

Stasis Trap: Cross-Layer Stealthy Attacks in Wireless Ad Hoc Networks^{*}

Kaigui Bian, Jung-Min Park, and Ruiliang Chen
Laboratory for Advanced Research in Information Assurance and Security (ARIAS)
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
{kgbian, jungmin, rlchen}@vt.edu

Abstract—Denial-of-Service (DoS) attacks pose a major threat to the availability of wireless ad hoc networks. Fault tolerant operation of wireless ad hoc networks will depend on the placement of DoS countermeasures in sufficiently robust form. In this paper, we describe a novel type of DoS attack called the *Stasis Trap* attack, and propose a technique for detecting such an attack. *Stasis Trap* attack has two distinguishing characteristics—it has a cross-layer design, and is stealthy. The *Stasis Trap* attack has a cross-layer design in that it is launched from the MAC layer but its aim is to degrade the end-to-end throughput of flows at the transport layer by exploiting TCP’s congestion-control mechanism. Specifically, an adversary launches a *Stasis Trap* attack against neighboring nodes by periodically preempting the wireless channel in order to cause large variations in the round trip time (RTT) of TCP flows. Channel preemptions are carried out by manipulating the back-off mechanism of the Distributed Coordinating Function of the 802.11 MAC protocol. The periodic preemptions induce large RTT variations in the TCP flows that are within the transmission range of the adversary. This in turn causes a significant drop in the throughput of those flows, thereby creating a “stasis trap” around the adversary that entangles TCP flows. The aforementioned attack severely degrades end-to-end throughput but has very little effect on MAC-layer throughput, and hence it is very hard to detect at the MAC layer, which is its point of attack. In this sense, this attack is stealthy. To detect the *Stasis Trap* attack, we propose a *minimax robust decentralized detection* framework with robust hypothesis testing.

I. INTRODUCTION

Wireless ad hoc networks extend mobility into the realm of autonomous, mobile, and wireless domains, where a set of wireless devices (nodes) form the network routing infrastructure in an ad hoc fashion. Because wireless ad hoc networks are likely to be deployed in mission-critical applications, the problem of ensuring a network’s reliability and availability is one of the most important factors in the design and deployment of such networks.

The IEEE 802.11 MAC protocol’s Distributed Coordination Function (DCF) [5] has been shown to be quite effective in coordinating channel access for contending nodes in a non-hostile environment. However, previous research [2, 6, 7, 8] has shown that the 802.11 DCF is vulnerable in a hostile environment. In a hostile environment, no notion of trust can be assumed; adversaries can exploit the contention coordination mechanism of the DCF to achieve their objectives, which can range from selfish exploitation of available network resources to malicious network disruption. Such attacks against the MAC protocol would require the

adversary to change the protocol’s parameters. The increased requirement for flexible and configurable protocols has led to wireless network devices that are programmable. This in turn has made it possible for an adversary to tamper with software and firmware, modify the wireless interface and MAC protocol parameters, and ultimately exploit the protocol.

There is a significant amount of research that exists on the security vulnerabilities of the three layers—viz., TCP, IP, and MAC layers. Various attacks against each layer have been documented. However, very little research exists on attacks that take a *cross-layer* approach. This type of attack exploits the vulnerability of a particular layer (attack point) to launch the attack, but ultimately aspires to disrupt the operations of another layer (target point). The JellyFish attack [1] is an example of such an attack. Such attacks are “stealthy” because they are difficult to detect using conventional detection schemes. Detection is difficult because the point of attack and the target point reside in different layers of the protocol stack.

In this paper, we investigate a novel type of cross-layer attack called the *Stasis Trap* attack that can be launched against 802.11-based ad hoc networks. In this attack, an adversary uses the MAC layer as the point of attack but ultimately aims to degrade TCP-layer throughput of flows within its transmission range. To solve the channel contention problem, DCF adopts a back-off mechanism, which favors the node with the smallest back-off deferment time amongst all contending nodes. In the *Stasis Trap* attack, an attacker preempts the channel with a high probability by exploiting this back-off mechanism. Specifically, the attacker manipulates the back-off values by using a small contention window size. According to the rules of DCF, other non-malicious nodes will refrain from transmitting during the attacker’s transmission. Once the channel is preempted, the attacker transmits data for a long enough period to cause noticeable delays in the TCP flows that are traversing through the neighboring nodes, and then halts transmission. After staying dormant for a certain time duration, the attacker preempts the channel again and repeats the same process. Note that the attack duration itself is short compared to the time interval between the attacks.

