DISA: Detection and Isolation of Sneaky Attackers in Locally-Monitored Multi-hop Wireless Networks

Issa Khalil*, Saurabh Bagchi+, Najah AbuAli*, M. Hayajneh*

* College of Information Technology, United Arab Emirates University, UAE
  {ikhalil@uaeu.ac.ae, najah@uaeu.ac.ae, mhayajneh@uaeu.ac.ae}
+ Dependable Computing System Lab (DCSL), School of Electrical & Computer Engineering
  Purdue University, USA, {sbagchi@purdue.edu}

Corresponding Author: Issa Khalil, ikhalil@uaeu.ac.ae

Abstract

Local monitoring has been demonstrated as a powerful technique for mitigating security attacks in multi-hop ad-hoc networks. In local monitoring, nodes overhear partial neighborhood communication to detect misbehavior such as packet drop or delay. However, local monitoring as presented in the literature is vulnerable to a class of attacks that we introduce here called stealthy packet dropping. Stealthy packet dropping disrupts the packet from reaching the destination by malicious behavior at an intermediate node. However, the malicious node gives the impression to its neighbors that it performed the legitimate forwarding action. Moreover, a legitimate node comes under suspicion. We introduce four ways of achieving stealthy packet dropping, none of which is currently detectable. We provide a protocol called DISA, based on local monitoring, to remedy each attack. DISA incorporates two techniques—having the neighbors maintain additional information about the routing path, and adding some checking responsibility to each neighbor.

We show through analysis and simulation that basic local monitoring (BLM) fails to efficiently mitigate any of the presented attacks while DISA successfully mitigates them.

Keywords: Packet dropping, multi-hop wireless networks, local monitoring, misrouting, transmission power control, malicious collusion.

1 Introduction

The traffic in wireless ad-hoc networks can be broadly classified into data and control traffic. Control traffic contains information to set up the network for data traffic to flow. Typical examples of control traffic include routing, monitoring the aliveness of nodes, topology discovery, and system management. Examples of data traffic include sensor readings and alert messages in surveillance environments.

It has been shown in the literature that wireless ad-hoc networks are vulnerable to a wide range of security attacks. The open nature of the wireless communication channels, the lack of
infrastructure, the fast deployment practices, and the hostile environments where they may be deployed, make them more susceptible to various kinds of attacks against both control and data traffic. Moreover, many ad-hoc networks such as sensor networks are resource-constrained, not only on energy but on bandwidth and computation as well. This limitation presents an additional challenge that any security protocol must live under. Control traffic attacks include wormhole [5], rushing [4], and Sybil [10]. The most notable data traffic attacks are blackhole, selective forwarding, and delaying of packets, in which respectively a malicious node drops data (entirely or selectively) passing through it, or delays its forwarding. Another worrisome data attack is the misrouting attack in which the attacker relays packets to the wrong next-hop with the effect that it does not reach the intended destination. These attacks could result in a significant loss of data or degradation of network functionality, say through disrupting network connectivity.

Cryptographic mechanisms alone cannot prevent these attacks since many of them, such as the wormhole and the rushing attacks, can be launched without needing access to cryptographic keys or violating any cryptographic check. To mitigate such attacks, many researchers have used the concept of cooperative Local Monitoring (BLM) within a node’s neighborhood (e.g., [1][2], [6]-[8], [24][28]). In local monitoring, nodes oversee part of the traffic going in and out of their neighbors. Different types of checks are done locally on the observed traffic to make a determination of malicious behavior. For systems where arriving at a common view is important, the detecting node initiates a distributed protocol to disseminate the alarm. Many protocols have been built on top of local monitoring for intrusion detection (e.g., [3]), building trust and reputation among nodes (e.g. [1], [2]), protecting against control and data traffic attacks (e.g. [6]-[8]) and in building secure routing protocols (e.g., [8], [9]). These attacks are detected by a group of nodes, called guard nodes that perform local monitoring. The guard nodes are normal nodes in the network and perform their basic functionality in addition to monitoring. Under local monitoring, a guard node verifies, for a fraction of the packets, if it is being forwarded within the requisite delay bound, without modification and without fabrication.

In this paper, we introduce a new class of attacks against traffic in wireless multi-hop ad hoc networks called stealthy packet dropping. In stealthy packet dropping the attacker achieves the objective of disrupting the packet from reaching the destination, through malicious behavior at an intermediate node. However, the malicious node gives the impression to its neighbors capable of overseeing the packet that it has performed the required action (e.g., relaying the packet to the correct next-hop en route to the destination). This class of attacks is applicable to unacknowledged packets. Due to the resource constraints of bandwidth and energy, much traffic in multi-hop ad hoc wireless networks (e.g., sensor networks) is unacknowledged. This is particularly true for the more common data traffic or broadcast control traffic than for rare unicast control traffic.

In this paper, we introduce four modes of the stealthy packet dropping attack. We distinguish between an external malicious node, which does not possess the cryptographic keys in the network, and internal compromised nodes, which do. The latter kind of nodes is created by compromising an erstwhile legitimate node. Consider a scenario in which a node called $S$ is forwarding a packet to a compromised node called $M$. $M$ is supposed to relay the packet to the next-hop node $D$. The first form of the attack is called packet misrouting. In this mode, $M$ relays the packet to the wrong next-hop neighbor, i.e., a neighbor node other than $D$. The result is that the packet does not reach its intended next-hop ($D$) while $M$ appears to the guards as doing its job correctly. The second mode is called the power control attack. In this mode, $M$ controls its transmission power to relay the packet to a distance less than that of the $D$. Therefore, the packet does not reach the next-hop while the attacker avoids detection by many guards. The third form of the attack is called the controlled-jamming attack. In this mode, the attacker uses another colluding node (external or internal) in the range of $D$ to transmit data at the same time when
starts relaying the packet to $D$. Therefore, a collision occurs at $D$, which prevents the packet from being correctly received by $D$, while $M$ looks to be performing its functionality correctly. The final mode of stealthy packet dropping is called the identity delegation attack. In this mode, the attacker colludes with a node $E$ placed close to the source node $S$. $E$ is allowed to use $M$’s identity and transmit the packet. Since $E$ is almost at the same place as $S$, $D$ does not receive the packet. At the same time, the guards of $M$ are deceived that $M$ relayed the packet to the next-hop. In each of these attack types, the adversary not only can successfully perform the attack, but it also hides the presence of its malicious activities. Additionally, in each attack type, a legitimate node is accused of packet dropping.

