Mitigating Selective Forwarding Attacks with a Channel-Aware Approach in WMNs

Devu Manikantan Shila, *Student Member, IEEE*, Yu Cheng, *Senior Member, IEEE*, and Tricha Anjali, *Senior Member, IEEE*

***Abstract*—In this paper, we consider a special case of denial of service (DoS) attack in wireless mesh networks (WMNs) known as selective forwarding attack (a.k.a gray hole attacks). With such an attack, a misbehaving mesh router just forwards a subset of the packets it receives but drops the others. While most of the existing studies on selective forwarding attacks focus on attack detection under the assumption of an error-free wireless channel, we consider a more practical and challenging scenario that packet dropping may be due to an attack, or normal loss events such as medium access collision or bad channel quality. Specifically, we develop a channel aware detection (CAD) algorithm that can effectively identify the selective forwarding misbehavior from the normal channel losses. The CAD algorithm is based on two strategies, channel estimation and traffic monitoring. If the monitored loss rate at certain hops exceeds the estimated normal loss rate, those nodes involved will be identified as attackers. Moreover, we carry out analytical studies to determine the optimal detection thresholds that minimize the summation of false alarm and missed detection probabilities. We also compare our CAD approach with some existing solutions, through extensive computer simulations, to demonstrate the efficiency of discriminating selective forwarding attacks from normal channel losses.**

***Index Terms*—Wireless mesh network, selective forwarding attack, gray hole attack, channel aware detection, optimal detection threshold.**

I. INTRODUCTION

WIRELESS mesh networks (WMNs) [1] are emerging as a popular choice for Internet service providers (ISPs) to provision broadband wireless access in the future. The WMNs are expected to incorporate the attributes of self-organization, self-healing, and self-configuration for high reliability and scalability. In spite of the multiple aspects of advantages, the WMNs lack security guarantees due to its open medium, distributed architecture, and dynamic topology [1]-[5].

The WMN is a multi-hop network, which relies on mesh routers to forward the packets to the destination. It is clear that successful collaboration among routers is the foundation for a strong and reliable network. Cryptography solutions can be used to protect the mesh routers from most of the routing protocol attacks—selective forwarding, blackhole, sinkhole, and wormhole attacks [2], [3], [5]-[7]. Nevertheless, if the routers are compromised, the attacker will gain access to the public/private keys of the compromised routers and then break through the cryptographic system. Therefore, to achieve complete security in a network, it is preferred to use cryptographic solutions as a first line of defense and non-cryptographic solutions as a second line of defense.

In this paper, we investigate a special case of denial of service (DoS) attack, known as *selective forwarding* attack or *gray hole* attack. With such an attack, the misbehaving router accepts the packet for transmission but refuses to forward certain packets by simply dropping them. If an attacker drops all the packets, the attack is then called black hole which has been well studied [2]-[5]. To launch a selective forwarding attack, an attacker may compromise or hijack the mesh router that belongs to the network, known as *internal attacks*; or attack the network from outside, known as *external attacks* [8], [9], [10]. To prevent external attacks, routers may employ an authentication mechanism, e.g., TESLA [7], to avoid the attacks from unauthorized routers. However, internal attacks may pose severe threats and are difficult to defend by cryptographic measures alone. We thus focus on a non-cryptographic approach to counteract the dropping misbehavior launched by internal attackers.

While most of the existing studies [2], [3], [4], [11] on selective forwarding attacks focus on attack detection under the assumption of an error-free wireless channel, we consider a more practical and challenging scenario that packet dropping may be due to gray hole attacks, or *normal loss events* such as medium access collision or bad channel quality. Specifically, we develop a *channel aware detection* (CAD) algorithm that can effectively identify the selective forwarding attackers by filtering out the normal channel losses.

The CAD approach is based on two procedures, *channel estimation* and *traffic monitoring*. The procedure of channel estimation is to estimate the *normal loss rate* due to bad channel quality or medium access collision. The procedure of traffic monitoring is to monitor the *actual loss rate*; if the monitored loss rate at certain hops exceed the estimated loss rate, those nodes involved will be identified as attackers. Specifically, the traffic monitoring procedure at each intermediary node[1] along a path monitors the behaviors of both its upstream and downstream neighbors, termed as *upstream monitoring* and *downstream monitoring*, respectively. The

Manuscript received May 13, 2009; revised September 15, 2009 and January 3, 2010; accepted February 28, 2010. The associate editor coordinating the review of this paper and approving it for publication was I. Habib.

The authors are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA 60616 (e-mail: {dmanikan, cheng, anjali}@iit.edu).

Y. Cheng's work was supported in part by NSF grant CNS-0832093.

Digital Object Identifier 10.1109/TWC.2010.090700

[1] The terms nodes and routers are used interchangeably in this paper for convenience.

1536-1276/10$25.00 © 2010 IEEE

channel estimation procedure at each node correspondingly sets an *upstream detection threshold* and *downstream detection threshold*. Each node judges the behavior of its neighbors by comparing the upstream/downstream observations against the detection thresholds to identify the misbehaving nodes. In particular, the thresholds will be dynamically adjusted with the normal loss rates to maintain the detection accuracy when network status changes. In summary, this paper has four-fold contributions:

- The channel estimation is integrated with traffic monitoring to achieve channel-aware detection of gray hole attack, which can effectively identifies selective forwarding misbehavior hidden in the normal loss events due to bad channel quality or medium access collisions.
- In CAD, upstream and downstream traffic monitoring are combined to achieve a versatile detection method. In addition to gray hole attack, the CAD can also detect *limited transmit-power* attack [2], *on-off* attack [12]-[13] and *bad mouthing* attack [14].
- We carry out analytical studies of the false alarm and missed detection probabilities for the CAD scheme. Based on the analytical model, the optimal upstream/downstream detection thresholds can be computed to minimize the summation of false alarm and missed detection probabilities. The thresholds are dynamically adjusted with the channel status to maintain the efficiency of CAD under varying network conditions.
- Extensive computer simulation results are presented to demonstrate the efficiency of CAD, in comparison with some existing methods.

The rest of the paper is organized as follows. Section II reviews more related work. Section III describes the system model and basic assumptions. Section IV presents the proposed CAD algorithm. Section V discusses how to estimate the normal loss rates due to channel quality and collisions. Section VI computes the optimal detection threshold to minimize the sum of false alarm and missed detection probabilities. Section VII presents the simulation results to demonstrate the performance of CAD. Section VIII gives the conclusion remarks.

II. Related Work

In the last few years, several secure routing protocols resilient to external attacks, such as SAODV [9], SEAD [10], ARAN [15] and Ariadne [16], were proposed. However, none of these protocols are capable in defending against internal attacks. Wireless specific attacks such as *rushing* attacks, *wormhole* attacks were recently identified and studied. These attacks can form a serious threat, because once launched the attacker can easily inject bogus packets, eavesdrop on communication or *selectively drop the data packets*. RAP [17] prevents the rushing attack by waiting for up to m ROUTE REQUEST packets and then randomly selecting one to transmit the data packets, rather than always selecting the first ROUTE REQUEST packet for forwarding. However, RAP has significant network overhead and is ineffective if the adversary has compromised m or more nodes. Packet leashes [18] and LiteWorp [8] are two well-known techniques to defend against wormhole attacks. The former one restricts the maximum transmission distance of the packet by using either a clock synchronization or location information. The latter one uses guard nodes to overhear the communications between the neighboring nodes and exploits the directional antenna techniques [19].

Most of the prior works related to selective forwarding attacks were studied in the area of ad hoc and sensor networks. Karlof *et al.*[3] first proposed selective forwarding attacks and suggested that multipath forwarding can be used to counter these attacks in sensor networks. However, the algorithm fails to suggest a method to detect and isolate the attackers from the network. In [4], the authors propose a scheme that randomly selects part of the intermediate nodes along a forwarding path as checkpoint nodes which are responsible for generating acknowledgments for each packet received. If suspicious behavior is detected, it will generate an alarm packet and deliver it to source node. Some of the key disadvantages of the scheme are: (1) The algorithm suffers from high overhead because for each received packet the intermediate nodes need to send an acknowledgment back to the source node; (2) The algorithm assumes that the channel is perfect and any packet loss is due to the presence of malicious nodes. In [5], we present a detection algorithm based on the end-to-end path throughput (path delivery rate) to detect the selective forwarding attacks in mesh networks. The algorithm can trace back to one-hop neighborhood of the attacker but cannot pin-point the attacker. The algorithm also fails to identify the attacker in the presence of false reports. In [2], a Watchdog technique is proposed, where a node monitors its neighbors to determine whether they forward the packet to intended destination. Consequently, if a node does not overhear a neighbor forwarding more than a threshold number of packets, it concludes that the neighbor is adversarial. The scheme fails to detect the attacker in presence of limited transmit power attack, selective dropping and bad mouthing attack (see section IV for the description of attacks) which can be addressed by the detection approach proposed in this paper. In [6], the authors propose a selective forwarding detection scheme Byzantine-resilient multicast protocol (BSMR) for multicast routing protocols. Specifically, in BSMR nodes determine the reliability of links (or abnormal losses in links) by comparing the perceived data rate with the one advertised by the source node on the basis of detection thresholds δ, Δ. If the perceived data rate falls below the rate indicated by the source node by more than a threshold, the node that is a direct descendant of an adversarial node updates its weight list and initiate a new route discovery process by including the weight lists of the links in the route requests (a higher weight list implies low reliability). Nonetheless, BSMR relies on static detection thresholds, which are independent of channel quality and medium access collisions and we highlight this drawback through our simulations which shows that the BSMR fails to detect the attackers in a number of malicious dropping rates. In [21], the authors present a game theoretic analysis of securing cooperative ad hoc networks against insider attacks in the presence of noise and imperfect monitoring. Though the analysis considers the normal channel errors, it assumes that the normal loss probabilities at different nodes are the same and remain unchanged. Such an assumption may lead to inaccurate analysis in practice, because different nodes may experience different loss probabilities at different times, depending on the local channel quality, number of interfering nodes in a neighborhood, and traffic dynamics at each node.

The CAD approach proposed in this paper departs from the previous solutions in three aspects. (1) CAD considers a practical scenario where a packet loss may be due to bad channel quality, medium access collisions, or purposeful packet dropping; and propose a method to discriminate attacks from those normal loss events. (2) CAD utilizes both upstream and downstream traffic monitoring for enhanced performance; the Watchdog approach [2] relies on downstream monitoring alone. (3) While the existing studies have requirements such as directional antennas [19], clock synchronization [18], and guard nodes [8], CAD is a lightweight algorithm for multi-hop networks. We initially introduced the concept of channel-aware detection of gray hole attacks in [24], the CAD approach is elaborated in this paper with implementation details, analytical model, performance optimization, and computer simulations.