The periodic preemptions of the channel by the attacker will cause periodic delay spikes in the TCP flows that traverse through the neighboring nodes. In turn, this will likely cause the Retransmission Timeout (RTO) of the flows to expire. The congestion control mechanism will react to the RTO expiration by reducing the congestion window size to one and retransmitting outstanding packets. This chain of events will

^{*} The work presented in this paper was supported in part by the National Science Foundation under Grant CNS-0524052.

result in the serious drop of the end-to-end throughput of the flows. Using the above technique, the attacker effectively creates a “stasis trap” around itself that degrades the throughput of any flow that traverses through it.

Detecting the Stasis Trap attack is difficult. We address the problem of detection by casting the problem within a *robust decentralized detection* framework [10]. The framework includes multiple *distributed observers* and a *data fusion center* that performs *robust hypothesis testing*.

The rest of the paper is organized as follows. Related work is presented in Section II, and technical background is outlined in Section III. A discussion on the *Stasis Trap* attack is provided in Section IV. The robust decentralized detection framework is introduced in Section V, and we conclude the paper in Section VI.

II. RELATED WORK

MAC layer misbehaviors in 802.11 DCF can be classified into two categories: malicious Denial-of-Service (DoS) attacks and selfish misbehaviors. MAC-layer DoS attacks [4] prevent or hamper non-malicious nodes from accessing the channel. For instance, in RTS/CTS attacks [8], an adversary sends RTS/CTS frames to spuriously reserve the channel without real data transmissions. In another type of attack, called the NAV attack [2], an adversary sets large duration values in RTS (request to send) or CTS (clear to send) frames to reserve a longer time duration than the duration of actual transmissions. In [6], the authors describe one type of selfish misbehavior in which a selfish host unilaterally modifies parameters in the back-off mechanism to get priority access to the channel. As a result, the selfish node achieves better throughput.

Malicious misbehavior can be readily identified by a detection technique [8], in which neighbors calculate the actual transmission time by sensing DATA/ACK frames. Assuming the random back-off values are observable, a MAC receiver can carry out a sequential test [7] to analyze the distribution of this random variable. Previous MAC-layer DoS countermeasures all utilize a centralized detection framework that employs independent observers who do not cooperate with each other.

III. EXPLOITING THE 802.11 DCF AND CONVENTIONAL DETECTION TECHNIQUES

In 802.11 DCF, a host with data to transmit has to defer transmission for a back-off period to avoid an RTS collision [5]. The back-off value is randomly chosen from the range $[0, CW]$, where CW is the contention window size. When the back-off timer is decremented to zero, the host can send an RTS frame to reserve the channel. If an attacker intentionally

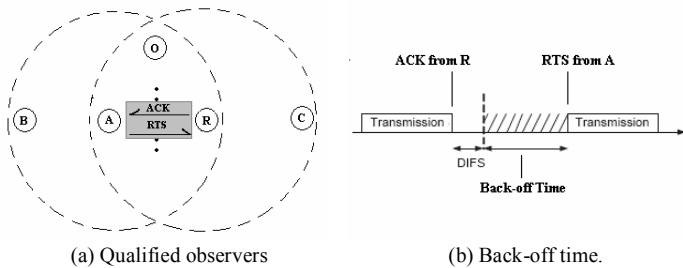


Fig. 1. Illustrations of two fundamental concepts.

uses a small CW value, it can preempt the channel and prevent others from accessing the channel with a high probability.

To detect such misbehaviors, we need to persistently observe the back-off values and perform a sequential analysis. To detect an attack launched by a node, say node A in Fig. 1(a), the back-off values between consecutive MAC transmissions from A need to be measured by one of the nodes within the transmission range of A [8]. Suppose that attacker A transmits to node R . In this scenario, only nodes R and O can calculate the back-off time by computing the time interval between the reception of the ACK transmitted from R and the reception of the RTS transmitted from A (see Fig. 1(b)). Note that the DIFS (Distributed Inter-Frame Space) value is fixed in 802.11. We use the term *qualified observers* to describe the nodes that are located within the transmission ranges of both the MAC sender and the receiver. These observers are “qualified” to calculate the back-off times used by the sender. In Fig. 1(a), R and O are qualified observers while B and C are not. In conventional attack detection schemes (e.g., [7]), qualified observers R and O perform sequential tests on observed back-off values until a decision (of whether A is acting maliciously) is made. This is often a time-consuming process.