We provide a protocol called DISA (Detection and Isolation of Sneaky Attackers in Locally-Monitored Multi-hop Wireless Networks) that is constructed on local monitoring and that can mitigate each attack type introduced above. The DISA mitigation technique takes two forms – having guard nodes maintain additional next-hop information gathered during route establishment, and adding some checking responsibility to each neighbor. The latter technique makes use of the fact that under three of the attacks, neighbors have differing views of a node in terms of amount of forwarding traffic generated by the node. Hence, a single one-hop broadcast cannot convince all the neighbors. On the other hand, we show that of the four modes of the stealthy packet dropping attack, BLM [6]-[8][24] is unable to detect any instance of three attack types while it is able to detect specific instances of the drop through power control attack. The attack instances BLM is able to detect depend on how the adversary constrains the range of the forwarded packet.

The work by Buchegger et al. [2][23] also relies on overhearing packet forwarding of neighbors and building reputation scores based on it. The attack class introduced here would be damaging to such a solution since the malicious actions cannot be detected and the adversary nodes will achieve high reputation scores. To the best of our knowledge, we are the first to propose a protocol suited to resource constrained wireless networks that can detect these four attack types.

We provide a theoretical analysis for the probability of success of the stealthy packet drop attack in a locally monitored network. We present the work for local monitoring in static sensor network. However, the technique is also valid under mobile situations. The further requirement would be a primitive for determining the neighbor relation securely. Several such protocols exist in the literature [5][25]-[27]. We also analyze the resource consumption cost of DISA. For example, our analysis shows that DISA maintains detection coverage above 70% for the transmission control packet drop attack type for the configuration in which BLM has no coverage. Additionally, we build a simulation model for the misrouting attack type using the ns-2 network simulator. We perform a comparative evaluation of local monitoring with and without DISA. For example, our simulation results show that DISA can deliver 85% of packets to the destination under 20% nodes compromised, while the deliver ratio drops to less than 50% in BLM under the same configuration. The likelihood of framing of legitimate nodes is also three-folds under BLM for the same network.

We summarize our contributions in this paper as follows:

1. We introduce the stealthy packet dropping class of attacks in locally-monitored networks. We detail four methods by which it can be launched without the malicious node being detected.
2. We provide mitigation remedy for each attack type with minimal addition to the resource consumption and responsibility of a node over baseline local monitoring.
3. We provide a mathematical analysis of the probability of success in launching the stealthy packet dropping attack and probability of detection in both BLM and DISA.
4. We show through simulations the security advantage of DISA over BLM.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3
provides the foundations and background knowledge. Section 4 describes the stealthy packet dropping attack. Section 5 describes DISA and presents its mitigation techniques. Sections 6 and 7 present the mathematical analysis and the simulation results respectively. Finally, Section 8 concludes the paper.

2 Related Work

In the last few years, researchers have been actively exploring many mechanisms to ensure the security of control and data traffic in wireless networks. These mechanisms can be broadly categorized into the following classes—authentication and integrity services, protocols that rely on path diversity, protocols that use specialized hardware, protocols that require explicit acknowledgements or use statistical methods, protocols that overhear neighbor communication.

The path diversity techniques increase route robustness by first discovering multi-path routes [9], [13] and then using these paths to provide redundancy in the data transmission between a source and a destination. The data is encoded and divided into multiple shares sent to the destination via different routes. The method is effective in well-connected networks, but does not provide enough path diversity in sparse networks. Moreover, many of these schemes are expensive for resource-constrained wireless networks due to the data redundancy. Additionally, these protocols could be vulnerable to route discovery attacks, such as the Sybil attack, that prevent the discovery of non-adversarial paths.

Examples of protection mechanisms that require specialized hardware include [5], and [11]. The authors in [5] introduce a scheme called packet leashes that uses either tight time-synchronization or location awareness through GPS hardware. The work in [11] relies on hardware threshold signature implementations to prevent one node from propagating errors or attacks in the whole network.

A technique proposed to detect malicious behavior involving selective dropping of data, relies on explicit acknowledgement for received data using the same channel [13], or an out-of-band channel [12]. This method would render stealthy packet dropping detectable at the end point. However, the method incurs high communication overhead and has to be augmented with other techniques for diagnosis and isolation of the malicious nodes. A natural extension would be to reduce the control message overhead by reducing the frequency of ack-ing to one in every N data messages (in the above papers N=1). However, this may delay the adversary detection which may result in significant damage. In contrast, in DISA, the node is detected and diagnosed locally by its neighbors. Statistical measures have been used by some researchers for detection, e.g., [14] to detect wormhole attacks.

A widely used technique for mitigating control and data attacks in multi-hop wireless networks is cooperative local monitoring by overhearing traffic in the vicinity. The idea of overhearing traffic in the vicinity (e.g. [1]-[3]) has been used to build trust relationships among nodes in networks (e.g. [1], [2]), detect and mitigate certain kinds of attacks (e.g. [3], [6]-[8]), or discover routes with certain desirable properties, such as being node disjoint (e.g. [13]). Work in [8] provides detection of a wide class of control attacks against static sensor networks. However, local monitoring, as used by all researchers to date, fails to mitigate the stealthy packet dropping attack.

3 Foundations

3.1 Attack Model and System Assumptions

Attack model: An attacker can control an external node or an internal node, which, since it possesses the keys, can be authenticated by other nodes in the network. An insider node may be created, for example, by compromising a legitimate node. A malicious node can perform packet dropping by itself or by colluding with other nodes. The collusion may happen through out-of-band channels (e.g., a wireline channel). However, we do not consider the denial of service attacks
through physical-layer jamming [22], or through identity spoofing and Sybil attacks [10]. There exist several approaches to mitigate these attacks—[22] for jamming and [10] for the Sybil attack. A malicious node can be more powerful than a legitimate node and can establish out-of-band fast channels (e.g., a wireline link) or have high-powered controllable transmission capability.

**System assumptions:** We assume that all the legitimate communication links are bi-directional. We assume that secure neighbor discovery has been performed and that every node knows both first and second hop information. This can be achieved through the protocol described in [21] as well as by approaches developed by other researchers [4]. Note that while this knowledge is enormously useful, this by itself cannot mitigate many attack types. For example, further work is needed to detect the wormhole attack. Intuitively this information subsets the nodes from which a given node will accept packets but does not eliminate the possibility of malicious nodes within that subset. Local monitoring assumes that the network has sufficient redundancy, such that each node has more than an application defined threshold number of legitimate nodes as guards. We assume a key management protocol, e.g., [15], exists such that any two nodes can communicate securely.

### 3.2 Background: Local Monitoring

Local monitoring is a collaborative detection strategy where a node monitors the control traffic going in and out of its neighbors. This strategy was introduced in [6] for static sensor networks and here we give the background needed to understand the concepts presented in this paper.