III. System Model and Assumptions

In this section, we present the network model and threat model considered, and also indicate the assumptions for the CAD design. For convenience, the main mathematical notations used in this paper is summarized in Table I.

TABLE I
Summary of main Notations

Notations	Descriptions
S	Source node of a path
D	Destination node of a path
v_i	An intermediary node along a path
W_s	The interval (in terms of number of packets) between two consecutive PROBE packets
$n_{v_i-1}^{v_i}$	Number of packets received by node v_i from v_{i-1}
P_{dt}	Probability of distrust
τ_d (τ_u)	downstream (upstream) detection threshold
τ_d^* (τ_u^*)	Optimal downstream (upstream) threshold
$O_{v_i}^{v_{i+1}}$	opinion of node v_i to the upstream node v_{i+1}
$Q_{v_{i+1}}^{v_i}$	opinion of node v_{i+1} to the upstream node v_i
κ	Number of retransmission attempts by the source node when a PROBE packet is lost
P_e	Observed actual loss rate over a link
p_o	Probability of Collision
p_e	Wireless loss probability due to bad channel quality
p_r	Estimated normal loss rate over a link due to bad channel or collision
p_a	Selective dropping rate due to the attacker
p_l	Aggregate loss rate over a link under attack
$\widehat{p_r}$	Estimated loss rate with protection margin
R_b	Channel busyness ratio
P_G	Probability of loss when the wireless channel is in *good* state
P_B	Probability of loss when the wireless channel is in *bad* state
π_g (π_b)	Steady state probability of the Markovian wireless channel in good (bad) state
P_{FA}	Probability of False Alarm
P_{MD}	Probability of Missed Detection
h	Number of hops in a path
L_a^S	Length of the message appended by Source node S in PROBE packet
L_a^i	Length of the message appeded by each hop i in the path
L_{ack}^i	Length of a PROBE ACK
L_q	Length of a “querying” packet
L_{ack}^M	Length of a link-layer ACK
L_d	Length of a normal data packet

A. Network Model

We consider a single channel multi-hop infrastructure mesh network [1]. Infrastructure WMNs are commonly used in community and neighborhood networks. In this type of network, mesh nodes are statically deployed, e.g., on the roof of houses in a neighborhood, and communicate with one another to form a multi-hop wireless backbone. One or more mesh nodes are connected to the Internet and serve as gateways to provide Internet connectivity for the entire mesh network. The mesh nodes can aggregate traffic from its end clients and forward the traffic to and from the Internet.

B. Threat Model

In a wireless mesh network, we consider that the adversary may compromise certain mesh nodes through physical capture or software bugs, thus gaining full control of them. Once captured, the attacker gains access to all stored information, including public, private keys and reprogram them to behave in a malicious manner. In a multi-hop network like ad hoc, sensor, and mesh networks, effective routing algorithms are required to find high throughput path between source and destination. All the distributed routing protocols for multi-hop networks [20] assume that all the nodes are collaborative and behave normally. However, due to the open medium, the normal routing behavior can be attacked easily. A typical threat model to the distributed routing is that the attackers broadcast misleading routing messages.

We use an example to illustrate the attack. For a path, $v_1, v_2, \ldots, v_n$, between the source S and destination D, we assume that node v_2 is a compromised router that attracts network traffic by advertising itself as having the high quality path to the destination and then performs selective forwarding attacks on the data passing through it. Suppose that source S receives data from mesh client to forward to the destination D. On receiving the request for data transmission, it will check if it has an entry for node D in the routing table. If no entry is found, it will broadcasts a Route Request for that destination. Node v_2 claims that it has a better path to destination whenever it received Route Request packets and sends the reply back to source. The destination or other intermediate routers may send the reply if it has a fresh route to destination. If node S receives the reply from a normal behaving node before it gets the reply from the attacker, everything works well. However, the Route Reply from v_2 can reach node S first for any of the following two reasons. (a) A malicious router may be near to the source router; (b) A malicious router does not have to check the routing table when sending false routing information. As a result, node S will think that the Route Discovery Phase is complete, ignore all other Route Reply packets and forwards data packets to D via v_2. Node v_2 will form a selective forwarding attack in the network by selectively dropping the subset of the packets it receives. If v_2 drops all the packets, the attack is known as *black hole*.

In this work, we focus on developing an algorithm that defends against single and multiple selective dropping attackers in WMN. Moreover, the developed CAD algorithm has the side benefit to deal with some other attacks including *limited transmit power* attack [2], *on-off* attack [12] and *bad mouthing* attack [14].

C. Assumptions

Given a WMN statically deployed, we assume that an accurate channel model for each link could be established by measurement. We also assume that mesh routers have no energy constraints and each mesh node is assigned a pair of public/private keys by a trusted certification authority (CA) [31]. Since the main objective of this work is to provide an insight on the detection of gray-hole attackers in mesh networks, we ignore the details of key distribution in a wireless mesh network, which are available in the literature [31], [32]. For the message authentication used in the CAD design, we adopt the elliptic curve digital signature algorithm (ECDSA) with a 224-bit key (equivalent to the security level of RSA with a 2048-bit key) [33]. Note that if a node is compromised, the attacker will gain access to the stored keys in the victim node. Hence, we argue that in addition to cryptographic solutions, non-cryptographic solutions should be employed to achieve complete security in a network. The CAD algorithm design further takes the following assumptions:

- We assume that the majority of mesh routers are normal-behaving. The mesh network is strongly connected; given an attacker, there exists with high probability one or more paths with normal-behaving source and destination nodes passing the attacker. Thus, in CAD design, we always consider a path with trustworthy source and destination nodes. It is also assumed that the communication on every link between the mesh nodes is *bidirectional*.
- We assume that the gateway is compromise resilient in this work. Techniques addressing attacks to gateways have been discussed in the literature [25].
- We consider that each mesh node has a buffer of infinite size, and a packet can be dropped due to bad channel quality, medium access collision, or presence of an attacker.
- Since there may exist multiple routes from a source to a destination, a source could receive several route replies from a destination. We need the source node to cache these routes to mitigate the overhead incurred during new route discovery process.
- As mentioned before, since the main focus of this work is to address the selective forwarding attacks, we assume the system is free of the general attacks such as sybil attacks, collision (or jamming) attacks or node replication attacks. Techniques discussed in [26], [27], [28], [29] propose solutions to defend against these attacks.
- In this paper, we only deal with the scenarios that the attacking mesh nodes act alone, and the problem of colluding nodes is out of the scope of this paper.

IV. The Channel-Aware Detection Algorithm

In this section, we present the design and operation of the channel-aware detection algorithm in detail. For a node v_i in a forwarding path, we refer to v_{i-1} and v_{i+1} as its *upstream* (previous-hop) and *downstream* (next-hop) nodes, respectively.

A. Methodology

The basic principle of CAD is as follows. Each intermediary node along a given path v_i implements both the *downstream traffic monitoring*, that is, observing the behavior of its downstream node v_{i+1} to determine whether the node is misbehaving by dropping or tampering the data packets, and the *upstream traffic monitoring*, that is, observing the behavior of its upstream node v_{i-1} by measuring the loss rate over the link between v_{i-1} and v_i, denoted as $e_{i-1,i}$. These observations by node v_i are then compared against the upstream/downstream detection thresholds to detect misbehaviors. The main advantages of the algorithm stem from two facts: (i) Each node's behavior in the path is observed by its upstream and downstream neighbors; (ii) The thresholds are dynamically adjusted with the normal loss rates to maintain the detection accuracy when network status changes.

B. Design of CAD Algorithm

The essence of CAD is to identify intentional selective dropping from normal channel losses. A normal packet loss can occur due to bad channel quality or medium access collision under the infinite buffer assumption. These two types of loss events are independent and we will discuss how to estimate the normal losses in Section V and how to compute the optimal upstream/downstream detection thresholds in Section VI. In this section, we present the CAD algorithm based on given detection thresholds. An outline flowchart of the CAD algorithm is shown in Fig. 1.

In CAD, each mesh node maintains a history of packet count to measure the link loss rate. When a node receives a packet from the upstream, it updates the packet count history with the corresponding packet sequence number. We denote the number of packets forwarded by source S to destination D as W_s and the number of packets received successfully by the intermediate node v_i from the upstream node v_{i-1} as $n_{v_{i-1}}^{v_i}$, over a time window. When a router forwards a packet to the downstream node, it performs two operations: (i) For each packet relayed to the downstream, it buffers the link layer acknowledgments;[2] (ii) It also overhears downstream traffic and determines whether the node *forwarded* or *tampered*[3] the packet. For instance, when node v_{i-1} forwards a packet to v_i, it maintains the acknowledgment returned by v_i and overhears whether v_i tampered or forwarded the packet. Based on these

[2] These acknowledgments serve as a proof for the successful transmission of the packet to the downstream node. Note that to avoid the fabrication of the acknowledgment packets, the routers can attach a digital signature to the acknowledgment packet.

[3] All nodes in the forwarding path except destination maintains a copy of the recently forwarded packets and compares each overheard packet with the stored one to see if there is a match. We believe that encryption of payload or header of packets are not performed independently at each link (which can be an expensive operation); and hence if a mismatch occurs, the upstream node can easily identify the misbehaving downstream node.