Unfortunately, A can thwart the aforementioned detection scheme by changing its transmission destination. For instance, A can change the destination of its transmission to B before R or O can make a detection decision. Now that B is the new receiver, R and O are no longer qualified observers, and B becomes the only qualified observer. Similarly, A can switch back to R before B can make a detection decision. Using this technique, the attacker can thwart detection of sequential tests performed by neighboring nodes.

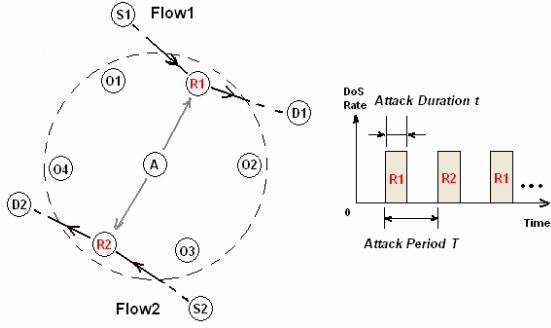
IV. THE STASIS TRAP ATTACK

In this section, we describe how an attacker can manipulate MAC protocols to launch a Stasis Trap attack. We provide an analytical model with simulation results and then evaluate several TCP variants’ vulnerability to this attack.

A. Overview of the Stasis Trap Attack

As discussed above, a MAC layer attacker can preempt the wireless channel by manipulations of the back-off mechanism. Moreover, the attacker can constantly change the transmission destination to evade detection from qualified observers. Due to the randomness of the back-off values, it is difficult for any single observer to collect sufficient evidence during a given observation period to make a detection decision. An example is shown in Fig. 2(a). In the figure, the attacker A can simply switch transmission destinations between $R1$ and $R2$ in a round-robin manner, so that no node can persistently observe A ’s back-off values.

In a Stasis Trap attack, an attacker preempts a channel in short periodic bursts. Hence, the attack has very little effect on the channel contention mechanism or the MAC-layer throughput of the corresponding links. The true target of the Stasis Trap attack is not the MAC layer but is the TCP layer. The attack’s aim is to degrade the end-to-end throughput of TCP flows traversing through the neighboring nodes (in Fig. 2(a), these would be Flow 1 and Flow 2). It is well known that



(a) Stasis Trap attack scenario (b) Attack pattern
Fig. 2. Stasis Trap attacks against TCP flows.

TCP’s RTO mechanism—which was originally designed to detect packet losses—can react inappropriately to frequent and large RTT (round trip time) delays. Specifically, if the RTT is delayed beyond the RTO, the TCP sender will assume packet loss, and respond by reducing the congestion window size to one MSS (maximum segment size) and retransmitting outstanding packets. Thus, if an attacker preempts the channel for a duration t (we call this the *attack duration*), which is longer than the RTO, then the targeted flows (such as Flow 1 and 2 in Fig. 2(a)) will simultaneously “time out”, since the packets of these flows cannot be forwarded due to channel contention. This, in turn, will trigger a response from the congestion control mechanism. As a result, the throughput of the flows will drop significantly. The attacker can periodically repeat the above process, using an *attack period* T , to cause periodic “outages”. The attack pattern is shown in Fig. 2(b).

B. Analysis of the Stasis Trap Attack

In 802.11, if a host’s data transmission is successful, the host resets its contention window size, CW , to a minimum value CW_{min} ; otherwise, CW is doubled, subject to a maximum of CW_{max} . To simplify the analysis, we assume that a legitimate host always selects a back-off value from $[0, CW_{min}]$ and an attacker selects a back-off value from $[0, CW_A]$. In the analysis, we only consider a single attacker; simultaneous attacks by multiple attackers are not considered.