For a node, say \( \alpha \), to be able to watch a node, say \( N_2 \), \( \alpha \) must be a neighbor of both \( N_2 \) and the previous hop from \( N_2 \), say \( N_1 \). Then we call \( \alpha \) a guard node for \( N_2 \) over the link \( N_1 \rightarrow N_2 \). We use the notation \( R(N) \) to denote the set of all nodes that are within the radio range of node \( N \) and \( G(N_1, N_2) \) to denote the set of all guard nodes for \( N_2 \) over a link \( N_1 \rightarrow N_2 \). Formally,

\[
G(N_1, N_2) = R(N_1) \cap R(N_2) - N_2, \quad N_2 \in R(N_1).
\]

For example, in Figure 1, \( G(X,A) = \{M,N,X\} \). Information from each packet sent from \( X \) to \( A \) is saved in a **watch buffer** at each guard. The guards expect that \( A \) will forward the packet toward the ultimate destination, unless \( A \) is itself the destination. Each entry in the watch buffer is time stamped with a time threshold, \( \tau \), by which \( A \) must forward the packet. Each packet forwarded by \( A \) with \( X \) as a previous hop is checked for the corresponding information in the watch buffer. The check can be to verify if the packet is fabricated or duplicated (no corresponding entry in the buffer), corrupted (no matching hash of the payload), dropped or delayed (entry is not matched within \( \tau \)).

![Figure 1: X, M, and N are guards of A over X→A](image)

A malicious counter (\( MalC(i,j) \)) is maintained at each guard node, \( i \), for a node, \( j \), at the receiving end of each link that \( i \) is monitoring over a sliding window of length \( T_{win} \). \( MalC(i,j) \) is incremented for any malicious activity of \( j \) detected by \( i \). The increment to \( MalC \) depends on the nature of the malicious activity, being higher for more severe infractions. When the growth in the counter value maintained by a guard node \( i \) for node \( j \) (\( MalC(i,j) \)) crosses a threshold rate (\( MalC_{th} \)) over \( T_{win} \), node \( i \) revokes \( j \) from its neighbor list (called direct isolation since it will henceforth not perform any communication with node \( j \)), and sends to each neighbor of \( j \), an authenticated alert message indicating \( j \) is a suspected malicious node. When a neighbor \( N_i \) gets the alert, it verifies the authenticity of the alert message. When \( N_i \) gets enough alert messages about \( j \), it marks the status of \( j \) as revoked (called indirect isolation). The notion of enough number of alerts is quantified by...
the detection confidence index $\gamma$. Each node maintains a memory of nodes that it has revoked through a local blacklist so that a malicious node cannot come back to its neighborhood and claim to be blameless. This constitutes local isolation of a malicious node by its current neighbors.

4 Stealthy Dropping Description

In all the modes of stealthy packet dropping, a malicious intermediate node achieves the same objective as if it were dropping a packet. However, none of the guard nodes using BLM become any wiser due to the action. In addition, some legitimate node is accused of packet dropping. Next, we describe the four attack types for stealthy packet dropping.

4.1 Drop through Misrouting

In the misrouting attack, a malicious node relays the packet to the wrong next-hop, which results in a packet drop. Note that, in BLM [6], a node that receives a packet to relay without being in the route to the destination either drops the packet or sends a one-hop broadcast that it has no route to the destination. The authors in [6] argue that that latter case would be more expensive and dangerous since it gives malicious nodes valid excuses to drop packets. Therefore, they go with the first choice, even though it may result in some false accusations.

Consider the example scenario in Figure 2. Node $A$ sends a packet to the malicious node $M$ to be relayed to node $B$. Node $M$ simply relays the packet to node $E$ which is not in the route to the final destination of the packet. Node $E$ drops the packet. The result is twofold: (i) node $M$ successfully drops the packet without being detected since all the guards of $M$ over $A \rightarrow M$ (regions I & II) have been satisfied by the transmission of $M \rightarrow E$, and (ii) legitimate node $E$ will be wrongly accused by its guards over $M \rightarrow E$ (regions II & III) as maliciously dropping packets.

4.2 Drop through Power Control

In this type of attack, a malicious node relays the packet by carefully reducing its transmission power, thereby reducing the range and excluding the legitimate next-hop node. This kind of transmission power control is available in today’s commercial wireless nodes, such as the Crossbow Mica family of nodes.

Consider the scenario shown in Figure 3. A node $S$ sends a packet to a malicious node $M$ to be relayed to node $T$. Node $M$ drops the packet by sending it over a range that does not reach $T$ (the
dotted circle centered at $M$). Figure 3(a) shows the guards of $M$ that are satisfied by the controlled transmission of $M$ (region II) and the set of guards that detect $M$ (region I) as dropping the packet since they did not overhear $M$. Figure 3(b) shows all the guards of $M$ over $S\rightarrow M$. Figure 3(d) shows the set of guards of $T$ over $M\rightarrow T$ that wrongly accuse $T$ of dropping the packet. The farther $T$ is from $M$ the better it is for the attacker since more guards can be satisfied and therefore, the stealthier the attack. For this attack to succeed, the attacker must know the location of each neighbor and the detection confidence index $\gamma$. Typically security is not achieved through obfuscation and therefore protocol parameters such as $\gamma$ are taken to be known to all and location determination is routinely run upon deployment of nodes. When the number of guards that are not satisfied by the controlled-power transmission is greater than $\gamma-1$, an intelligent attacker will refrain from lowering the transmission power since it will be detected and isolated by all its neighbors either directly or indirectly (Section 3.2). Here too, a successful attack not only achieves the effect of dropping the packet, but also causes a subset of the guards of $T$ over $M\rightarrow T$ to accuse node $T$ of dropping the packet.

4.3 Drop through Colluding Collision

In many wireless sensor network deployment scenarios, the 802.11 MAC protocol RTS-CTS mechanism that reduces frame collisions due to the hidden terminal problem and the exposed terminal problem are disabled for the sake of energy saving. This is also explained by the fact that packets in some wireless networks such as sensor networks are often quite small and fall below the threshold for packet length for which RTS/CTS is turned on.

The attacker may exploit the absence of the RTS/CTS frames to launch a stealthy packet dropping attack through collision induced by a colluding node. The colluding node creates a collision in the vicinity of the expected next-hop node at an opportune time. Consider the scenario shown in Figure 4. The malicious node $M_i$ receives a packet from $S$ to be relayed to $T$. Node $M_i$ coordinates its transmission with a transmission of some data generated by its colluding partner $M_j$ to $T$. It has the effect that $T$ is unable to get the packet relayed by $M_i$. The damage caused by this attack is twofold: (i) $M_i$ successfully drops the packet due to a collision at $T$ without being detected, and (ii) node $T$ is accused of dropping the packet by some of its guards over the link $M_i\rightarrow T$ (the guards that are out of the range of $M_j$, region I). Note that for $M_j$ to be able to send data to $T$, it has to be a legitimate neighbor (compromised by the attacker), otherwise, the attack would be considered a physical layer jamming [22], which is assumed to be detectable through techniques complementary to that presented in the paper (e.g., [8][22]).

4.4 Drop through Identity Delegation

In this form of the attack, the attacker uses two malicious nodes to drop the packet. One node is spatially close to the sender. The other node is the next-hop from the sender. The first malicious node could be external or an internally compromised node while the latter has to be an internally compromised node.