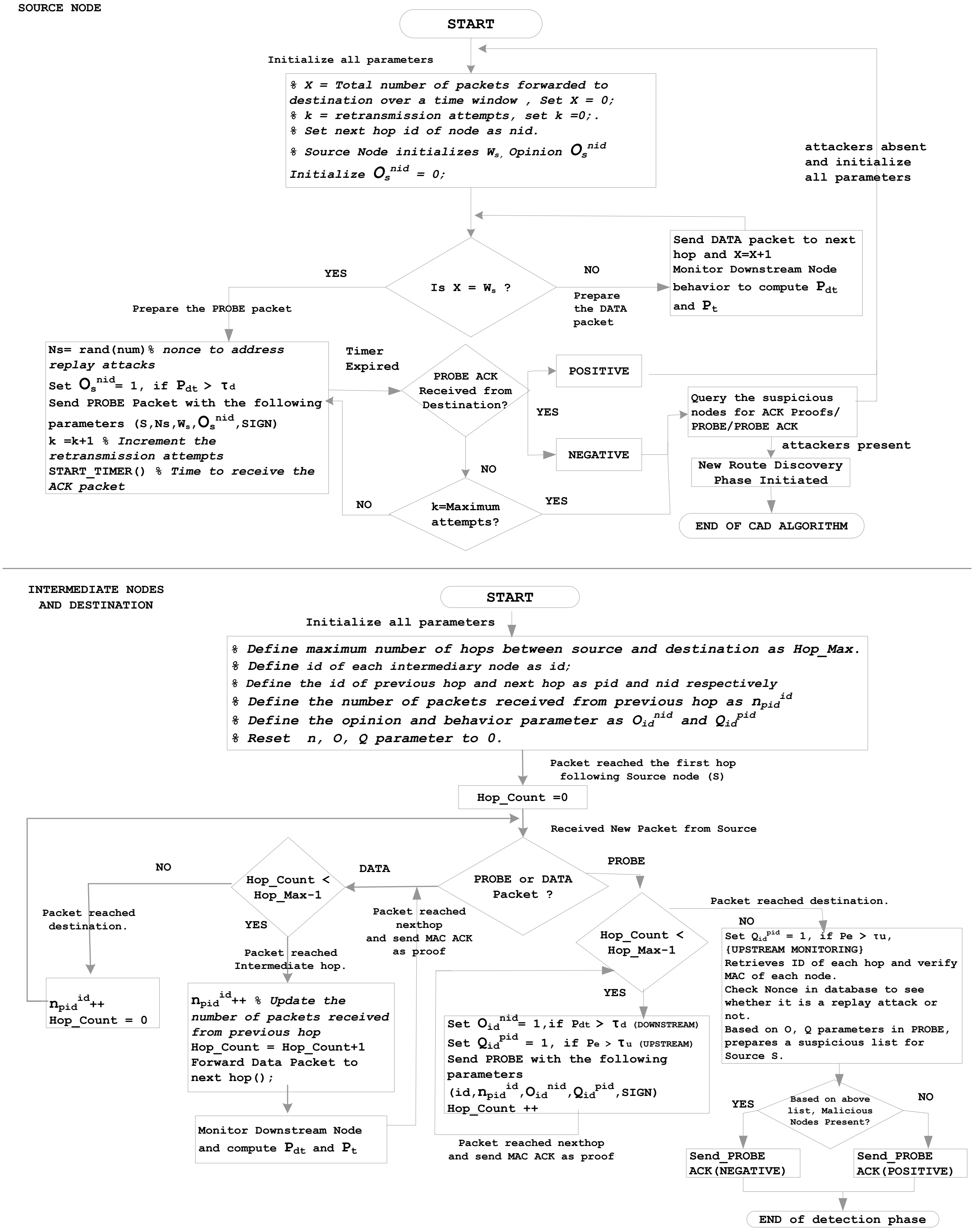
SOURCE NODE
START
Initialize all parameters
% X = Total number of packets forwarded to destination over a time window , Set X = 0;
% k = retransmission attempts, set k =0;.
% Set next hop id of node as nid.
% Source Node initializes Ws, Opinion Os^nid
Initialize Os^nid = 0;
attackers absent and initialize all parameters
Send DATA packet to next hop and X=X+1
Monitor Downstream Node behavior to compute Pdt and Pt
YES
NO
Is X = Ws ?
Prepare the DATA packet
Prepare the PROBE packet
Ns= rand(num)% nonce to address replay attacks
Set Os^nid= 1, if Pdt > τd
Send PROBE Packet with the following parameters (S,Ns,Ws,Os^nid,SIGN)
k =k+1 % Increment the retransmission attempts
START_TIMER() % Time to receive the ACK packet
Timer Expired
PROBE ACK Received from Destination?
POSITIVE
YES
NEGATIVE
Query the suspicious nodes for ACK Proofs/ PROBE/PROBE ACK
attackers present
New Route Discovery Phase Initiated
END OF CAD ALGORITHM
NO
NO
k=Maximum attempts?
YES
INTERMEDIATE NODES AND DESTINATION
START
Initialize all parameters
% Define maximum number of hops between source and destination as Hop_Max.
% Define id of each intermediary node as id;
% Define the id of previous hop and next hop as pid and nid respectively
% Define the number of packets received from previous hop as npid^id
% Define the opinion and behavior parameter as Oid^nid and Qid^pid
% Reset n, O, Q parameter to 0.
Packet reached the first hop following Source node (S)
Hop_Count =0
Received New Packet from Source
NO
Hop_Count < Hop_Max-1
DATA
PROBE or DATA Packet ?
PROBE
Packet reached destination.
Packet reached destination.
YES
Packet reached nexthop and send MAC ACK as proof
Packet reached Intermediate hop.
Hop_Count < Hop_Max-1
NO
npid^id++
Hop_Count = 0
npid^id++ % Update the number of packets received from previous hop
Hop_Count = Hop_Count+1
Forward Data Packet to next hop();
Monitor Downstream Node and compute Pdt and Pt
YES
Set Oid^nid= 1,if Pdt > τd (DOWNSTREAM)
Set Qid^pid = 1, if Pe > τu (UPSTREAM)
Send PROBE with the following parameters
(id,npid^id,Oid^nid,Qid^pid,SIGN)
Hop_Count ++
Packet reached nexthop and send MAC ACK as proof
Set Qid^pid = 1, if Pe > τu, {UPSTREAM MONITORING}
Retrieves ID of each hop and verify MAC of each node.
Check Nonce in database to see whether it is a replay attack or not.
Based on O, Q parameters in PROBE, prepares a suspicious list for Source S.
YES
Based on above list, Malicious Nodes Present?
NO
Send_PROBE ACK(NEGATIVE)
Send_PROBE ACK(POSITIVE)
END of detection phase

Fig. 1. An outline of CAD algorithm.

observations, each node maintains a *probability of distrust* P_{dt} (or *probability of trust* $P_t = 1 - P_{dt}$) for its downstream node. The probability of distrust maintained by a node regarding its downstream node is computed as follows:

$$P_{dt} = \frac{n_t + n_d}{n_f} \tag{1}$$

where n_t and n_d denote the number of packets tampered and dropped by the downstream node, respectively; and n_f denotes the total number of packets delivered to the downstream node.

We introduce two types of packets known as the PROBE packet and PROBE ACK packet for the detection of malicious routers [22]. The source S sends a PROBE packet after every W_s data packets. A smaller value of W_s represents more frequent probing and thus more timely detection of the attack; the cost is a higher bandwidth overhead due to the probing traffic. On receiving the PROBE, each node in the path marks the PROBE packet with its traffic monitoring information parameters. This technique is known as *packet marking* [30]. Specifically, for each PROBE packet send to destination, the source node attaches the number of packets transmitted to the particular destination, i.e., W_s. When the PROBE packet is passed along the path, each node v_i attaches the number of packets it received successfully from its upstream node v_{i-1} ($n^{vi}_{v_{i-1}}$) and also a mark of its *opinion to the downstream* node v_{i+1}, denoted as $O^{v_{i+1}}_{v_i}$. Let τ_d denote the *downstream detection threshold*; such an opinion is set by comparing the distrust probability P_{dt} to the threshold τ_d as

$$O^{v_{i+1}}_{v_i} = \begin{cases} 1 & \text{if } P_{dt} > \tau_d; \ \texttt{[misbehaving]} \\ 0 & \text{if } P_{dt} \le \tau_d; \ \texttt{[normal]}. \end{cases} \tag{2}$$

In addition, the downstream node v_{i+1} will attach its *opinion to the upstream* node v_i, denoted as $Q^{v_i}_{v_{i+1}}$. To obtain the opinion to the upstream, the node v_{i+1} needs to measure the loss rate over the link $e_{i,i+1}$ as

$$\begin{aligned} P_e^{i,i+1} &= 1 - \frac{\text{number of packets received by } v_{i+1} \text{ from } v_i}{\text{number of packets received by } v_i \text{ from } v_{i-1}} \\ &= 1 - \frac{n^{v_{i+1}}_{v_i}}{n^{v_i}_{v_{i-1}}}. \end{aligned} \tag{3}$$

where, node v_{i+1} computes $P_e^{i,i+1}$ based on the information, $n^{v_i}_{v_{i-1}}$, that each intermediary node (upstream node of v_{i+1} i.e., v_i) attaches in the PROBE packet. The opinion to the upstream is set by comparing the measured link loss rate to the *upstream detection threshold* τ_u as

$$Q^{v_i}_{v_{i+1}} = \begin{cases} 1 & \text{if } P_e^{i,i+1} > \tau_u; \ \texttt{[misbehaving]} \\ 0 & \text{if } P_e^{i,i+1} \le \tau_u; \ \texttt{[normal]} \end{cases} \tag{4}$$

As an example illustrating the PROBE packet, we consider a simple 4-hop path S, v_1, v_2, v_3,D. The messages carried by the PROBE packet and observed at each hop are denoted as M_1, M_2, M_3, and M_4, respectively. The information carried by the messages is as follows:

$$\begin{aligned} &S \overset{M_1}{\rightarrow} v_1 : M_1 = S||W_S||\eta_S||O^{v_1}_S||SIGN_S \\ &v_1 \overset{M_2}{\rightarrow} v_2 : M_2 = M_1||v_1||n^{v_1}_S||O^{v_2}_{v_1}, Q^S_{v_1}||SIGN_{v_1} \\ &v_2 \overset{M_3}{\rightarrow} v_3 : M_3 = M_2||v_2||n^{v_2}_{v_1}||O^{v_3}_{v_2}, Q^{v_1}_{v_2}||SIGN_{v_2} \\ &v_3 \overset{M_4}{\rightarrow} D : M_4 = M_3||v_3||n^{v_3}_{v_2}||Q^{v_2}_{v_3}||SIGN_{v_3} \end{aligned}$$

At each node, the message is attached with a 56-bytes ECDSA signature generated with a 28-bytes (224-bits) key. The ECDSA signature can protect the message from being tampered. Moreover, in order to prevent the replay attack, each source node further incorporates a nonce random number η_S to generate the signature for the first message M_1 attached with a PROBE packet, and the corresponding destination node stores the nonce number having been used.

When the destination (or gateway) receives the PROBE message, it first retrieves the ID of the last hop node, say, v_n and uses the corresponding public key to verify $SIGN_{v_n}$. If $SIGN_{v_n}$ is correct, it then retrieves the ID of the upstream node of v_{n-1} and verifies the $SIGN_{v_{n-1}}$. The destination node continues this process until it verifies all the signatures or it finds an incorrect signature. Once all the signatures are verified, the destination node D builds a list of suspicious nodes based on the downstream/upstream opinions, a kind of reputation [23], marked by each node in the forwarding path.

C. Detection of Attacks

With the CAD design, the downstream and upstream opinions regarding a node will be combined to detect attacks. Regarding node v_i, there are four possible combination cases.

Case A: $O^{v_i}_{v_{i-1}} = 1$ **and** $Q^{v_i}_{v_{i+1}} = 1$. This case indicates the *selective forwarding attack* by node v_i. In this case, the upstream node v_{i-1} has overheard that v_i dropped (or tampered) the packets, and obtained a distrust probability larger than the threshold τ_d; the downstream node v_{i+1} has also observed that the link loss rate P_e over the link $e_{i,i+1}$ is greater than the threshold, τ_u. Hence, both the downstream opinion $O^{v_{i+1}}_{v_i}$ and upstream opinion $Q^{v_i}_{v_{i+1}}$ are set to 1.