For an attacker to preempt the channel, the following condition needs to hold:

$$B_A < B_i,$$

where B_A and B_i are the back-off values of the attacker and its neighbor, node i , respectively. Suppose the relation of the two contention window sizes are given by

$$CW_A < CW_{min}, \text{ where } CW_A = CW_{min} / S \text{ and } S > 1,$$

where S denotes the attack aggressiveness parameter. Suppose

that the back-off values of n contending neighbors are n i.i.d. random variables. We further assume that the back-off variable has a *uniform distribution* in the range $[0, CW]$. Then the probability that $B_A < B_i$ is:

$$P(B_A < B_i) = \int_{CW_{min}/S}^{CW_{min}} P(B_A < B_i | B_i = y) \cdot \frac{1}{CW_{min}} dy + \int_0^{CW_{min}/S} P(B_A < B_i | B_i = y) \cdot \frac{1}{CW_{min}} dy = (2S - 1) / 2S. \quad (1)$$

The probability that the attacker’s back-off is smaller than that of all n neighbors is:

$$P(B_A < \min \{B_1, \dots, B_n\}) = \prod_{i=1}^n P(B_A < B_i) = [(2S - 1) / 2S]^n. \quad (2)$$

C. Simulation Results

Using the *ns-2* [11] simulator, we investigated the effect of the Stasis Trap attack on several variants of TCP. Node mobility was not considered in the simulations. The network topology that was used is shown in Fig. 2(a). Host A is the attacker, and it has six one-hop neighbors. The neighboring nodes are all non-malicious. The senders $S1$ and $S2$ initiate two TCP flows that traverse through intermediate nodes $R1$ and $R2$, respectively. The transmission range is 250 m, and the channel capacity is 1 Mbps. To evade persistent observation from neighboring nodes, A constantly switches the destination of transmission between $R1$ and $R2$. We fixed the value of the attack aggressiveness parameter S to four.

In the simulations, we considered three TCP variants: *Reno*, *Sack*, and *Vegas*. A large body of work has been done to enhance TCP’s ability to handle congestion while maximizing throughput. TCP Sack and TCP Vegas are two variants of TCP that have more advanced congestion control mechanisms compared to TCP Reno. In TCP Sack, when a receiver holds non-contiguous data, it sends duplicate ACKs bearing the Sack option to inform the sender what has been correctly received. TCP Vegas uses delay-based congestion control. This congestion control technique improves the fast retransmission mechanism by enabling the sender to trigger congestion control without having to wait for duplicate ACKs. Our simulation results show that all three TCP variants—TCP Reno, TCP Sack, and TCP Vegas—are vulnerable to the Stasis Trap attack. In Fig. 3, the average throughput of Flows 1 and 2 is plotted as a function of time. The attack duration, t , was set to one second and the attack period, T , was set to two seconds.

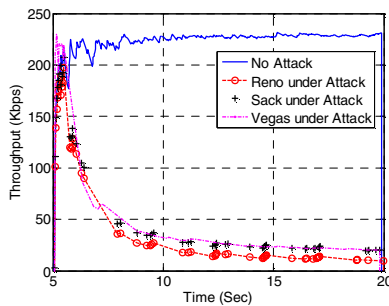


Fig. 3. TCP variants’ throughput vs. time.

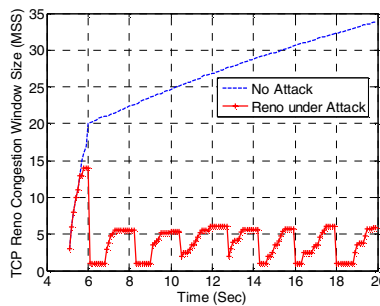


Fig. 4. TCP Reno Cwnd size vs. time.

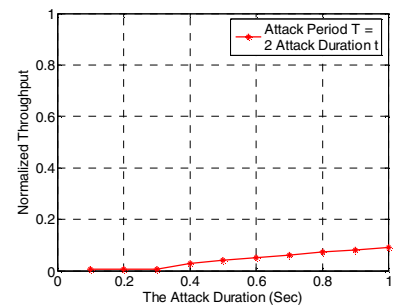


Fig. 5. Throughput vs. attack duration t .

Fig. 3 shows that the Stasis Trap attack is quite effective in degrading the throughput of all three TCP variants. In Fig. 4, the size of the congestion window is plotted as a function of time for TCP Reno. As before, t was set to one second and T was set to two seconds. We can see that the congestion window (Cwnd) size is periodically reduced to one MSS at fairly regular intervals; this interval coincides with the attack period.