Consider the scenario shown in Figure 5, node $S$ sends a packet to a malicious next-hop node $M_j$ to be relayed to node $T$. The attacker delegates the identity and the credentials of the compromised node $M_j$ to a colluding node $M_i$ close to $S$. After $S$ sends the packet to $M_j$, $M_j$ uses the delegated identity of $M_j$ and transmits the packet. The intended next-hop $T$ does not hear the message since $T \notin R(M_j)$. The guards of $M_j$ over $S\rightarrow M_j$ are the nodes in the shaded areas I & II and they are
all satisfied since they are in $R(M_1)$. Again, the consequences of this attack are twofold: (i) the packet has been successfully dropped without detection, and (ii) the set of nodes in the shaded area II overhear a packet transmission (purportedly) from $M_2$ to $T$. These nodes are included in $G(M_2, T)$ and will subsequently accuse $T$ of dropping the packet.

Figure 5: Identity delegation attack scenario

5 Stealthy Dropping Mitigation

In this section we propose two mechanisms to augment traditional local monitoring to detect stealthy packet dropping. The first mechanism mitigates the misrouting stealthy packet drop while the second mitigates the rest of the attacks.

5.1 Mitigating Misrouting Packet Drop

To detect this attack, the local monitoring has to incorporate additional functionality and information. The basic idea is to extend the knowledge at each guard to include the identity of the next-hop of the packet being relayed.

This additional knowledge can be collected during route establishment. Many multi-hop wireless routing protocols provide this knowledge without any modification while some changes are necessary in others. The first class includes both reactive routing protocols such as Dynamic Source Routing (DSR) and its variants [16] and proactive routing protocols such as TinyOS beacon routing [18] and Destination Sequenced Distance Vector routing (DSDV [19]). In all source routing protocols, the packet header carries the identity of all the nodes in the route from the source to the destination. Therefore, no additional traffic is required to be generated for the guard nodes to be able to detect this kind of attack. Moreover, no additional information is required to be maintained at the guards since each packet carries the required information in its header. In TinyOS beacon routing, the base station periodically broadcasts a beacon to establish a breadth first search tree rooted at the base station. Each node within the transmission range of the base station overhears the beacon, sets its parent to be the base station, sets the hop count to the base station to be one, and rebroadcasts the beacon. Each beacon carries the identity of the broadcasting node, the identity of its parent, and the hop count to the base station. Each guard overhearing the beacon broadcasting saves parent node identity for each neighbor. Later, when a node, say $B$, is sent a packet to relay, the guard of $B$ can detect any misrouting by $B$ since it knows the correct next-hop en route to the base station.

The second class of routing protocols requires modification to the protocol to build the next-hop information at the guards. Examples of these protocols are the reactive routing protocols that use control packet flooding of route requests ($REQ$) and route replies ($REP$) to establish the route between the source and the destination (e.g., LSR [8] and AODV [17]). In these protocols, when a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a route discovery process to locate the other node. It broadcasts a route request ($REQ$) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a “fresh enough” route to the destination is located. Along with its own sequence number and the broadcast ID, the source node includes in the $REQ$ the most recent sequence number it has for the destination. During the process of forwarding the $REQ$, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the
broadcast packet is received, thereby establishing a reverse path. Once the \textit{REQ} reaches the destination, the destination node responds by unicasting a route reply (\textit{REP}) packet back to the neighbor from which it first received the \textit{REQ}. As the \textit{REP} traverses along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the \textit{REP} came.

Next, we show the required changes to the basic version of AODV to enable the guards to build the necessary knowledge for detecting the misrouting attack. The idea behind the solution is that during route establishment, when the relation about which node to forward a packet between a given source-destination pair is determined, this information is broadcast by a neighbor to the guards which will be responsible for monitoring the node. To collect the next-hop identity information, the forwarder of the \textit{REQ} attaches the previous two hops to the \textit{REQ} packet header. Let the previous hop of \textit{M} be \textit{A} for a route from source \textit{S} to destination \textit{D}, and the next hop from \textit{M} be \textit{B} (Figure 2). When \textit{M} broadcasts the \textit{REQ} received from \textit{A}, it includes the identity of \textit{A} and its own identity (\textit{M}) in the \textit{REQ} header \textit{<S, D, REQ\_id, A, M>}. When \textit{B} and the other neighbors of \textit{M} get the \textit{REQ} from \textit{M}, they keep in a Verification Table (\textit{VT}) \textit{<S, D, RREQ\_id, A, M, ->} (last field is currently blank). When \textit{B} broadcasts the \textit{REQ}, the common neighbors of \textit{M} and \textit{B} update their \textit{VT} to include \textit{B} \textit{<S, D, RREQ\_id, A, M, B>}. When \textit{B} receives a \textit{REP} to be relayed to \textit{M}, it includes in that \textit{REP} the identity of the node that \textit{M} needs to relay the \textit{REP} to, which is \textit{A} in this example. Therefore, all the guards of \textit{M} now know that \textit{M} not only needs to forward the \textit{REP} but also that it should forward it to \textit{A} and not any other neighbor.

Therefore, two tasks have been added to the functionality of the guards in monitoring the \textit{REP} packets. First, the guard \textit{G} of a node \textit{N} verifies that \textit{N} forwards the \textit{REP} to the correct next-hop. In the example above, \textit{G2} verifies that \textit{M} forwards the \textit{REP} to \textit{A}. Second, \textit{G} verifies that \textit{N} has updated the forwarded \textit{REP} header correctly. In the example shown above, \textit{G2} verifies that when the input packet to \textit{M} from \textit{B} is \textit{<REP, S, D, REQ\_id, C, B, M>}, then the output packet from \textit{M} should be \textit{<REP, S, D, REQ\_id, B, M, A>}. Thus \textit{M} and its guards over the link \textit{B\rightarrow M} know that the next-hop is \textit{A} from the information built in the \textit{VT} table during the \textit{REQ} flooding.

Using the additional information mentioned above, DISA detects misrouting attacks as follows. In the example above, assume that \textit{S} is sending a data packet to \textit{D} through a route that includes \textit{<Y, A, M, B, C>}. The malicious node \textit{M} cannot misroute the data packet received from \textit{A} to a node other than the next-hop, \textit{B} since each guard of \textit{M} over the link \textit{A\rightarrow M} has an entry in its \textit{VT} which indicates \textit{B} is the correct next-hop. This results in an additional checking activity for the guard node involved in local monitoring–verifying the data packet is forwarded to the correct next hop, as indicated by the entry in the guard node’s \textit{VT}. Moreover, \textit{M} cannot frame another neighbor, say \textit{X}, by misrouting the packet to \textit{X}. The guards of \textit{X} over \textit{M\rightarrow X} do not have an entry like \textit{<S, D, RREQ\_id, Y, A, M, X>} and therefore, they would not increment the MalC of \textit{X} when it drops the packet.

