Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service

Wenyuan Xu, Timothy Wood, Wade Trappe, Yanyong Zhang
Wireless Information Network Laboratory (WINLAB)
Rutgers, The State University of New Jersey
73 Brett Rd.
Piscataway, NJ 08854
wenyuan, twood, trappe, yyzhang@winlab.rutgers.edu

ABSTRACT

Wireless networks are built upon a shared medium that makes it easy for adversaries to launch denial of service (DoS) attacks. One form of denial of service is targeted at preventing sources from communicating. These attacks can be easily accomplished by an adversary by either bypassing MAC-layer protocols, or emitting a radio signal targeted at jamming a particular channel. In this paper we present two strategies that may be employed by wireless devices to evade a MAC/PHY-layer jamming-style wireless denial of service attack. The first strategy, channel surfing, is a form of spectral evasion that involves legitimate wireless devices changing the channel that they are operating on. The second strategy, spatial retreats, is a form of spatial evasion whereby legitimate mobile devices move away from the locality of the DoS emitter. We study both of these strategies for three broad wireless communication scenarios: two-party radio communication, an infrastructured wireless network, and an ad hoc wireless network. We evaluate several of our proposed strategies and protocols through ns-2 simulations and experiments on the Berkeley mote platform.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: [distributed networks, network communications]

General Terms

Security

Keywords

Denial of Service, Jamming, CSMA

1. INTRODUCTION

Recent advancements in wireless technologies have caused a shift in computing away from the traditional wired Internet, towards new paradigms of mobile computing. As wireless progressively becomes ubiquitous, affordable, and part of our daily lives, a plethora of security challenges will arise that were not present in the traditional network paradigm. Even now, for existing wireless networks, security is often cited as a major technical barrier that must be overcome before widespread adoption of mobile services can occur.

Security and privacy for wireless systems is complicated by the fact that wireless devices can be cheaply and easily purchased. The combination of the commodity nature of wireless technologies and an increasingly sophisticated user-base ultimately means that adversaries will be able to easily gain access to communications between wireless devices either by purchasing their own device and running it in a monitor mode, or by employing slightly more sophisticated software radios capable of monitoring a broad array of radio technologies. Further, adversaries are now empowered to easily mount a variety of security attacks, such as injecting false data into the network, launching denial of service attacks, or even disrupting the routing and delivery of legitimate data.

Many wireless security threats are being addressed through appropriately designed network security architectures [1–6]. These technologies are, essentially, modifications of traditional security services, such as confidentiality, authentication, and integrity, to the wireless domain. The wireless medium, however, introduces many threats that are not simply addressable through conventional security mechanisms. One important class of security threats that may be launched by adversaries, which are difficult to address through conventional network security techniques, are denial of service attacks. Traditionally, denial of service is concerned with filling user-domain and kernel-domain buffers [7]. However, in the wireless domain, due to the shared nature of the wireless medium, the adversary is empowered to prevent others from even communicating. An adversary can simply disregard the medium access protocol and continually transmit on a wireless channel. By doing so, he either prevents users from being able to commence with legitimate MAC operations, or introduces packet collisions that force repeated backoffs, or even jams transmissions. Such MAC and PHY-layer security threats for wireless networks have been known for some time, and recently the issue of MAC-layer weaknesses in 802.11 has been revisited by a recent announcement by the Australian CERT [8].

This paper focuses on addressing the problem of denial of service attacks targeted at the MAC or physical layer of wireless networks. At first glance, it seems that there is nothing that can be done since assuming that an adversary can continually blast on a channel grants the adversary considerable power. Nonetheless, security is a constant battle between the security expert and the clever adversary, and therefore we have chosen to take inspiration for our work

WiSe'04, October 1, 2004, Philadelphia, Pennsylvania, USA.

from Sun Tze's famous *The Art of War*:

He who can't defeat his enemy should retreat.

Translating this philosophy into the wireless domain, we propose that wireless devices employ spectral and spatial evasion strategies in order to protect against DoS.

We begin in Section 2 by presenting an overview of the denial of service problem. Here we introduce our basic assumptions as well as introduce the three different wireless network scenarios that we will study in this paper. Following the setup of the problem, in Section 3, we present protocols for detecting the denial of service. We examine *channel surfing*, our first defense against MAC/PHY-layer denial of service attacks in Section 4. Channel surfing involves valid participants changing the channel they are communicating on when a denial of service attack occurs. In Section 5, we examine *spatial retreats*, which involves legitimate network devices moving away from the adversary. We present conclusions in Section 6.