Case B: $O^{v_i}_{v_{i-1}} = 0$ **and** $Q^{v_i}_{v_{i+1}} = 1$**.** This case can indicate two attacks. (1) *Limited transmit power attack by* v_i: In this attack, node v_i could limit its transmission power such that the signal is strong enough to be overheard by the upstream node v_{i-1} but too weak to be received by the downstream node v_{i+1}. The Watchdog scheme [2] relies on downstream monitoring only and can not detect such kind of attack, with the downstream opinion $O^{v_i}_{v_{i-1}} = 0$ indicating normal. Nevertheless, with the CAD method the downstream node v_{i+1} by upstream monitoring will observe the high loss rate over link $e_{i,i+1}$ and sets $Q^{v_i}_{v_{i+1}}$ to indicate the abnormal behavior of node v_i. (2) *Bad mouthing attack by node* v_{i+1}*:* In this attack, the downstream node v_{i+1} falsely accuses that the loss rate over link $e_{i,i+1}$ is greater than the threshold τ_u and sets $Q^{v_i}_{v_{i+1}} = 1$. The node v_i, if normally behaved in fact, can use the recorded link-layer acknowledgment returned by node v_{i+1} as evidence to detect such a false accusation attack.

Case C. $O^{v_i}_{v_{i-1}} = 1$ **and** $Q^{v_i}_{v_{i+1}} = 0$**.** This case indicates a *phony marking attack by node* v_i. In this attack, node v_i lies about the number of packets it received from the upstream node to cheat the downstream node. For example, suppose that v_{i-1} forwarded 5 packets to v_i and v_i (being a malicious node) dropped 2 packets. The packet count at intermediate nodes v_i and v_{i+1} will then be $n^{v_i}_{v_{i-1}}$ = 5 and $n^{v_{i+1}}_{v_i}$ = 3. However, if v_i marks the PROBE packet with a counterfeited value of 3 for $n^{v_i}_{v_{i-1}}$, node v_{i+1} will observe the normal behavior and assigns $Q^{v_i}_{v_{i+1}} = 0$. Since the CAD method incorporate downstream monitoring, the upstream node v_{i-1} can find the misbehavior of node v_i. The link layer acknowledgments received by v_{i-1} from v_i can serve as evidence, when v_i selectively dropped the packets and counterfeit the marking $n_{v_{i-1}}^{v_i}$.

Case D. $O_{v_{i-1}}^{v_i} = 0$ **and** $Q_{v_{i+1}}^{v_i} = 0$**.** This case indicates that node v_i behaves normally.

In addition to the attacks mentioned above, the CAD method can also effectively defend against the *on-off* attack [12]. In this attack, the malicious node behaves well and badly alternatively, hoping that it can remain unidentified while causing damage to the network. With CAD, the source periodically sends PROBE packets with a period of W_s. As long as the attack in an on-duration causes abnormal behavior in an observation window of length W_s, the attack could be detected. Moreover, it is not difficult to see that the CAD approach can effectively detect multiple independent attackers along a path. An interesting issue we are going to consider in our future work is that multiple attackers can collude with each other, which is a further challenging scenario.

D. Actions with PROBE ACK

The CAD design requires the destination node D to send a PROBE ACK message for every PROBE packet received from the source node. The PROBE ACK message is also secured with digital signature, similar to a PROBE message. Further, the PROBE ACK message also includes the nonce random number sent by the source node to ensure that attacker cannot replay old communications. There are three possible cases regarding the PROBE ACK; in different cases, the source node S will take different actions correspondingly.

Negative PROBE ACK. If the PROBE ACK message returned to source S includes a list of suspicious router(s) based on CAD, we term that the destination node return a *negative* PROBE ACK to the source node. When S receives the negative PROBE ACK message, it will query the suspicious routers for the proof of link layer acknowledgments. If a router is detected as misbehaving, source may use another path in route cache to forward the remaining data packets and informs the network to evict the misbehaving node. The details of node eviction is out of the scope of paper.

Positive PROBE ACK. If no suspicious nodes are listed in the PROBE ACK message, we term that the destination node return a *positive* PROBE ACK to the source node. The source node just continues the normal data transmission upon the positive PROBE ACK.

PROBE ACK Timeout. The source may not receive a reply from the destination within a timeout interval, due to two possible reasons: (a) The PROBE packet is dropped by the malicious router or due to normal loss events, and hence D will not be triggered to return any PROBE ACK; (b) The PROBE ACK packet returned by D is dropped by the malicious router or due to normal loss events along the reverse path. To tackle these situations, we propose two actions: (i) Every router buffers the PROBE and PROBE ACK packets. (ii) The source node retransmits the PROBE for κ times. After κ attempts, if S still can not receive a reply from D, it will initiate a hop by hop query for the PROBE and PROBE ACK packets and implement attack detection based on CAD. For example, if it turns out that node v_i did not receive the PROBE packet, which implies that node v_{i-1} did not forward the packet, the upstream and downstream neighbors of v_{i-1} can then be queried for their opinion parameters. If misbehavior is detected, the source node will perform the same operation as listed in the negative ACK case.

V. Estimation of Normal Losses

In this section, we discuss how to estimate the normal loss rate due to channel quality or medium access collision.

A. Loss due to Channel Quality

We estimate the loss due to wireless channel quality, termed as *wireless loss probability*, by modeling the underlying time varying wireless channel as a two-state Markov model [34], [35]. The two-state Markov model has two states, G and B, which represents the *good* and *bad* states respectively. In the *good* states, losses occur with a probability of P_G, while in the *bad* state they happen with a probability of P_B; $P_G < P_B$. The transition probabilities of the model are defined by P_{bg} from transition from state B to G and P_{gb} vise versa. The wireless loss probability, p_e, of the Markov channel is given as

$$p_e = P_G \pi_G + P_B \pi_B \tag{5}$$

where π_G and π_B are the steady state probabilities and can be computed as $\pi_G = \frac{P_{bg}}{P_{bg}+P_{gb}}$ and $\pi_B = \frac{P_{gb}}{P_{bg}+P_{gb}}$, respectively. Since a wireless mesh network is normally deployed statically for long time, we assume that the channel parameters P_{bg}, P_{gb}, P_G, and P_B can be accurately estimated by observing historical data. The technical details of how to estimate the channel parameters can be referred to [34].

B. Loss due to MAC Layer Collisions

We consider the wireless mesh network is based on the popular IEEE 802.11 distributed coordinate function (DCF) MAC protocol [36], which takes a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. In the MAC layer, a packet may be lost due to MAC layer collisions, when multiple transmissions happen in the same slot. The *packet collision probability* for a given transmission, denoted as p_o, can be estimated by measuring the channel busyness ratio, denoted as R_b. The *channel busyness ratio* is defined as the proportion of time that the channel is in the status of successful transmission or collision. It is very convenient for a node to monitor the channel busyness ratio as a CSMA-based MAC protocol works on physical and virtual carrier sensing mechanisms. For a given observation window, the channel idling time can be easily computed by tracing the backoff counter values, the leftover part within the observation window is the channel busy time.

We adopt the widely-used virtual slot model [37], [38] to analyze the MAC layer loss. Consider that the total number of nodes competing the channel is n. Let p_t denote the probability that a node transmits in a certain time slot. For the MAC channel at steady state, the probabilities for observing an idle, successful, and colliding slot (denoted as p_i, p_s, and p_c, respectively) can be expressed as

$$\begin{cases} p_i &= (1-p_t)^n \\ p_s &= np_t(1-p_t)^{n-1} \\ p_c &= 1-p_i-p_s. \end{cases} \tag{6}$$

The channel busyness ratio can then be computed as

$$R_b = 1 - \frac{p_i\sigma}{p_i\sigma + p_sT_s + p_cT_c} \tag{7}$$

where σ, T_s, and T_c denote the idle slot length, the duration of a successful transmission, and the duration of a collision, respectively, which can be determined from the 802.11 standard [37].

The packet collision probability p_o is the probability that one node encounters collisions when it transmits, which is linked to the probability p_t as

$$p_o = 1-(1-p_t)^{n-1}. \tag{8}$$

Using the relationship of (8) in (6), we can express p_i, p_s, and p_c as functions of p_o. Further applying $p_i(p_o)$, $p_s(p_o)$, and $p_c(p_o)$ into (7), the channel busyness ratio R_b can then be written as a function of p_o. If the value of R_b is obtained by channel monitoring, the packet collision probability can then be computed based on its relationship to R_b. Note that in order to estimate the packet collision probability p_o, it is also required to know the number of nodes competing channel with or the number of interfering node to a tagged node (i.e., n-1). Since the wireless mesh network has a static topology, the number of interfering node to each node can be obtained based on the network topology and the interference range of the wireless network card.

C. Normal Loss Rate

Considering both the effects of bad channel quality and medium access collisions, the aggregate normal loss rate can be expressed as follows:

$$p_r = p_e + p_o - p_ep_o \approx p_e + p_o$$

It is worth noting that the MAC layer may incorporate retransmission schemes [37], [38] to improve the successful delivery rate of a packet. In our context, we focus on the loss rate regarding each transmission. We would like to emphasize that the estimated normal loss probability p_r has a local meaning, which is computed by each node locally by monitoring the physical-layer channel quality and MAC-layer channel collision.

VI. Configuration of Optimal Thresholds

In this section, we discuss how to compute the optimal upstream/downstream thresholds (τ_u^*/τ_d^*) that minimize the sum of false alarm and missed detection probabilities in upstream/downstream monitoring.

A. Probability of False Alarm

A false alarm occurs when the detection scheme gives an alarm but no threat exists in fact. In CAD, attack detection is based on the combination of downstream and upstream monitoring. The downstream/upstream monitoring opinions are configured by comparing the monitored loss rates with the downstream/upstream detection thresholds. Due to the randomness nature, even without selective forwarding attack, a burst of normal loss events in certain situations may lead to the false alarm.

Downstream Monitoring False Alarm. With CAD, along a given path an upstream node (say, v_{i-1}) overhears its downstream's transmission to determine whether the node is behaving properly or not. However, the node v_{i-1} has no way to detect whether a loss is due to attack or normal events. Any loss event will increase the distrust probability. Suppose that v_{i-1} successfully forwarded N packets to its downstream node v_i. When there is no attack, the downstream monitoring threshold τ_d allows $N\tau_d$ normal loss events in the N packets without incurring a false alarm. Assume that the normal loss events are independent. The *downstream monitoring false alarm probability*, denoted as P_{FA}^d, can be computed as:

$$\begin{aligned} P_{FA}^d &= \sum_{i=N\tau_d+1}^{N} \binom{N}{i} p_r^i(1-p_r)^{(N-i)} \\ &= 1 - \sum_{i=0}^{i=N\tau_d} \binom{N}{i} p_r^i(1-p_r)^{(N-i)} \\ &\approx 1 - \frac{1}{\sqrt{2\pi}} \int_{\frac{0-Np_r-1/2}{\sqrt{Np_r(1-p_r)}}}^{\frac{N\tau_d-Np_r+1/2}{\sqrt{Np_r(1-p_r)}}} e^{\frac{-y^2}{2}}dy. \end{aligned} \tag{9}$$

The third step of (9) is due to the fact that the binomial distribution can be well approximated by the Gaussian distribution, when N is large [39].