5.2 Mitigating other Stealthy Attacks

The key observation behind the other types of the stealthy packet dropping attack is that the attack defeats local monitoring based detection by reducing the number of guards that overhear a packet to zero or to a number that is less than the confidence index. In the power control attack shown in Figure 3(a), the attacker narrows the guards that can detect the packet drop into the lightly shaded area (region I in Figure 3(a)) while the majority of the guards (region II in Figure 3(a)) are satisfied. In the colluding collision attack (Figure 4) and identity delegation attack (Figure 5), the attacker completely evades detection by satisfying all the guards (the nodes in region I of Figure 4 and of Figure 5).

The countermeasure we propose against these attacks is based on the observation that an adversary evades detection of dropping packets by allowing only a subset of guards to overhear the message being forwarded. Therefore, we expand
the set of nodes that can guard a node from only the common neighbors of the node being monitored and its previous-hop node to include all the neighbors. Since all neighbors are included in verifying the node, by definition, some neighbor will see evidence of stealthy packet drop. The detection technique makes use of the fact that, under the stealthy packet dropping attacks, neighbors have differing views of a node in terms of the volume of traffic it has forwarded and all the neighbors cannot be convinced by a single broadcast. To achieve this goal we need to introduce additional tasks for the nodes in the network. (i) Each node keeps a count of the number of messages each of its neighbors had forwarded over a predetermined time interval and (ii) each node has to announce the number of packets it has forwarded over some period of time. The adversary evades detection of stealthy packet dropping by allowing only a subset of guards to overhear the packet being forwarded. Thus, the subset of guards that had overheard the packet forwarding would have a higher count than the nodes that did not overhear the forwarding. By forcing a node to announce the number of messages it has forwarded over some period of time, a malicious node would have the problem of satisfying two sets of neighbors that expect to hear different counts through a single broadcast.

A neighbor of a node, say $N$, that collects the number of forwarded packets by $N$ and compares the result with the count announced by $N$ is called a comparator of $N$, denoted by $C(N)$. For any node $N$ all nodes in radio range $R(N)$ act as comparators of $N$. Recall that a guard of a node $B$ over the link $Y \rightarrow B$, has been defined in the BLM as any node that lies within the transmission range of both $Y$ and $B$. Therefore, each guard of $N$ over a certain link is a comparator of $N$, however, not every comparator of $N$ is a guard of $N$. The function of a comparator is to count the total number of packets forwarded from the node within a time period. During some time periods node $N$ may be required to announce the number of messages it has forwarded in that period. If a comparator’s count is not within an acceptable range of the announced forward count, the comparator increments its malicious counter for the announcing node.

In order to reduce radio traffic, we do not require all nodes to announce their forward count for every time period. Instead a node must announce within the time period that it receives a broadcast message request to announce. Whenever a node, say $A$, overhears a packet from a node $N$ that is not within the neighbor list of $A$, node $A$ broadcasts a 3-hop request for $N$ to announce its forward count. If node $N$ and all of its neighbors are within 3 hops of the requestor then the neighbors of $N$ will act as comparators of $N$ and expect to hear the correct forward count announced. The basic idea is that a malicious node that has dropped a packet faces a dilemma; some of its neighbors have overheard the dropped packet and expect it to be included in the send count while other neighbors have not heard the packet so they expect a send count of one less message. However, note that a suspicion would not be raised by a discrepancy of one due to natural losses (channel conditions and collisions). Detection is triggered only when the discrepancy crosses a predetermined threshold.

For simplicity of exposition, for the following examples, we will consider that a discrepancy of a single packet is sufficient for detection. Consider the power drop attack scenario shown in Figure 3(a), the neighbors of $M$ within the dotted circle would have one more count for the number of packets forwarded by $M$ as compared to the counters in the rest of $M$’s comparators. In each of the last three attack modes, the attacker is faced by two sets of neighbors that have different views about him. The best the attacker can do is to satisfy the larger set, however, the nodes of the other set would detect the discrepancy and propagate the detection knowledge to the nodes of the other set. All the nodes of the smaller set would then directly isolate the malicious node. The nodes of the larger set indirectly isolate the malicious node if the number of nodes in the smaller set is greater than or equal to the detection confidence index $\gamma$. 
6 Analysis

The analysis gives the isolation and false isolation probabilities under colluding collision and power control attacks for both BLM and DISA.

Assumptions: We consider a homogeneous network of nodes where the nodes are uniformly distributed in the field with density \( d \). For simplicity, we assume that the field is large enough that edge effects can be neglected.

Attacker model: The malicious node uses an omni-directional antenna. Its goal is to have the effect of dropping the packet from reaching the legitimate next-hop node. The detection probability in the case of power control attack is a lower bound since we assume that the adversary can control the transmission power level to be infinitesimally smaller than that required to reach the next hop.

6.1 Colluding Collision

Consider Figure 6, let the distance between \( S \) and \( M_1 \) be \( x \), the distance between \( M_1 \) and \( M_2 \) be \( y \), the distance between \( M_1 \) and \( T \) be \( z \), and the distance between \( S \) and \( M_2 \) be \( w \).

The intersection area between \( R(S) \) and \( R(M_1) \) is given by,
\[
A_{\text{inter}}(S \_ M_1) = 2r^2 \cos^{-1}\left(\frac{x}{2r}\right) - \frac{1}{2} x \sqrt{4r^2 - x^2}
\] (1)

The intersection area between \( R(S) \) and \( R(M_2) \) is given by,
\[
A_{\text{inter}}(S \_ M_2) = 2r^2 \cos^{-1}\left(\frac{w}{2r}\right) - \frac{1}{2} w \sqrt{4r^2 - w^2}
\] (2)

The guards of \( M_1 \) over the link \( S \rightarrow M_1 \) that do not experience the plotted collision are given by,
\[
G_{M_1} = [A_{\text{inter}}(S \_ M_1) - A_{\text{inter}}(S \_ M_2)] \cdot d
\] (3)

Since \( M_2 \) is a colluding node with \( M_1 \), the best situation for the attackers is when \( R(M_2) \) does not overlap with \( R(S) \) \((w \geq 2r)\). This guarantees that all the guards of \( M_1 \) over the link \( S \rightarrow M_1 \) will overhear \( M_1 \) without experiencing the plotted collision. In this case, the probability of detection or isolation of \( M_1 \) is zero.