2. PROBLEM OVERVIEW

We set the stage for the problem by starting with a parable. Suppose Alice and Bob are socializing with each other at a party and, suddenly, the malicious Mr. X walks up. Without any regard for proper social etiquette, he interrupts them and begins to take over the conversation. Each time Alice tries to talk, Mr. X interrupts her and tells an inane story. Bob, likewise, doesn't fare any better. Alice and Bob both wait a polite amount of time in order to give Mr. X an opportunity to his behavior. However, after some time, it becomes clear that Mr. X will not give in and that our two heroes are destined to have a poor reunion and regret ever attending the party. The natural question that arises from this story is "What should Alice and Bob do?" One option is they could excuse themselves from the conversation and meet up with each other later in a different location. Another option is they could switch to another mode of communication, such as using hand signals or even a different language.

The story of the social party is a simple, motivating example for the problem of wireless denial of service we study in this paper. In the case of wireless communication, Alice and Bob correspond to two communicating nodes A and B, while Mr. X corresponds to an adversarial interferer X. The adversary X may interfere with A and B's ability to communicate by either ignoring MAC-layer protocols, or by emitting a signal of sufficient energy on the channel used by A and B. For example, in 802.11 networks, an adversary may employ a powerful device driver to bypass card firmware and repeatedly send out packets. As a consequence, all devices within the radio range of X will think that the channel is occupied and will defer transmission of data. Similarly, at the PHY-layer, an adversary may use any device capable of emitting energy in the frequency band corresponding to the channel A and B are communicating on. For example, it is well-known that Bluetooth devices and microwaves share spectrum with 802.11b, and hence can be used to interfere with an 802.11b network. Additionally, programmable radios and waveform generators are other devices that an adversary could use to jam the channel.

Although there are many different scenarios where a jamming-style DoS may take place, we will focus on three basic classes of wireless networks, as depicted in Figure 1:

1. **Two-Party Radio Communication:** The two-party scenario is the baseline case, in which A and B communicate with each other on a specific channel. As long as interferer X is close enough to either A or B, its transmission will interfere with the transmission and reception of packets by A and B.

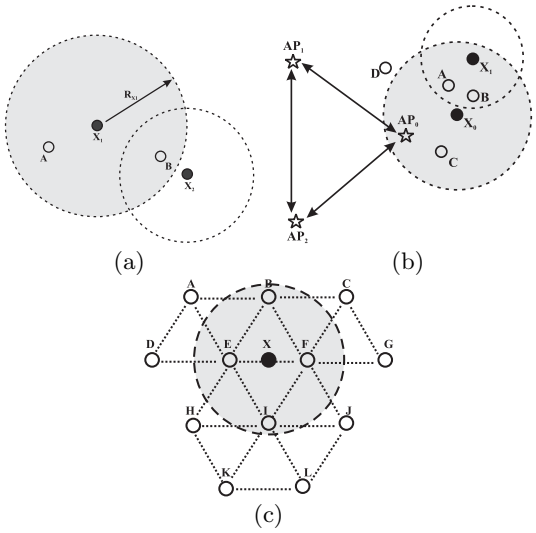


Figure 1: The three wireless communication scenarios studied in this paper: (a) two-party radio communication, (b) infrastructured wireless networks, and (c) ad hoc wireless networks. The adversary is depicted by X.

2. **Infrastructured Wireless Networks:** Infrastructured wireless networks, such as cellular networks or wireless local area networks (WLANs), consist of two main types of devices: access points and mobile devices. Access points are connected to each other via a separate, wired infrastructure. Mobile devices communicate via the access point in order to communicate with each other or the Internet. The presence of an interferer, such as X_0 or X_1, might make it impossible for nodes to communicate with their access point.

3. **Mobile Ad Hoc Wireless Networks:** Ad hoc networks involve wireless devices that establish opportunistic connections with each other in order to form a communication network. Typically, ad hoc networks employ multi-hop routing protocols in order to deliver data from one network node to another. The presence of an interferer may bring down whole regions of the network.

It should be noted that in all three cases, the adversary could act as a hidden node during the denial of service, thereby affecting only one of two communicating participants. For example, in Figure 1 (a), adversary X_2 is located to the right of B and blocks B from being able to acquire the communication channel while A might not be able to detect X_2.