Our simulation results suggest that the effectiveness of the Stasis Trap attack is not sensitive to the value of t in relation to the RTO value. We were able to observe that the Stasis Trap attack can cause the flows' throughput to drop down close to zero even when the value of t is smaller than the RTO value as long as the attack frequency, $1/T$, is sufficiently high. For the plot shown in Fig. 5, we set the attack period, T , as $T = 2t$, and varied the attack duration, t , from 100 ms to 1 second. The figure shows that the TCP throughput is close to zero even when the value of t is as small as 100 ms. In another set of simulation experiments, we fixed the value of t and varied the value of T from t to 1 second. Fig. 6 shows two sets of results, one with t set to 0.1 seconds and the other one with t set to 0.2 seconds. We can see from the figure that the throughput is zero or close to zero when the attack period, T , is less than 0.5 seconds. From Figs. 5 and 6, we can conclude that the amount of throughput degradation due to the Stasis Trap attack is sensitive to the attack frequency, $1/T$.

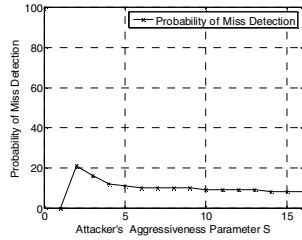
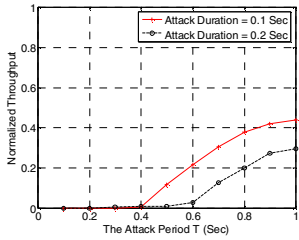


Fig. 6. Throughput vs. attack period T . Fig. 7. Probability of miss detection vs. S .

V. ROBUST DECENTRALIZED DETECTION IN WIRELESS AD HOC NETWORKS

In this section, we propose a detection framework that is capable of detecting Stasis Trap attacks. The framework is based on a *minimax robust decentralized detection* approach.

A. Minimax Robust Decentralized Detection

We consider a decentralized detection framework [9]. In this framework, each distributed sensor node within a group makes a *local decision* based on its own observations, and then sends the local decision information to a node that collects and processes this information; this node is called the *fusion center*. The fusion center applies a fusion rule to process the collected local information and ultimately makes a global decision based on the processed result. In a *centralized* detection scheme, each sensor transmits its measurements to the fusion center, which solves a classical hypothesis testing problem to select one of two hypotheses. In contrast, in our approach, each sensor carries out hypothesis testing and sends its local decision to the fusion center, which applies a global decision rule to make the final decision. In other words, hypothesis testing is done *locally* as well as globally—in this sense, our framework is

decentralized. There are two kinds of fusion rules, namely *block detection* and *sequential detection*. In block (fixed-sample-size) detection, distributed sensors communicate with the fusion center for a fixed period of time. In sequential detection, the sensors keep communicating with the fusion center until the fusion center makes a final decision via a sequential test. In our framework formulation, we choose block detection since it incurs less communication overhead.

For each sensor to perform robust hypothesis tests, it needs to know the distribution of the attacker's back-off value. As discussed in Section III, it is not feasible for each sensor (or observer) to obtain such information. Hence, we utilize the *Least Favorable Distribution (LFD)* [10], which denotes the distribution of the attacker's back-off values that maximizes the detection time in a decentralized detection scheme. In a manner similar to the technique used in [3], we replace the attacker's unknown distribution with the LFD in the robust hypothesis tests performed by each sensor.

We adopt the *minimax robust detection* approach where the goal is to optimize system performance when the system is operating at the least favorable operating point. In our framework, the least favorable operating point corresponds to the worst-case instance of an attack, and system performance is measured in terms of detection *false alarm* (P_{FA}) and *miss detection* probabilities (P_M).

The topology of the network under consideration is illustrated in Fig. 2(a). The one-hop neighbors of attacker A (i.e., the qualified observers $O1$ and $O2$, or $O3$ and $O4$) act as sensors, and the receiver ($R1$ or $R2$) act as a fusion center.

B. Problem Formulation

We need to define the hypotheses that will be used in the hypothesis testing. These are:

H_0 : the suspect node is legitimate;

H_1 : the suspect node is misbehaving.