Upstream Monitoring False Alarm. With CAD, a downstream node (say, v_{i+1}) also monitors the behavior of the upstream node, i.e., measuring the loss rate over the link $e_{i,i+1}$. The upstream monitoring face the same problem as downstream monitoring that the intentional dropping and normal loss rates can not be discriminated. Suppose that node v_{i+1} knows from the PROBE packet that $n_{v_{i-1}}^{v_i} = N'$. When there is no attack, the upstream monitoring threshold τ_u allows $N'\tau_u$ normal loss events in the N' packets without incurring a false alarm. Assuming independent loss events, the *upstream monitoring false alarm probability*, denoted as P_{FA}^u, can be computed as

$$\begin{aligned} P_{FA}^u &= \sum_{i=N'\tau_u+1}^{N'} \binom{N'}{i} p_r^i(1-p_r)^{(N'-i)} \\ &= 1 - \sum_{i=0}^{i=N'\tau_u} \binom{N'}{i} p_r^i(1-p_r)^{(N'-i)} \\ &\approx 1 - \frac{1}{\sqrt{2\pi}} \int_{\frac{0-N'p_r-1/2}{\sqrt{N'p_r(1-p_r)}}}^{\frac{N'\tau_u-N'p_r+1/2}{\sqrt{N'p_r(1-p_r)}}} e^{\frac{-y^2}{2}}dy \end{aligned} \tag{10}$$

with Gaussian approximation applied.

CAD False Alarm. In CAD, a mesh node v_i is listed as suspicious if any of the following conditions occur: (a)

$O^{v_i}_{v_{i-1}} = 1$; (b) $Q^{v_i}_{v_{i+1}} = 1$; and (c) $O^{v_i}_{v_{i-1}} = 1$, $Q^{v_i}_{v_{i+1}} = 1$. Based on the upstream/downstream false alarm probability, the aggregate *CAD false alarm probability* is

$$P_{FA} = P^d_{FA} + P^u_{FA} - P^u_{FA}P^d_{FA} \tag{11}$$

B. Missed Detection Probability

A false clear or missed detection occurs when the detection scheme does not give an alarm but a threat exists. Let p_a denote the selective dropping rate; the aggregate loss rate over a link under attack will be

$$p_l = p_r + p_a. \tag{12}$$

Given the downstream/upstream detection thresholds τ_d/τ_u, the *downstream monitoring missed detection probability* can be computed as

$$\begin{aligned} P^d_{MD} &= \sum_{i=0}^{i=N\tau_d} \binom{N}{i} p_l^i (1-p_l)^{(N-i)} \\ &\approx \frac{1}{\sqrt{2\pi}} \int_{\frac{0-Np_l-1/2}{\sqrt{Np_l(1-p_l)}}}^{\frac{N\tau_d-Np_l+1/2}{\sqrt{Np_l(1-p_l)}}} e^{\frac{-y^2}{2}} dy, \end{aligned} \tag{13}$$

and the *upstream monitoring missed detection probability* can be computed as

$$\begin{aligned} P^u_{MD} &= \sum_{i=0}^{i=N'\tau_u} \binom{N'}{i} p_l^i (1-p_l)^{(N'-i)} \\ &\approx \frac{1}{\sqrt{2\pi}} \int_{\frac{0-N'p_l-1/2}{\sqrt{N'p_l(1-p_l)}}}^{\frac{N'\tau_u-N'p_l+1/2}{\sqrt{N'p_l(1-p_l)}}} e^{\frac{-y^2}{2}} dy. \end{aligned} \tag{14}$$

With CAD, a selective dropping attacker will be detected when either the downstream or the upstream monitoring opinion is set as 1, so a missed detection can happen only when both upstream and downstream monitoring observed loss rates are below the thresholds. Thus, the *CAD missed detection probability* is

$$P_{MD} = P^d_{MD} \times P^u_{MD}. \tag{15}$$

C. Optimal detection thresholds

Based on the above discussions, it can be seen that the false alarm probability decreases with an increasing threshold, while the missed detection probability increases. In fact, the summation of the false alarm and missed detection probabilities is a convex function of the thresholds τ_d and τ_u. Fig. 2 illustrates the convex function versus thresholds τ_d and τ_u. With the convexity, the optimal thresholds, τ^*_d and τ^*_u, can be computed by minimizing the summation of the false alarm and missed detection probabilities according to

$$\frac{d}{d\tau_d}(P_{FA} + P_{MD})|_{\tau_d=\tau^*_d} = 0 \tag{16}$$

$$\frac{d}{d\tau_u}(P_{FA} + P_{MD})|_{\tau_u=\tau^*_u} = 0 \tag{17}$$

With the Gaussian approximation results indicated in (9), (10), (13), and (14), the differentiation in (16) and (17) can be computed by applying Leibniz's integral rule [40] and then to derive the optimal thresholds.

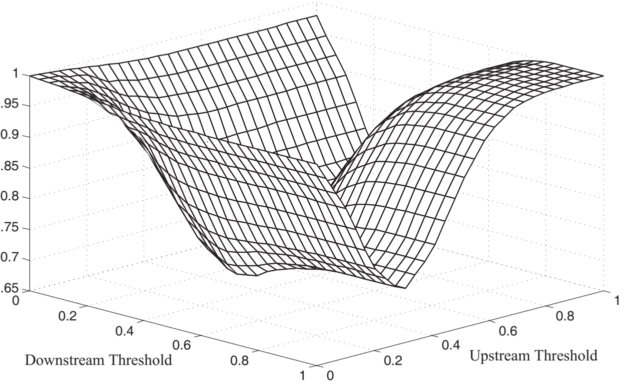

Fig. 2. The sum of false alarm and missed detection probabilities versus the detection thresholds τ_d and τ_u, with $p_r = 0.3$ and $p_a = 10\%$.

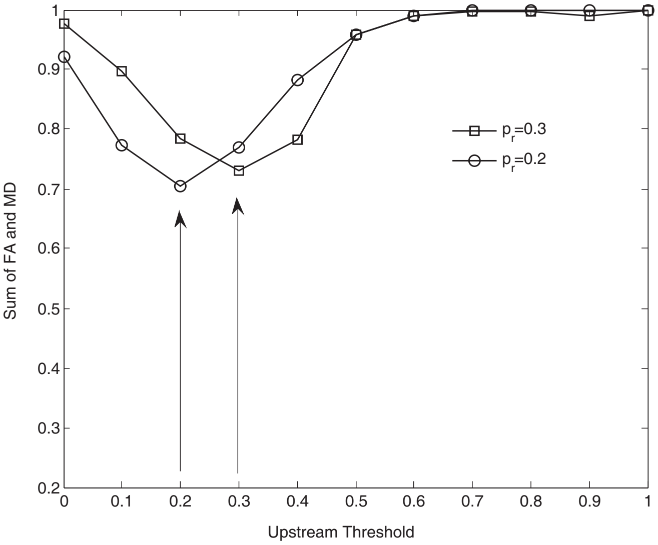

Fig. 3. Illustration of the optimal upstream detection threshold τ^*_u (indicated by the arrows) due to the convexity, with $p_a = 10\%$.

The analysis shows that the optimal detection thresholds equal the normal loss rate p_r, which is intuitively understandable. Fig. 3 illustrates the optimal threshold τ^*_u for cases $p_r = 0.2$ and 0.3, with $p_a = 0.1$. However, just using the normal loss rate as detection thresholds will lead to high false alarm rate in practice. The reason for such high sensitivity is that the monitoring threshold is based on history data; if the estimated normal loss rate is set as detection threshold, any extra fluctuation of the normal loss events in the coming observation window will trigger a false alarm. The false alarm will then trigger unnecessary re-routing option, resulting in wasted resource usage.

A common practice to alleviate the false alarm due to sensitivity is to set the threshold with a margin by applying the exponential weighted moving average (EWMA) [41]. Since we assume that the stochastic parameters of the wireless channel is available by history measurement, the online estimation is of importance to the packet collision probability p_o here. Let $p_o(i)$ denote the estimated packet collision probability in the ith observation window (referring to Section V-B). The mean value and standard deviation of the estimated p_o, denoted as

$\overline{p_o}$ and σ_{p_o}, can be computed by EWMA as

$$\overline{p_o}(i+1) = \alpha\overline{p_o}(i) + (1-\alpha)p_o(i), \quad k = 1, 2, \cdots \tag{18}$$

$$\sigma_{p_o}(i+1) = \beta\sigma_{p_o}(i) + (1-\beta)|\overline{p_o}(i) - p_o(i)|, \quad k = 1, 2, \cdots \tag{19}$$

where $\alpha, \beta \in [0,1]$. We pick the values as $\alpha = 1/8$ and $\beta = 1/4$ according to those used in transmission control protocol (TCP) for estimating the round-trip time (RTT) [41]. Regarding the wireless loss probability over the two-state Markov channel, we can have

$$\overline{p_e} = P_B\pi_B + P_G\pi_G \tag{20}$$

$$\sigma_{p_e} = |P_B - \overline{p_e}|\pi_B + |P_G - \overline{p_e}|\pi_G. \tag{21}$$

For attack detection at each observation window, the estimated normal loss rate $\hat{p_r}$ will then be set with a protection margin as

$$\hat{p_r} = (\overline{p_o} + \overline{p_e}) + k(\sigma_{p_o} + \sigma_{p_e}) \tag{22}$$

A good heuristic margin is at $k = [3, 4]$ according to the suggestions in [41]. The normal loss rate estimation $\hat{p_r}$ is further used to compute the optimal thresholds to avoid unnecessary false alarms. Note that the protection margin will not impact the missed detection performance much, because a non-trivial selective forwarding attack normally leads to monitored loss rates significantly larger than the detection thresholds. The effect of protection margin will be demonstrated in the performance evaluation part.

VII. Performance Evaluation

A. Simulation Model

We use the Berkeley's Network Simulator NS2 (v2.29) [42] for simulations; the simulator includes wireless extensions made by the CMU Monarch project. The proposed algorithm CAD is incorporated with the AODV [43] implementation of NS2. We would like to emphasize that the CAD implementation is independent of the underlying routing protocols, although the AODV protocol is adopted in our experiments. Given a path determined by the routing protocol, the CAD messages for "probing" and "querying" routers will be communicated along the path to detect possible attacks; upon a positive detection of adversaries, the CAD will then trigger the underlying routing protocol to activate a new route discovery process.

