evaluation. The synthetic trace we use is the random walk trace based on the random walk model. The gateway is at a fixed location in this trace. The real-world trace we use is the iMote Infocom 2005 trace. Since this real-world trace does not include location information of the nodes, the times when nodes come in contact with the gateway is not available. Therefore, we simulate a random walk with the same number of nodes as in the real-world trace, obtain the times when nodes come in contact with gateway, and use those in the real-world trace.

We implement the detection schemes proposed in Section III. Similar to [13], [14], we assume that the node packet generation rate is around 10 to 12 packets per hour. The routing algorithm is Source Spray and Wait. The default values of all simulation parameters is listed in Table I. Unless otherwise stated, we use these values of the parameters for the analytical and the simulation results. Each simulation is run for a total of 100 times with different seed values and the average value of the results is plotted on the graphs.

A compromised node launches a breadth attack by generating \((\gamma \times M)\) packets (\(\gamma\) packets at \(M\) equally spaced times) in each time interval and a depth attack by generating \((L+\delta)\) copies of each packet. An attack happens when the first copy of a packet, disallowed by the DTN, is transferred by the node launching the attack.

B. Experiment Metrics

We use the following metrics for the evaluation of our detection schemes:

1) **Average Detection Time** \((T)\) : the average of the detection times after the resource-misuse attacks happen.

2) **False Positive Rate** \(fp\) : the ratio of the number of packets wrongly flagged as attacks to the total number of flagged packets.

3) **False Negative Rate** \(fn\) : the ratio of the number of undetected attack packets to the total number of non-flagged packets.

C. Validation of Analytical Results

Our analytical and simulation results of the average detection time for deterministic detection of a depth attack for the random walk trace are shown in Figure 3. We observe that the analytical results are an upper bound of the simulation results, thereby verifying the correctness of our analysis.

D. Summary of Simulation Results

In figures in this section, “deterministic” refers to only deterministic detection and “probabilistic” refers to a combination of deterministic and probabilistic detection. We observe that the average detection time of our detection schemes is low. For example, in Figures 4(a) and 5(a), the average detection time is less than 1,000 seconds for a breadth attack and is less than 5,500 seconds for a depth attack, even when the percentage of compromised nodes \(C\) is 40%. These values for the average detection time are reasonable in a DTN. Also, when \(C\) is 30%, the average detection time for “probabilistic” detection is about 23% lower in a non-colluding attack scenario and is about 10% lower in a colluding attack scenario than that of “deterministic” detection, for depth attacks. We also observe that the false positive and false negative rates of our detection scheme for depth attacks are low. For example, in Figures 6 and 7, both the false positive and false negative rates are less than 6%, when \(C\) is 40%. We next evaluate the effect of different parameters on our experiment metrics.

E. Average Detection Time

Figures 4(a) and 5(a) show the effect of the percentage of compromised nodes \(C\) on the average detection time. Only the source node launching a non-colluding attack participates in the attack, irrespective of the value of \(C\). Hence, as \(C\) increases, the average detection time remains about the same for a non-colluding attack.

Figure 4(b) shows the effect of the threshold \(M\) and the multiplier for packets \(\gamma\) on the average detection time for a breadth attack. For a particular value of \(M\), as \(\gamma\) increases, the number of extra packets generated in a time interval by a node launching a breadth attack increases, and due to this, the rate at which packets get dropped from node buffers also

---

**TABLE I DEFAULT SIMULATION PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N)</td>
<td>50</td>
</tr>
<tr>
<td>(K)</td>
<td>10,000</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>2,500</td>
</tr>
<tr>
<td>(\beta)</td>
<td>100,000</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>10</td>
</tr>
<tr>
<td>(\delta)</td>
<td>5</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>30</td>
</tr>
<tr>
<td>(M)</td>
<td>3</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>2</td>
</tr>
<tr>
<td>(L)</td>
<td>10</td>
</tr>
<tr>
<td>(\delta)</td>
<td>16</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>0.001</td>
</tr>
</tbody>
</table>
attacks. For a particular value of \( L \) \( \delta \) copies of a packet \( \gamma \) value of variation of the stronger of the two opposing effects. The effect of the overall effect on the average detection time depends on the strength of the network increases. If the average number of contact opportunities per node increases, since the density of the network increases. If the average number of contact opportunities per node increases, the number of nodes that obtain copies of packets used in \( \delta \) attacks is also high, the number of nodes that obtain copies of packets used in \( \delta \) or \( \gamma \) depth attacks is also high, causing the average detection time of those attacks to be low.

\[ T \]

\[ C \]

\[ N \]

\[ \]
undetected attack packets increases at a relatively fast rate for a \textit{colluding} attack compared to that for a \textit{non-colluding} attack as \( C \) increases, causing the false negative rate to also increase at a relatively fast rate.

VI. RELATED WORK

Security attacks in a DTN have been addressed in previous work. Dropping data [13], flooding with useless data [13], corrupting routing tables [13], [14], false acknowledgments of delivery [13], [14], strong attacks launched using global knowledge of network topology and future transfer opportunities [13], black-hole attacks [16] and wormhole attacks [17] are some examples. Lee et al. [18] address flooding attacks on probabilistic DTN routing algorithms by proposing a suitable queuing policy for node buffer management.

An authentication scheme using public key cryptography, public key infrastructure (PKI) [19] or identity-based cryptography (IBC) [19] could exist in a DTN. Burgess et al. [13] argue that the complexity of an authentication scheme in a DTN may dissuade node participation in routing, and demonstrate that even DTNs that lack security are robust to some attacks. Choo et al. [14] study the robustness of a DTN without an authentication scheme against routing attacks.

Hsiao et al. [20] propose a secure probabilistic threshold-based event validation protocol for VANETs. The broad idea to probabilistically estimate if a count value exceeds a threshold is similar to that of our probabilistic detection scheme.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose both a deterministic and a probabilistic scheme to detect denial-of-service attacks and excess data generation by selfish nodes in a DTN. Simulation results show that our detection schemes have low average detection latency and low false positive and false negative rates.

In the future, we shall study the use of a distributed scheme to detect \textit{resource-misuse} attacks in a DTN. The digital signature a source node attaches to a copy of a packet could be required to also contain the packet number of the packet in the time interval it was generated in (e.g., second packet in third time interval) and the copy number of the packet (e.g., fourth copy). A node launching a \textit{resource-misuse} attack is then forced to claim the same packet/copy number for different packets/copies to ensure that the packet/copy numbers do not exceed the respective system thresholds. During a contact opportunity between two nodes, the header and packet/copy numbers of copies of packets in the node buffers could be exchanged and checked for inconsistencies, i.e., different packet identifiers with the same packet number or different copies with the same copy number. Thus, individual nodes would themselves be responsible to detect the attacks.

REFERENCES