In our problem formulation, we use a *minimax test* that minimizes the maximum of P_{FA} and P_M . Assuming all qualified observers use the identical local decision rule ϕ and the receiver (fusion center) has a fusion rule γ , our goal is to find a pair of optimal decision rules (ϕ, γ) that minimize the value of $\max(P_{FA}, P_M)$. Hence, the problem can be formulated as

Problem (P1):

$$\inf_{(\phi, \gamma)} \left(\max \{ P_{FA}((\phi, \gamma), P_0), P_M((\phi, \gamma), P_1) \} \right).$$

Let a node's back-off value be represented by the random variable X , which is uniformly distributed in $[0, CW_{min}]$. Let P_0 represent the distribution of X under H_0 . Similarly, let P_1 represent the distribution of X under H_1 . Under H_0 , X is uniformly distributed in $[0, CW_{min}]$; under H_1 , it is uniformly distributed in $[0, CW_A]$. Since the value of CW_A is controlled by the attacker, P_1 is not known completely. However, we can confine P_1 to be a distribution in an *uncertainty class*, which includes distributions corresponding to attacks with different CW_A values. Let $f_1(x)$ denote the pdf of P_1 . The *uncertainty class* F_1 can be defined as:

$$F_1 = \{ P_1 : f_1(x) = S / CW_{min}, x \in [0, CW_{min} / S], S > 1 \}. \quad (3)$$

Then, a LFD, Q_1 , can be defined as the distribution of the worst-case attack over the uncertainty class F_1 . When P_1 is

replaced with Q_1 , Problem (P1) becomes equivalent to the following minimax robust detection problem

Problem (P2):

$$\inf_{(\phi, \gamma)} \left(\max \left\{ P_{FA}((\phi, \gamma), P_0), P_M((\phi, \gamma), Q_1) \right\} \right).$$

Assume that there are N qualified observers capable of measuring the back-off value X . Suppose that at time k , an observer i has collected a vector of observations $X_i^k = (x_i^1, \dots, x_i^k)$, derives a local decision $u_i^k = \phi(X_i^k)$, and sends this u_i^k to the receiver. Suppose, at the same time, the receiver overhears local decisions from neighboring observers and performs a fusion policy γ . Then, the optimal decision rules (ϕ, γ) to Problem (P2) take the form of likelihood ratio tests [10], where P_0 has the pdf $f_0(x)$ and Q_1 has the pdf $f_1^*(x)$.

Now, let us derive the local decision rule at each sensor node. Since x_i^k are independent observations, the log likelihood ratio for X_i^k between two hypotheses H_1 and H_0 is

$$l_q = \ln \left(\prod_{j=1}^k \frac{f_1^*(x_i^j)}{f_0(x_i^j)} \right) = \sum_{j=1}^k \ln \left(\frac{f_1^*(x_i^j)}{f_0(x_i^j)} \right). \quad (4)$$

The local decision rule ϕ is a monotone threshold likelihood ratio test (MLRT) with the following deterministic thresholds:

$$u_i^k = \phi(X_i^k) = \begin{cases} 1, & \text{if } l_q \in (0, t_1), \\ d, & \text{if } l_q \in (t_{d-1}, t_d), \\ D, & \text{if } l_q \in (t_{D-1}, \infty). \end{cases} \quad (5)$$

The local decision u_i^k takes discrete integer values in the range $[1, D]$. The likelihood that the observed node is malicious increases as the value of u_i^k increases. The values t_1, \dots, t_{D-1} are deterministic monotone thresholds used in MLRT.

Now, let us derive the fusion rule applied at the fusion center. Suppose, at time k , an observer i has reported a vector of local decisions $U_i^k = (u_i^1, \dots, u_i^k)$, and the fusion center collects a group of local decision vectors (U_1^k, \dots, U_N^k) . Let $f_0(u_i^k)$ denote the pmf of U_i^k under H_0 , and let $f_1(u_i^k)$ denote the pmf of U_i^k under H_1 , which can be derived from Q_1 . Since u_i^k are independent local decisions, the likelihood ratio for U_i^k is:

$$\Lambda = l_q(U_1^k, \dots, U_N^k) = \prod_{j=1}^k l_q(u_1^j, \dots, u_N^j) = \prod_{j=1}^k \left(\prod_{i=1}^N \frac{f_1(u_i^j)}{f_0(u_i^j)} \right). \quad (6)$$

The global decision rule γ is the following likelihood ratio test (LRT):

$$\gamma(U_1^k, \dots, U_N^k) = \begin{cases} \text{accept } H_1, & \text{if } \Lambda > th \\ \text{accept } H_0, & \text{otherwise} \end{cases}, \quad (7)$$

where th is the threshold predefined in LRT and Λ is the *accumulative detection information*, which needs to be shared by all neighbors. The performance of our detection framework is shown in Fig. 7. The figure shows the probability of miss detection (false negative) as a function of S .