The simulation area consists of a square grid of 36 mesh routers located in 1000 meter by 1000 meter. We place stationary sources and destinations on the opposite sides of the grid with multiple forwarding paths between them. A random number generator is employed to randomly locate the malicious nodes in the paths of source and destination pairs. Albeit each malicious node can attack independently (i.e., different selective dropping rates), for the simplicity of presentation we assign same dropping probability for all the attackers. Each source node initiates an user datagram protocol (UDP)/constant bit rate (CBR) traffic flow to its intended destination. Simulations were performed for a duration of 200 seconds, and each data point in the result graphs is an average of 10 runs. We use a value of two for the PROBE retransmission attempts ($\kappa = 2$), when a PROBE packet is not received within a timeout interval. We configure the retransmission time of PROBE packet according to the average packet RTT, which is configured as 1000ms in our simulations, as suggested by references [44], [45] and our experiments.

TABLE II
Wireless loss probability over Markovian Channel, $P_G = 0$ and $P_B = 1$

Wireless loss Probability p_e	P_{gb}	P_{bg}
0	0	1
0.1	0.11	0.99
0.12	0.13	0.953
0.18	0.19	0.866
0.2	0.22	0.88

The physical layer and the IEEE 802.11 MAC layer we use are included in the CMU wireless extensions to NS2. We use the default values of IEEE 802.11 (see [37], [38]) in all our simulations. The transmission range of each node is approximately 250m. The two-state Markov model has been used to model the wireless errors. Table II indicates the transition probabilities of the model used in the simulation. For simplicity, we set the probabilities P_G and P_B to 0 and 1 respectively. In practice, the channel models for different links normally take different configurations. Referring to Section IV, it can be seen that the CAD design can seamlessly work with heterogeneous link models. In this section, we assume all the links have the same channel model for the convenience of performance illustration.

We study the performance of CAD for the following cases. First, we examine the adaption of the downstream and upstream thresholds with the dynamic network load. We vary the packet arrival rate of CBR flows to generate different network load. Second, we examine the potential of CAD in identifying the misbehaving nodes by filtering out the normal channel losses. We also investigate the performance of CAD in the absence of any normal losses. Moreover, we study the sensitivity of PROBE packets interval (W_s) on the performance of CAD. Finally, we compare the performance of CAD with existing schemes such as BSMR and Watchdog (WD).

B. Simulation Results

In the simulation results presented in this section, the curves labeled as "MAL" refer to malicious nodes and the curves labeled as "CAD", "BSMR" and "WD" respectively indicate the proposed and existing dropping detection schemes. We define "no detection" as the scenario where no detection algorithm is employed. We evaluate the performance using the following metrics:

(a) Packet Delivery Ratio (PDR): PDR is computed as the percentage of transmitted data packets that are actually received by the destination.

(b) Overhead: The overhead is calculated as the ratio of CAD-related transmissions (including Probe, Probe Ack, and querying for link-layer acknowledgments upon negative Probe Ack) in terms of bytes to data transmissions [2]. Along a path, a packet being forwarded across h hops would be counted as h transmissions. Note that each CAD-related transmission is secured with a ECDSA signature of 56 bytes to protect the messages from being tampered. For convenience of presentation, we compute the average overhead amortized to each node, denoted as V. Consider a typical path with h hopes. Assume there are m attackers uniformly distributed between the source and destination nodes, so the average distance between an attacker and the source node is $\frac{h-1}{2}$. The average overhead V along a path with m attackers can then be computed as

$$\begin{aligned} E[V|\text{a path with } m \text{ attackers}] = \\ \frac{hL_a^S + \sum_{i=1}^{h-1}(h-i)L_a^i + hL_{ack}^P}{hW_sL_d} \\ + \frac{\frac{m(h-1)}{2}\left[L_q + W_s(1-p_a-p_r)L_{ack}^M\right]}{hW_sL_d} \end{aligned} \tag{23}$$

where L_a^S represents the length of the message appended by source node S in the PROBE packet, L_a^i the length of the message appended by hop-i ($i = 1, \cdots, h-1$) node, L_{ack}^P the length of a PROBE ACK, L_q the length of a "querying" packet, L_{ack}^M the length of a link-layer ACK, and L_d the length of a normal data packet. Specifically, the denominator and numerator of (23) represent the total amount of data transmissions and CAD overhead transmissions, respectively, in terms of bytes along a path during a probing interval. In our CAD design, all the L_a^i have the same length denoted as L_a, so the overhead in (23) can be simplified as

$$\begin{aligned} &E[V|\text{a path with } m \text{ attackers}] = \\ &\frac{L_a^S + \frac{h-1}{2}L_a + L_{ack}^P + \frac{m(h-1)}{2h}\left[L_q + W_s(1-p_a-p_r)L_{ack}^M\right]}{W_sL_d} \end{aligned} \tag{24}$$

Given a path without attackers, the overhead can be reduced to

$$\begin{aligned} E[V|\text{a path with no attackers}] &= \\ &\frac{hL_a^S + \sum_{i=1}^{h-1}(h-i)L_a^i + hL_{ack}^P}{hW_sL_d} \\ &= \frac{L_a^S + \frac{h-1}{2}L_a + L_{ack}^P}{W_sL_d} \end{aligned} \tag{25}$$

In our CAD algorithm, the specific configurations are $L_a^S = 82$ bytes, $L_a = 58$ bytes, $L_{ack}^P = 81$ bytes, $L_q = 29$ bytes and $L_{ack}^M = 44$ bytes.

1) Optimal Thresholds with Dynamic Channel Status: Fig. 4 depicts the adaption of the threshold with the varying network load. We can make the following observations. First, as the load increases, the channel collision probability increases with the channel busyness ratio, R_b. Second, since the threshold τ_d is designed to take into account the normal loss events, we can see an increase in threshold with the probability of collisions. Third, the graph also depicts the P_{dt} parameter estimated by the *upstream* node in the presence of an attacker (10% gray-hole attack). In Fig. 5, we make the similar observations. As the network load varies, we see that the upstream threshold τ_u increases with the estimated normal loss rate, p_r. The graph also indicates the loss rate (P_e) observed by the *downstream* node in the presence of 10% dropping attacker. In Fig. 4 and 5, we can clearly see that the misbehavior observed by the upstream and downstream nodes exceeded the protection margin of the thresholds τ_d and τ_u even in the presence of normal losses. Hence, CAD can effectively identify the attackers in the presence of normal channel losses.

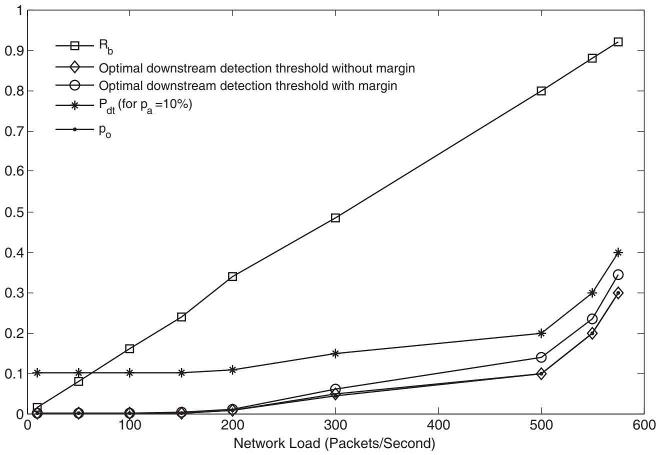

Fig. 4. Optimal thresholds with dynamic channel status: traffic monitoring and downstream detection threshold regarding the network load, with $p_e = 0$ and $p_a = 10\%$.

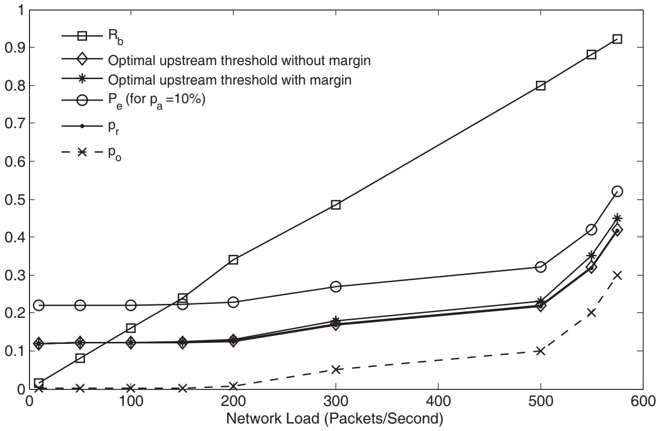

Fig. 5. Optimal thresholds with dynamic channel status: traffic monitoring and upstream detection threshold regarding the network load, with $p_e = 0.12$ and $p_a = 10\%$.

2) Performance of CAD: In this part, we show the performance of CAD in the presence of attackers and normal channel losses. We also highlight the effect of PROBE packets interval in terms of overhead and PDR.

Sensitivity on PROBE packet interval. In this scenario, we study the impact of interval (W_s) of PROBE packets on the performance of CAD. In Fig. 6, we present the PDR *vs* W_s curves for CAD in four scenarios with the selective dropping rate p_a set as 10%, 20%, 30%, and 50%, respectively. The curves demonstrate that CAD performed better at smaller intervals (at W_s = 10 and 10% dropping, PDR = 97%) as opposed to scenario where PROBE packets are sent at larger intervals (at W_s = 50 and 10% dropping, PDR = 89%). The reason that accounts for high PDR is that when a PROBE packet is sent within smaller intervals an attacker, if present, will be detected earlier and hence less number of packets will be lost due to malicious behavior.

In Fig. 7, we examine the tradeoff between the overhead and

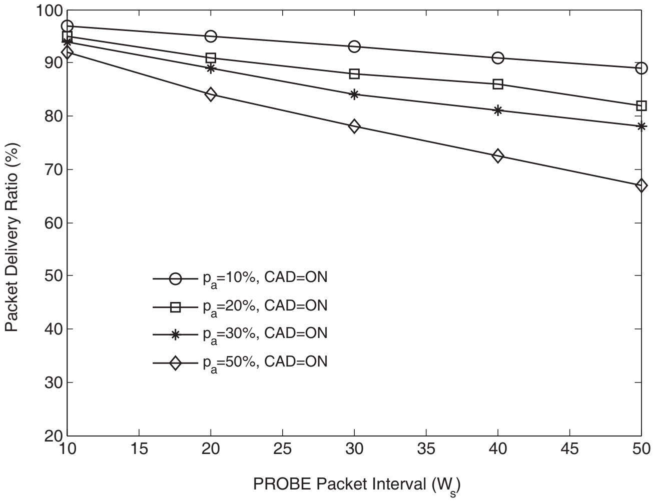

Fig. 6. Sensitivity on PROBE packet interval: performance of CAD in terms of PDR regarding PROBE packet interval W_s. Here, we consider "single" attacker (MAL=ON) in the path between source and destination.