The intersection area between \( M_1 \) and \( M_2 \) which include \( T \) is given by,
\[
A_{\text{inter}}(M_1 \_ M_2) = 2r^2 \cos^{-1}\left(\frac{y}{2r}\right) - \frac{1}{2} y \sqrt{4r^2 - y^2}
\] (4)

The intersection area between \( M_1 \) and \( T \) is given by,
\[
A_{\text{inter}}(M_1 \_ T) = 2r^2 \cos^{-1}\left(\frac{z}{2r}\right) - \frac{1}{2} z \sqrt{4r^2 - z^2}
\] (5)

Therefore, the guards of \( T \) over the link \( M_1 \rightarrow T \) that will not experience the plotted collision are those on area I in Figure 1. Those guards wrongly accuse \( T \) as dropping the packet and are given by,
\[
G_T = [A_{\text{inter}}(M_1 \_ T) - A_{\text{inter}}(M_1 \_ M_2)] \cdot d
\] (6)

The only way detection can happen in BLM is if a node hears \( S \rightarrow M_1 \) communication but not \( M_1 \) forwarding the packet to \( T \). The latter can happen only if the node is a neighbor of \( M_2 \). Thus, more than or equal to \( \gamma \) nodes from among those that are neighbors of \( S \) and \( M_2 \) must hear the \( S \rightarrow M_1 \) communication for isolation to occur.

6.1.2 DISA

In DISA, \( M_1 \) will have two sets of comparators, the first set includes those comparators that do not experience the collisions and thus will overhear \( M_1 \) packet forwarding. We call this set the Plus set \( S_p \). The other set includes those comparators that suffer from the plotted collision and thus will not overhear \( M_1 \) forwarding. We call this set the Minus set \( S_m \).
\[ S_m = A_{\text{incr}} (M_1 \rightarrow M_2) \cdot d \quad (7) \]
\[ S_p = \pi r^2 d - S_m \quad (8) \]

For isolation to occur, the number of nodes in the smaller of the two sets that hear the broadcast from \( M_1 \) must be greater than or equal to \( \gamma \).

\[ \text{Figure 7: Probability of false isolation with the number of neighbors under colluding collision attack. } r = 30, x = 30, \gamma = 30, y = 45. \]

\[ \text{Figure 8: Probability of false isolation with } y \text{ under colluding collision attack. } r = 30, x = 30, \gamma = 3, d = 0.005 \text{ (14 neighbors).} \]

We plot the probability of false isolation as a function of the average number of neighbors (Figure 7) and as a function of the distance between malicious nodes (Figure 8) under the colluding collision attack with BLM. As the distance between the two colluding nodes \( (y) \) increases, false isolation becomes more likely since there are more nodes that hear \( M_1 \rightarrow T \) and are not affected by the collision from \( M_2 \). The likelihood of false detection also increases with the density since there are more guards that are fooled by the adversary into accusing the legitimate node \( T \). Note that the probability of false isolation in DISA is zero.

We plot the probability of isolation with the average number of neighbors for both BLM and DISA in Figure 9. As the average number of neighbors increases, the number of available guards and comparators increase. In BLM, the only way isolation happens if more than \( \gamma \) nodes that are neighbors of both \( S \) and \( M_2 \) hear \( S \rightarrow M_1 \) communication. With increasing the average number of neighbors, the number of such nodes increases. A similar behavior is seen in DISA since the number of nodes that suffer from the plotted collision increases with increasing node density. However, high isolation probability is achieved at much lower density in DISA (7 neighbors) than that in BLM (28 neighbors).

\[ \text{Figure 9: Probability of isolation with the number of neighbors under colluding collision attack. } r = 30, x = 30, \gamma = 3, y = 15 \]

We plot the probability of isolation with the distance between the two malicious nodes \( (y) \) for both BLM and DISA in Figure 10. As the distance between the colluding malicious nodes increases,
the distance between the source \( S \) and the malicious node \( M_2 \) also increases. The only way isolation happens in BLM if more than \( \gamma \) nodes that are neighbors of both \( S \) and \( M_2 \) hear \( S \rightarrow M_1 \) communication. With increasing distance between \( S \) and \( M_2 \), the number of such nodes decreases. A similar behavior is seen in the probability of isolation with \( y \) for DISA. However, the fall (with distance between colluding malicious nodes) happens much later than that in BLM.

**Figure 10**: Probability of isolation with \( y \) under colluding collision attack. \( r = 30, x = 30, \gamma = 3, d = 0.005 \) (14 neighbors)

### 6.2 Power Control Attack

Consider any two randomly selected neighbor nodes, \( S \) and \( M \), as shown in Figure 3(a). Nodes \( S \) and \( M \) are separated by a distance \( X \), and the communication range is \( r \). \( X \) is a random variable that has the probability density function of \( f_X(x) = 2x/r^2 \) with range \((0,r)\). This follows from the assumption of uniform distribution of the nodes.

#### 6.2.1 Basic Local Monitoring (BLM)

The guards of \( M \) over the link from \( S \rightarrow M \) lie on the shaded area shown in Figure 3 (b). The subset of guards that can be satisfied by the controlled power transmission of \( M \) lies on the shaded area shown in Figure 3 (c), we call these guards the happy guards \( N_h \). Finally, the subset of guards of \( T \) over \( M \rightarrow T \) that wrongly accuse \( T \) of dropping the packet are shown in the shaded area of Figure 3 (d), we call these guards the fooled guards, \( N_f \).

The shaded area in Figure 3 (c) is found to be,

\[
\text{Area}(c) = \begin{cases} 
  r^2 \cos^{-1}\left(\frac{r}{2y}\right) + y^2 \left(\pi - 2\cos^{-1}\left(\frac{r}{2y}\right)\right) & \text{when } y > r/2 \\
  \pi y^2 & \text{when } y \leq r/2 
\end{cases}
\]

which is the same as the shaded area in Figure 3(d). Recall that \( Y \) is a random variable with probability density function of \( f_Y(y) = 2y/r^2 \) with range \((0,r)\). Therefore, \( N_h = N_f = \text{Area}(c) \times d \). Finally, the number of guards that can detect the power control attack is \( N_d = N_g - N_h \). The condition for a successful attack is \( N_d < \gamma \).

#### 6.2.2 DISA

The expected number of comparators of any node is \( N_c = \pi r^2 d \). The subset of comparators that can overhear \( M \) are those that lie within the dotted circle of Figure 3(c), we call these comparators the Plus Comparators \( C_p \). The subset of comparators that cannot overhear the transmission of \( M \) are those that lie within the legitimate transmission range of \( M \) but out of the dotted circle, we call these the Minus Comparators \( C_m \).

\[
C_p = \pi y^2 d \\
C_m = N_c - C_p = \pi (r^2 - y^2) \cdot d
\]

The condition for successful attack is \( \min(C_p, C_m) < \gamma \), since the intelligent adversary broadcasts a message count that satisfies the larger of the two sets.