C. Deployment Issues

Suppose attacker A is transmitting to $R1$ (see Fig. 2(a)). In this case, the qualified observers $O1$, $R1$, and $O2$ act as distributed sensor nodes, and receiver $R1$ acts as the fusion center (note that $R1$ acts as both a sensor and the fusion center). As mentioned before, to evade persistent surveillance, A may switch the destination of its transmission from $R1$ to $R2$ before $R1$ can make a detection decision. In this case, $R2$ needs to

have access to the accumulated detection information so that it can take over the detection process from $R1$.

To address the problem of sharing detection information among nodes, we propose a simple protocol called *Echo*. Let us use the network shown in Fig. 2(a) as an example. In *Echo*, the current receiver $R1$ (which is also the fusion center) sends a detection payload, which includes the accumulative detection information Λ , to the node being observed, which is node A ; in response, A broadcasts this information to its neighboring nodes. Note that A cannot broadcast falsified information since this will be readily detected by $R1$. Suppose that A switches the transmission destination to $R2$. This is no longer a problem since observers $O3$, $R2$, and $O4$ can resume the detection process by using the information broadcast by A .

The detection decision is made by either $R1$ or $R2$; the one that collects a sufficient number of local decisions first makes the detection decision.

VI. CONCLUSION

In this paper, we have described a novel DoS attack called the Stasis Trap attack. Using simulation results, we have demonstrated the effectiveness of the attack. This attack employs a cross-layer design in the sense that the point-of-attack is at the MAC layer but the attack's impact is only evident at the transport layer. This unique design makes the task of detecting such attacks very difficult. To address the problem of attack detection, we proposed a detection framework based on a minimax robust decentralized detection approach. This detection framework overcomes the drawbacks of the conventional MAC-layer attack detection schemes that employ a centralized detection approach.

REFERENCES

- [1] I. Aad, J. Hubaux and E. W. Knightly, "Denial of service resilience in ad hoc networks," in *Proceedings of Mobile computing and networking*, Sept. 2004, pp. 202–215.
- [2] J. Bellardo, S. Savage and D. Medina, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions," in *Proceedings of the USENIX Security Symposium*, August 2003, Aug. 2003, pp. 15–27.
- [3] E. Geraniotis and Y.A. Chau, "Robust data fusion for multisensor detection systems," *IEEE Transactions on Information Theory*, Vol. 36(6), Nov. 1990, pp. 1265–1279.
- [4] V. Gupta, S. Krishnamurthy and M. Faloutsos, "Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks," in *Proceedings of MILCOM*, 2002, pp. 1118–1123.
- [5] IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification, 802.11, 1999.
- [6] P. Kyasanur and N. Vaidya, "Selfish MAC Layer Misbehavior in Wireless Networks," *IEEE Transactions on Mobile Computing*, Vol. 4(5), Sept.-Oct. 2005, pp. 502–516.
- [7] S. Radosavac, J. S. Baras and I. Koutsopoulos, "A Framework for MAC Protocol Misbehavior Detection in Wireless Networks" in *Proceedings of the 4th ACM workshop on Wireless security*, 2005, pp. 33–42.
- [8] M. Raya, J.P. Hubaux and I. Aad, "DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots," in *Proceedings of MobiSys*, June 2004, pp. 84–97.
- [9] P. K. Varshney, *Distributed Detection and Data Fusion*, Springer-Verlag, New York, 1997.
- [10] V. V. Veeravalli, T. Basar and H. V. Poor, "Minimax robust decentralized detection," *IEEE Transactions on Information Theory*, Vol. 40(1), Jan. 1994, pp. 35–40.
- [11] VINT Group, "UCB/LBNL/VINT network simulator ns (version 2)," available at: <http://www.isi.edu/nsnam/ns>.