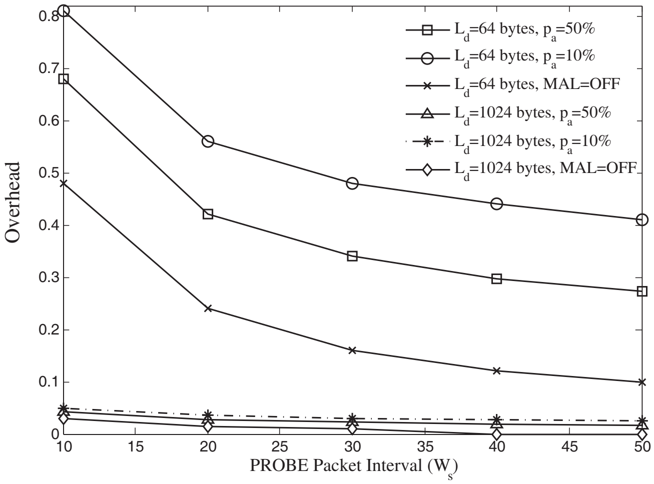

Fig. 7. Sensitivity on PROBE packet interval: performance of CAD in terms of transmission overhead regarding PROBE packet interval W_s. Here, we consider "single" attacker ($p_a = 10\%$ and $p_a = 50\%$) and "no" attacker scenario (MAL=OFF) in the path between source and destination. The average number of hops between source and destination is $h = 6$.

the PROBE interval W_s by presenting the CAD performance with selective dropping rates set as 10% and 50%. We compare the overhead under cases with attackers and without attackers, and the CAD overhead is computed according to (24) and (25) respectively. For the former case, we randomly place one attacker between the source and destination nodes (i.e., $m = 1$) along each path. The average number of hops of a path in our simulation topology is $h = 6$. Moreover, we also evaluate the overhead regarding the normal data packet size with two specific scenarios $L_d = 64$ and $L_d = 1024$ bytes, respectively. In Fig. 7, we can have the following observations. (a) Smaller probing interval W_s will lead to larger overhead, due to the more frequent CAD-related transmission. (b) CAD has a larger overhead with attackers present compared to the case without attackers. The reason is that the source node need to use extra messages to query the intermediate nodes for the link-layer acknowledgment proofs. (c) The normal data size has an obvious impact on the overhead. Since the amount of CAD transmission is fixed every W_s data packets, a larger data packet size implies a smaller overhead ratio regarding the normal data transmission. In practice, a preferred configuration is to use relative large data packet size for small overhead and small probing interval for high packet delivery ratio. (d) The overhead theoretically increases with the number of hops, mainly through the item $\frac{h-1}{2}L_a$. However, in our CAD design, L_a takes a value of 58 bytes, which constrains the overhead to a low level in practical application. For example, even for an impractically large network with average hops $h = 20$, the overhead (with a normal data packet size of 1024 bytes) is 0.091 and 0.03 for $W_s = 10$ and 50 respectively in the presence of 10% dropping attacker. It is noteworthy that all the four aspects of observations can be mathematically explained by equations (24) and (25).

Detection with normal losses absent. In Fig. 8, we evaluate the PDR as a function of different selective dropping rates. In order to highlight the impact of selective dropping attack, we control the normal loss rates as small as possible: the channel quality is assumed perfect and the traffic load is controlled low so that the channel collision probability is small. The curves in Fig. 8 illustrate the performance of CAD in the presence and absence of attacker(s), where we can have four observations. First, the PDR is degraded in the presence of an attacker (87% in the case of 10% dropping and 49% in case of 50% dropping). Second, after using CAD, the PDR is improved (97% in the case of 10% dropping and 92% in case of 50% dropping) because CAD detects the attacker and forwards the traffic to the destination through a different path. Third, the maximum PDR achieved using the CAD is less than the ideal case (98%), where no routers exhibit malicious behavior. The reason behind this is that some packets are already lost before CAD detects the attacker. Finally, Fig.8 shows the PDR in the presence of *multiple independent attackers*. In this scenario, we randomly choose 15% of the routers as selective dropping attackers in the forwarding paths between source and destination pairs. We make the following observations. In the case where "no detection" algorithm is employed, we see a significant degradation in the PDR (when p_a = 10%, PDR = 72% and PDR=12% when p_a= 50%). However, after using CAD, the PDR is improved to 87% in the case of 10% dropping and 66% in case of 50% dropping.

Detection with normal losses present. In this case, we study how CAD performs in the presence of both attacker(s) and normal loss events. We simulate the two-state Markovian model $((P_{gb}, P_{bg}) = (0, 1), (0.11, 0.99), (0.13, 0.953), (0.19, 0.866), (0.22, 0.88))$ in this experiment. We make the following observations from Fig.9: (a) The increased channel loss rate does cause more packet loss and hence the maximum PDR achieved (0.8 in case of $p_r = 0.2$) is less than the ideal case (98% in Fig.8); (b) The PDR is degraded in the presence of 10% (PDR = 0.69, $p_r = 0.2$) and 20% (PDR = 0.56, $p_r = 0.2$) selective dropping attacker when "no detection" algorithm is employed; (c) After using CAD, the PDR is improved (0.78 in the case of 10% dropping and 0.74 in the case of

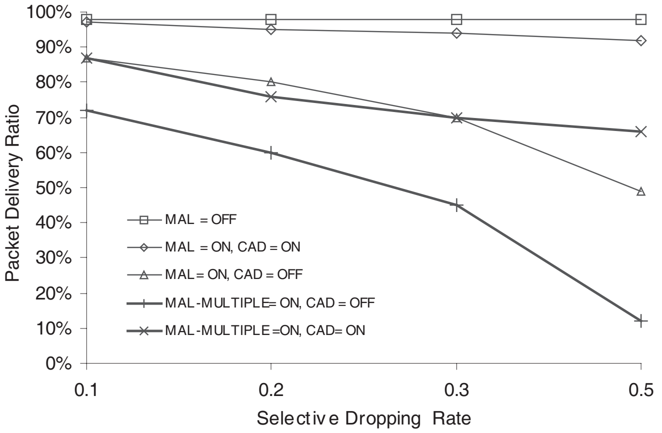

Fig. 8. Detection with normal losses absent: performance of CAD with normal losses absent and $W_s = 10$. Here, we consider three scenarios: (a) "no" attacker (MAL=OFF) (b) "single" attacker (MAL =ON); (c) "multiple" attackers (MAL-MULTIPLE=ON) present in the forwarding path between source and destination.

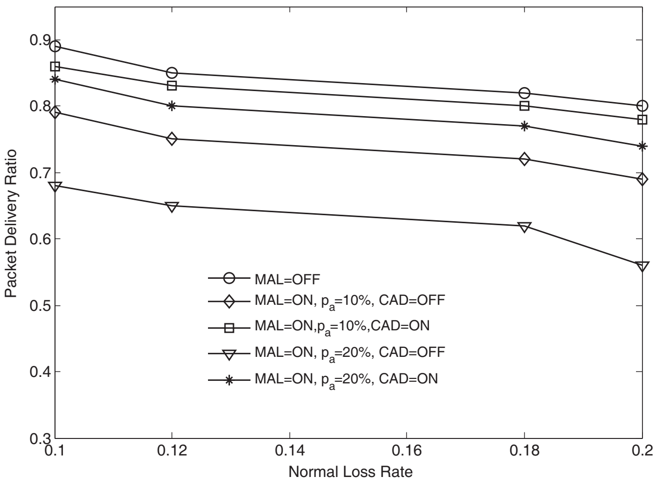

Fig. 9. Detection with normal losses: performance of CAD with normal losses present and $W_s = 10$. Here, we consider two scenarios: (a) "no" attacker (MAL=OFF), (b) "single" attacker (MAL =ON) present in the forwarding path between source and destination.

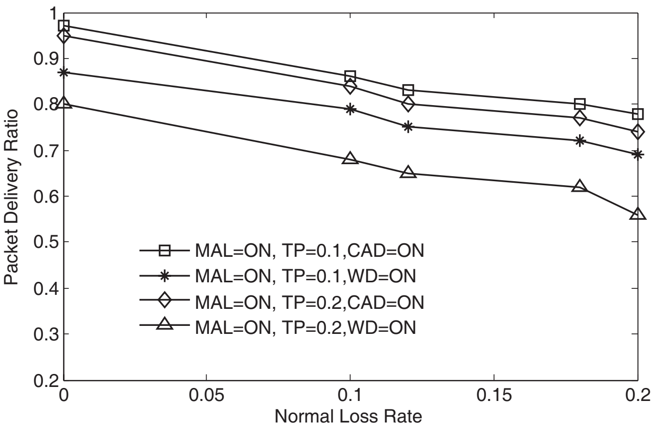

Fig. 10. CAD vs Watchdog (WD): performance of CAD compared to the Watchdog technique, with $W_s = 10$. Here we consider the presence of "single attacker" in the path between source and destination.

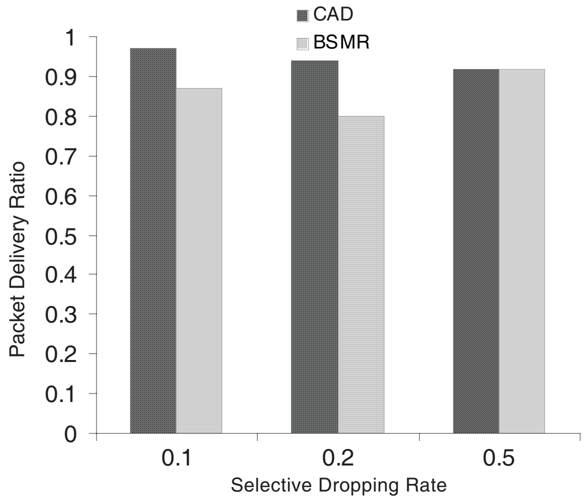

Fig. 11. CAD vs BSMR: performance of CAD compared to the BSMR technique (static thresholds $\delta = 10\%, \Delta = 20\%$), with $W_s = 10$. Here we consider the presence of "single attacker" in the path between source and destination.

20% dropping) because unlike prior works, CAD takes the loss rate of the channel into consideration while setting the detection thresholds (τ_d, τ_u). Hence, we can conclude that the increased channel loss rate will not prevent CAD from the detection of selective dropping attackers.

3) Comparison with Other Approaches: In this part, we compare CAD with the existing schemes Watchdog and BSMR.