In Figure 11, we plot the probability of isolation as a function of \( \gamma \) under the power control attack for both BLM and DISA. For the plot, transmission range is 50 m, distance between \( S \) and \( M \) is the transmission range (which gives the smallest number of guards on \( S \rightarrow M \) and hence the lower bound on the detection probability), and each node has on an average 40 neighbors \( (d = 0.005) \). The analytical result shows that as \( \gamma \) increases, the probability of isolation decreases sharply with BLM. For a reasonable \( \gamma \) value of 3, the isolation probability with BLM is less than 0.5. A similar behavior is seen with DISA. However, the figure shows that DISA is considerably more effective.
than BLM in isolating malicious nodes indulging in packet dropping through the power control attack. This is due to the design of DISA which has more comparators than the guards in BLM.

Finally, we analyze the additional resource requirements of DISA over BLM. These are (i) state maintenance of the next-hop node, counter for the number of packets forwarded by each neighbor both of which are linear in terms of number of neighbors, (ii) broadcast of the counters in an on-demand basis, triggered by a relatively rare event, (iii) two node identifiers in each route request and reply packet. These are not onerous additions even for a resource constrained environment.

7 Simulation results

We use the ns-2 simulation environment [20] to simulate a data exchange protocol, individually with BLM and with DISA. We distribute the nodes randomly over a square field with a fixed average node density. Thus, the field size varies with the number of nodes. We use a generic on-demand shortest path routing protocol that floods route requests and unicasts route replies in the reverse direction. A route, once established, is not used forever but is evicted from the cache after an idle period $\text{TOut}_{\text{Route}}$ if no other packet has been forwarded to the particular destination. We simulate the misrouting attack as a representative of stealthy packet dropping. When a malicious node gets a data packet, it relays the packet to a wrong next-hop with a probability of $f_{\text{dat}}$. A malicious node does not generate any data of its own. The simulation also accounts for losses due to natural collisions. The guards inform all the neighbors of the detected malicious node through multiple unicasts. For each run, malicious nodes are chosen at random.

\textbf{Input parameters}: Each node acts as a data source and generates data using an exponential random variable with inter-arrival rate $\phi$. The destination is chosen at random and used for a random time following an exponential distribution with rate $\xi$. We use $N_M$ for the number of malicious nodes and $N$ for the total number of nodes. The input parameters with the experimental values are given in Table 1.

\textbf{Output parameters}: The output parameters include (i) the fraction of data packets received (\textit{delivery ratio}) calculated as the total number of packets successfully received by final destinations over the total number of packets sent, (ii) the \textit{wrong isolation probability} which includes both (a) the framing ratio, which is defined as the fraction of good nodes that have been incorrectly isolated due to the attack over the total number of good nodes, and (b) the false isolation ratio, which is defined as the fraction of good nodes that have been isolated due to natural causes (collisions and losses on the wireless channel) over the total number of good nodes, (iii) the malicious node \textit{isolation probability}, which is defined as the number of malicious nodes isolated to the total number of malicious nodes, (iv) the average \textit{end-to-end delay} of data packets, which is the time a packet takes after leaving the source until it reaches its final destination, and (v) the average \textit{isolation time}, which is the time it takes from the first malicious action of a node until it gets isolated, averaged over all the isolated malicious nodes. Note that here we only consider framing as a result of the misrouting attack and we do not consider the kind of framing where enough number of malicious nodes in a neighborhood

![Figure 11: Probability of isolation with $\gamma$ under power control attack. $r = 50, x = 50, d = 0.005$.](image)
frame a legitimate neighbor. The latter kind of framing is identical to that in BLM and has already been analyzed in [6].

The output parameters are measured at the end of the simulation time (2000 seconds). The output parameters are obtained by averaging over 30 runs. The reasoning provided for some experimental results was arrived at by careful examination of the simulation logs. When a claim is made of difference between DISA and BLM, the difference is significant at 95% confidence level.

**Table 1: Input parameters for DISA simulation**

<table>
<thead>
<tr>
<th>Param.</th>
<th>Value</th>
<th>Param.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx Range (r)</td>
<td>50 m</td>
<td>φ</td>
<td>1/70</td>
</tr>
<tr>
<td>ξ</td>
<td>1/280</td>
<td>fdat</td>
<td>1.0</td>
</tr>
<tr>
<td>T_win</td>
<td>100 s</td>
<td>N_M</td>
<td>0-20</td>
</tr>
<tr>
<td>γ</td>
<td>1-7</td>
<td>τ</td>
<td>0.03 sec</td>
</tr>
<tr>
<td># nodes (N)</td>
<td>100</td>
<td>BW</td>
<td>40 kbps</td>
</tr>
<tr>
<td>T\text{OutRoute}</td>
<td>50 s</td>
<td>MalC_th, C_t</td>
<td>50, 150</td>
</tr>
</tbody>
</table>

**7.1 The Effect of the Number of Malicious Nodes (N\_M)**

Figure 12 shows the variations in delivery ratio as the number of malicious nodes varies. The figure shows that the delivery ratio decreases as N\_M increases. This is due to the packets dropped before the malicious nodes are detected and isolated. As N\_M increases, this initial drop increases and thus the delivery ratio decreases. Moreover, as N\_M increases, the isolation probability decreases. Therefore, the malicious nodes that could not be detected continue to drop packets and this decreases the delivery ratio. The delivery ratio in BLM is much less than in DISA and the difference increases as the number of malicious nodes increases. This is due to two main reasons. The first is that BLM fails to detect most of the malicious nodes and thus they continue to drop packets constantly. The second is that some of the good nodes in BLM get framed by the adversary and thus become isolated, which reduces the overall throughput.

**Figure 12: Effect of N\_M on delivery ratio**

Figure 13 shows the variations in isolation probability as N\_M varies. Most significantly, we observe that DISA has much better isolation probability compared to BLM (almost 80% for DISA and 40% for BLM). The figure shows that the isolation probability slightly decreases as we increase N\_M. This is because the number of available guards and comparators in the network decreases as more and more nodes get compromised.

**Figure 13: Effect of N\_M on true isolation**

Figure 14 shows the variations in wrong isolation as N\_M varies. The figure shows that the wrong isolation increases as we increase N\_M. Wrong isolation is a composite of false isolation and framing. False isolation decreases as we increase N\_M. This comes from the fact that the number of good nodes decreases as we increase N\_M. This in turn results in a decrease of the indirect false isolation since a node may not have more than γ good nodes to agree on falsely isolating a neighbor. Moreover, as N\_M increases,
the data traffic decreases since malicious nodes are not generating data. This in turn decreases the chance for collisions and consequently decreases false isolation. Framing intuitively increases as we increase $N_M$ as more malicious nodes negatively affect more good nodes. False isolation is almost the same for both DISA and BLM since we run the simulation under the same setup for both. In DISA, framing is negligible, and therefore, the wrong isolation curve follows the trend of false isolation. In BLM wrong isolation is much higher than that in DISA because framing in BLM is much higher. Moreover, because framing is dominant in BLM, the trend of wrong isolation follows that of framing.