CAD *vs* WD. In this scenario, we investigate the impact of the *limited transmit power attack* [2] on the performance of CAD and WD. The transmit power attack rate denoted as TP is the number of packets lost due to this attack out of the number of packets successfully received by attacker. It is known that a straightforward application of WD under the limited transmit power attack is not efficient [2]. As shown in Fig. 10, the PDR under Watchdog is obviously smaller (0.87 for TP = 0.1) than that under CAD (0.97 for TP = 0.1), when the attacker performs a transmit power attack. The reason for enhanced performance in CAD is that it employs both traffic overhearing and channel loss observation to detect an attacker. Therefore, when an attacker performs this attack, CAD can detect it because the loss rate observed by the downstream node exceeds the detection threshold τ_u. Hence, we can conclude that both upstream and downstream monitoring is necessary for accurately detecting the attackers.

CAD *vs* BSMR. In this case, we study the impact of detection thresholds on the performance of CAD and BSMR. From Fig.11, we make two observations. First, the PDR of BSMR is degraded (0.86 for 10% dropping) as compared to CAD (0.97 for 10% dropping) in the presence of 10% and 20% dropping attackers. The reason behind this is unlike CAD, BSMR employs *static* thresholds that are independent of normal channel losses which inturn prevented BSMR in detecting the 10% and 20% dropping misbehaviors. Hence, we argue that a channel-aware threshold is necessary for the accurate detection of attackers. Second, in the presence of 50% dropping attackers, CAD and BSMR performed better and the PDR is increased to 92% when compared to a case where "no detection" scheme is employed (49% in the case of 50% dropping).

VIII. Conclusion and Future Work

In this paper, we proposed an effective algorithm to detect and locate the selective forwarding attackers in WMNs. The particular challenging scenario we consider is that the intentional selective dropping may be interleaved with normal loss events due to wireless channel quality or medium access collisions. The proposed channel aware detection algorithm utilizes the methodologies of channel estimation and upstream/downstream traffic monitoring to discriminate the selective dropping attack from the estimated normal loss rates. We demonstrate how to compute the false alarm and missed detection probabilities for the CAD algorithm, and further derive the optimal detection thresholds to minimize the summation of the the false alarm and missed detection probabilities. Our simulation results show that with the presence of normal losses, CAD can detect the attackers efficiently and thereby increased the packet delivery ratio of the network. In this work, we assume that the system is free from collision or jamming attacks; we observe that when an attacker introduce noise to simulate a noisy channel, it indeed affects the sensing process which inturn leads to inaccurate threshold. In future, we plan to address these attacks and then extend CAD to deal with such attacks. For future work, we also plan to have more in-depth investigation of the scenario where multiple malicious nodes act in collusion.

References

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, no. 9, pp. S23-S30, Sept. 2005.

[2] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. International Conference on Mobile Computing and Networking*, Boston, MA, 2000.

[3] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Elsevier's AdHoc Networks J.*, vol. 1, no. 2-3, pp. 293-315, Sept. 2003.

[4] B. Xiao, B. Yu, and C. Gao, "CHEMAS: identify suspect nodes in selective forwarding attacks," *J. Parallel and Distrib. Computing*, vol. 67, no. 11, pp. 1218-1230, Nov. 2007.

[5] D. Manikantan Shila and T. Anjali, "Defending selective forwarding attacks in mesh networks," in *Proc. 2008 Electro/Information Technology Conference* , Ames, IA, May 2008.

[6] R. Curtmola and C. Nita-Rotaru, "BSMR: Byzantine-resilient secure multicast routing in multi-hop wireless networks," in *Proc. Sensor, Mesh and Ad Hoc Communications and Networks*, June 2007.

[7] A. Perrig, R. Canetti, D. Tygar, and D Song, "The TESLA Broadcast Authentication Protocol," in *RSA CryptoBytes*, Summer 2002.

[8] I. Khalil, S. Bagchi, and N. B. Shroff, "LiteWorp: detection and isolation of the wormhole attack in static multihop wireless networks," *Computer Networks: The International J. Computer and Telecommun. Networking* , vol. 51, no. 13, pp. 3750-3772, Sept. 2007.

[9] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. ACM Workshop on Wireless Security (WiSe 2002)*, Sept. 2002.

[10] Y. Hu, D. B. Johnson, and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 175-192, July 2003.

[11] Y. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Trust modeling and evaluation in ad hoc networks," in *Proc. IEEE GLOBECOM '05*, vol. 3, Dec. 2005.

[12] L. F. Perrone and S. C. Nelson, "A study of on-off attack models for wireless ad hoc networks," *Operator-Assisted (Wireless Mesh) Community Networks*, pp. 1-10, Sept. 2006.

[13] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks," in *Proc. INFOCOM, 2006*, pp. 1-13, Apr. 2006.

[14] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "Attacks on trust evaluation in distributed networks," in *Proc. 40th Annual Conference on Information Sciences and Systems 2006*, no. 22-24, pp. 1461-1466, Mar. 2006.

[15] K. Sanzgiri, D. LaFlamme, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "Authenticated routing for ad hoc networks," *IEEE J Sel. Areas Commun.*, vol. 23, no. 3, pp. 598-610, Mar. 2007.

[16] Y. Hu, D. B. Johnson, and A. Perrig, "Ariadne: a secure on-demand routing protocol for ad hoc networks," in *Proc. Mobicom'02*, pp. 12-23, 2002.

[17] Y. Hu, D. B. Johnson, and A. Perrig, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proc. ACM Workshop on Wireless Security (WiSe)*, pp. 30-40, 2003.

[18] Y. Hu, D. B. Johnson, and A. Perrig, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proc. IEEE INFOCOM 2003*, vol. 3, pp. 1976-1986, Mar. 2003.

[19] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Proc. Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2004.

[20] M. E. M. Campista, D. G. Passos, P. M. Esposito, I. M. Moraes, C. V. N. de Albuquerque, D. C. M. Saade, M. G. Rubinstein, L. H. M. K. Costa, and O. C. M. B. Duarte, "Routing metrics and protocols for wireless mesh networks," *IEEE Network*, vol. 22, no. 1, pp. 612, Jan. 2008.

[21] W. Yu, Z. Ji, and K. J. R. Liu, "Securing cooperative ad-hoc networks under noise and imperfect monitoring: strategies and game theoretic analysis," *IEEE Trans. Inf. Forensics and Security*, vol. 2, no. 2, pp. 240-253, June 2007.

[22] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, "Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks," in *Proc. IEEE IPCCC*, pp. 463-469, 2004.

[23] S. Buchegger and J. Y. Le Boudee, "Self-policing mobile ad hoc networks by reputation systems," *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 101-107, July 2005.

[24] D. Manikantan Shila, Y. Cheng, and T. Anjali, "Channel-aware detection of gray hole attacks in wireless mesh networks," in *Proc. GLOBECOM*, 2009, submitted.

[25] Y. Desmedt and Y. Frankel, "Shared generation of authentication and signatures," *Advances in Cryptology (CRYPTO'91)*, Berlin, pp. 457-469, 1991.

[26] J. Newsome, E. Shi, D. Song, and A. Perrig, "Sybil attack in sensor networks: analysis and defenses," in *Proc. IPSN '04*, New York, pp. 259-268, 2004.

[27] C. Piro, C. Shields, and B. N. Levine, "Detecting the sybil attacks in mobile ad hoc networks," in *Proc. SecureComm*, 2006.

[28] B. Parno, A. Perrig, and V. D. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proc. IEEE Symposium on Security and Privacy*, 2004, pp. 49-63.

[29] L. Mingyan, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *Proc. IEEE INFOCOM 2007*, pp. 1307-1315, Anchorage, AK, May 2007.

[30] Y. Fan, Y. Hao, and L. Zhen, "Catching "moles" in sensor networks," in *Proc. International Conference on Distributed Computing Systems*, Toronto, Canada, June 2007.

[31] X. Wu and N. Li, "Achieving privacy in mesh networks," in *Proc. Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks,* Oct. 2006, Alexandria, VA, USA.

[32] T. Staub, D. Balsiger, M. Lustenberger, and T. Braun, "Secure remote management and software distribution for wireless mesh networks," in *Proc. ASWN,* Santander, Spain, May 2007, pp. 4754.

[33] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *International J. Inf. Security*, vol. 1, no. 1, pp. 36-63, Aug. 2001.

[34] V. R. Gandikota, B. R Tamma, and C. S. R. Murthy, "Adaptive FEC-based packet loss resilience scheme for supporting voice communication over ad hoc wireless networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 10, pp. 1184-1199, 2008.

[35] S. Miller and J. McDougall, "Sensitivity of wireless network simulations to a two-state Markov model channel approximation," in *Proc. IEEE GLOBECOM*, vol. 2, no. 1, pp. 697-701, Dec. 2003.

[36] Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std. 802.11 Std., 1999.

[37] H. Zhai, X. Chen, and Y. Fang, "How well can the IEEE 802.11 wireless LAN support quality of service?" *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 3084-3094, Nov. 2005.

[38] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[39] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, 3rd ed., 2001.

[40] M. H. Protter and C. B. Morrey, *A First Course in Real Analysis*. Springer, 2nd ed., 1991.

[41] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*. Pearson/Addison Wesley, 2007.

[42] K. Fall and K. Varadhan, NS notes and documentation, The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC, 1997.

[43] I. D. Chakeres and E. M. Belding-Royer, "AODV routing protocol implementation design," in *Proc. International Workshop on Wireless Ad Hoc Networking (WWAN)*, Tokyo, Japan, Mar. 2004.

[44] D. Koutsonikolas, J. Dyaberi, P. Garimella, S. Fahmy, and Y. C. Hu, "On TCP throughput and window size in a multihop wireless network testbed," in *Proc. WiNTECH07*, Sep. 2007, Montral, Quebec, Canada.

[45] D. A. Maltz, "On-demand routing in multi-hop wireless mobile ad hoc networks," Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, May 2001.

Devu Manikantan Shila (S'07) received the MS degree in computer engineering from Illinois Institute of Technology, Chicago, USA, in 2007. She is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, Illinois Institute of Technology. Her current research interests include security of wireless mesh networks, algorithm design, capacity analysis and applications of game theory in wireless networks.

Yu Cheng (S'01-M'04-SM'09) received the B.E. and M.E. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada, in 2003. From September 2004 to July 2006, he was a postdoctoral research fellow in the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA, as an Assistant Professor. His research interests include next-generation Internet architecture and management, wireless network performance analysis, network security, and wireless/wireline interworking. He received a Postdoctoral Fellowship Award from the Natural Sciences and Engineering Research Council of Canada (NSERC) in 2004, and a Best Paper Award from the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine'07), Vancouver, British Columbia, August, 2007. He served as a Technical Program Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009. He is an Associated Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an Area Editor for ELSEVIER JOURNAL OF COMPUTER NETWORKS.

Tricha Anjali (S'01-M'04-SM'09) received the (Integrated) M.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Bombay, in 1998 and Ph.D. degree from Georgia Institute of Technology in May 2004. Currently, she is an Assistant Professor at the Electrical and Computer Engineering Department at the Illinois Institute of Technology. Her research interests include design and management of MPLS and optical networks, wireless mesh networks.