**Figure 14: Effect of $N_M$ on false isolation**

Figure 15 shows the variations in end-to-end delay as $N_M$ is varied. The figure shows that the end-to-end delay initially increases as $N_M$ increases and then starts to decrease. As $N_M$ increases, the route reestablishment frequency increases. This is due to the fact that a route remains active for a time $T_{Out\text{Route}}$ and this timer is reset with every packet forwarded using that route. Consequently cutting the flow of packets (by maliciously dropping the packet) causes the route entries to stale. Therefore, additional traffic is generated to reestablish the route which increases the end-to-end delay. The opposing pull comes from the fact that as $N_M$ increases the traffic decreases. This reduces the contention in the network which in turn decreases the end-to-end delay. As $N_M$ increases beyond a point, the latter factor dominates and the overall result is a decrease in the end-to-end delay. The end-to-end delay in DISA is slightly higher than in BLM. This is due to the modification of the routing protocol in DISA which makes the route establishment time slightly higher.

**Figure 15: Effect of $N_M$ on end-to-end delay**

**Figure 16: Effect of $N_M$ on isolation time**

Figure 16 shows the variations in the average isolation time as $N_M$ varies. The most significant thing to note here is that the average isolation time in BLM is 3 to 7 times longer than that in DISA. This is due to the fact that DISA is more efficient in detection and isolation compared to BLM. The figure also shows that the average isolation time increases as $N_M$ increases. As $N_M$ increases local isolation becomes less effective since the number of legitimate neighbors decreases and if this goes below $\gamma$, then local isolation has to wait for direct isolation individually by each legitimate neighbor. Moreover, as $N_M$ increases the data traffic in the network decreases (in the simulation malicious nodes do not send data) which results in a decrease in the number of packets that a single malicious node may drop. This in turn results in
elongating the isolation time of malicious nodes. Finally, the figure shows that the average isolation time decreases in BLM as $N_M$ increases. As we increase $N_M$, the number of guards in BLM decreases sharply which makes it harder to isolate nodes. Therefore, with high number of malicious nodes, only those nodes with large neighborhoods may be isolated. In such neighborhoods, isolation takes shorter which makes the overall average isolation time to be shorter.

### 7.2 The Effect of $\gamma$

Figure 17 shows the variations in the isolation probability as $\gamma$ varies. The figure shows that as the value of $\gamma$ increases the percentage of isolation decreases since it becomes more difficult to get agreement on malicious behavior from at least $\gamma$ comparators. Most importantly, the figure shows that the percentage of isolation in DISA is much better than that in BLM due the better detection strategy used by DISA.

![Figure 17: Effect of $\gamma$ on isolation probability](image)

Figure 18 shows the variations in delivery ratio as $\gamma$ varies. It shows that the delivery ratio in DISA is almost twice as that in BLM. This is due to the effective detection and isolation of malicious nodes in DISA which eliminates the capability of malicious nodes to continue dropping packets after being isolated. The figure also shows that the delivery ratio decreases as $\gamma$ increases due to the decrease in isolation probability.

![Figure 18: Effect of $\gamma$ on delivery ratio](image)

Figure 19 shows the variations in the percentage of wrong isolation (isolation of good node due to both natural and malicious activities). The figure shows that percentage of wrong isolation exponentially decreases with increasing $\gamma$. As $\gamma$ increases it becomes harder and harder to get agreement of at least $\gamma$ comparators to isolate a node (see the explanation of Figure 14 for more details). Most importantly the figure shows that the percentage of wrong isolation in DISA is almost zero since DISA does not suffer from framing as BLM.

![Figure 19: Effect of $\gamma$ on wrong isolation](image)

Figure 20 shows the variation in average isolation time as $\gamma$ varies. The figure shows that the average isolation time increases with increasing $\gamma$ to a point after which it starts to decrease. Initially, as $\gamma$ increases it becomes harder to isolate node due the required agreement of at least $\gamma$ nodes which makes the isolation time longer. After increasing $\gamma$ beyond a point, only the nodes which have enough comparators get isolated. Therefore, the time averaging includes fewer nodes with high probability of isolation which makes the average isolation time shorter.

![Figure 20: Effect of $\gamma$ on average isolation time](image)
Most importantly, note that the average isolation time in DISA is much shorter than that in BLM (almost twice as that of DISA).

Figure 20: Effect of $\gamma$ on average isolation time

7.3 The Effect of Node Density

As the node density increases the average number of neighbors per node increases. This in turn increases the number of available guards for BLM and comparators for DISA. The node density can be increased by increasing the number of deployed nodes, increasing the transmission range, or both.

Figure 21: Effect of node density on isolation probability

Figure 21 shows the variation in the isolation probability as the number of neighbors varies (by varying the transmission range) with $\gamma$ set to 5 and $N_M$ set to 20. The intuition we presented above is borne out by these results—the probability of isolation increases with increasing number of neighbors for both BLM and DISA. Importantly, the figure shows that DISA always performs much better than BLM due to the enhanced detection capabilities.

8 Discussion

Here we have described the design of DISA, which fundamentally relies on the ability of some guard nodes to overhear the behavior of neighboring nodes. This basic feature of wireless networks has been leveraged by many researchers, for almost a decade now starting from [28]. Any technique that relies on this has the shortcoming that it can be bypassed by a powerful adversary that can accurately place malicious nodes or precisely control transmission power of a malicious node. Intrinsically, the placement or the transmission power control can be used to hide the behavior from the requisite number of guard nodes, e.g., the next hop node does not get the packet but the guards see it. In that case, no detection will occur. DISA suffers from this shortcoming as does all the work that relies on the feature.

The memory cost of a technique like DISA may be of concern since overheard packets have to be maintained in memory. However, the common case behavior is that of nodes behaving legitimately. Therefore, the packets are forwarded quickly and do not have to be kept in memory for long. Our experiments on a real testbed have shown that a buffer size of 5 is adequate for a density where each node has 8 neighbors. The method to limit the overhearing energy cost has been shown in [24].

9 Conclusion

We have introduced a new class of attacks called stealthy packet dropping which disrupts a packet from reaching the destination by malicious behavior at an intermediate node. This can be achieved through one of four attack types—misrouting, controlling transmission power, malicious jam at an opportune time, and malicious identity sharing. However, the malicious behavior cannot be detected by any behavior-based detection scheme presented to date. Specifically,
we showed that local monitoring based detection which relies on overseeing behavior of a neighboring node cannot detect these attacks. Additionally, it will cause a legitimate node to be accused. We then presented a protocol called DISA based on local monitoring to remedy each attack. The solution takes two forms – having nodes maintain additional routing path information, and adding some checking responsibility to each neighbor. We showed through analysis and simulation that BLM fails to mitigate any of the presented attacks while DISA successfully mitigates them.

In future work, we are considering detection techniques for multi-channel wireless networks. The listening activity for detecting malicious behavior is more complicated due to the presence of multiple channels. We also plan to analyze the impact of the detection technique on the network throughput under different adversary models.

10 References