3. DETECTING JAMMING-STYLE DOS

MAC-layer or PHY-layer denial of service attacks differ from other types of denial of service attacks as they are targeted at the *radio channel* while traditional denial of service attacks are targeted at a network entity's *internal memory state*. For example, many denial of service attacks can be classified as attacks in which an adversary seeks to fill a buffer residing in a network device or within an application program. The class of DoS attacks that we study in this paper involve adversaries trying to prevent devices from being able to communicate and we therefore refer to them as jamming-style DoS attacks.

Our basic problem involves an adversary controlling the communication channel, either by ignoring proper MAC-layer etiquette or by employing a radio emitter as a jammer that will interfere with successful PHY-layer decoding of legitimate packets. The differences between these two attacks suggest that we might want to perform detection either at the MAC-layer or at the PHY-layer. In this section we present several strategies that may be employed by a single wireless device to determine whether it has been blocked by a DoS.

We note that the detection techniques discussed below have a non-zero probability of a false positive. Thus, the strategies might classify legitimate high-levels of network traffic as a DoS attack. In order to differentiate between legitimate and illegitimate traffic, one could add authentication and authorization services at each node. We would argue, however, that although legitimate traffic might cause a false positive and is not strictly a DoS, such congestion is nonetheless bad for network operation and it might be desirable to employ our DoS evasion strategies to avoid congestion issues. Additionally, we note that the detection schemes discussed might declare the presence of a denial of service attack when actually a node has merely experienced a fault, or consumed all of its power resources. Introducing fault tolerance into denial of service detection is a challenging issue that will be investigated in future work.

3.1 MAC-layer Detection

In most forms of wireless medium access control, there are rules governing who can transmit at which time. For example, one popular class of medium access control protocols for wireless devices are those based on carrier sense multiple access (CSMA). CSMA is employed in Berkeley motes as well as in both infrastructure and infrastructureless (ad hoc) 802.11 networks. The MAC-layer protocol for 802.11 additionally involves an RTS/CTS handshake.

During normal operation of CSMA, when A tries to transmit a packet, it will continually sense the channel until it detects the channel is idle, after which it will wait an extra amount of time (known as the propagation delay) in order to guarantee the channel is clear. Then, if RTS/CTS is used it will send the RTS packet, or otherwise will send the data packet. Suppose we assume that the adversary X is continually blasting on a channel and that A attempts to transmit a packet. Then, since X has control of the channel, A will not pass carrier-sensing, and A may time-out or hang in the carrier-sensing phase.

Unfortunately, this time-out could have occurred for legitimate reasons, such as congestion. It is therefore important to have some mechanism to distinguish between normal and abnormal failures to access the channel. In order to differentiate between channel-access failures due to normal network behavior from those due to malicious behavior, we propose to use a thresholding mechanism based on the sensing time to discriminate between normal MAC-layer delays and abnormal delays due to a malicious adversary. The idea is, each time A wishes to transmit, it will monitor the time spent sensing the channel, and if that time is above a threshold (or if it is consistently above the threshold), it will declare that a DoS is occurring. We investigate two different approaches for determining an appropriate threshold level: a theoretically determined threshold based on a simple channel occupancy model, and an empirically determined threshold model.

3.1.1 Theoretically Setting the Threshold

For the theoretically-set threshold, we will employ a simplified model for CSMA as well as some simplifying assumptions about the underlying network traffic. We assume that

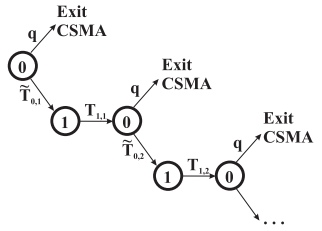


Figure 2: The events leading to exiting CSMA, and the corresponding probability parameters used in the derivation.

there are a finite set of radio sources that collectively act as an independent Poisson source with a packet generation rate of λ packets/sec. Further, we assume that transmission of packets is completed at an average rate of μ according to an exponential distribution[1].

Under these assumptions, we may model the channel as a two-state Markov model, where state 0 is the Channel-Idle state and state 1 is the Channel-Occupied state. The two-state model for the channel is simply an $M/M/1/1$ queue, and hence we may use queuing theory to derive $p_1 = P(\text{state-1})$, the steady-state probability that the channel is occupied at an arbitrary time [10]. Let $\rho = \lambda/\mu$, then at steady state $p_1 = \rho/(1+\rho)$ and $p_0 = 1/(1+\rho)$.

Now, suppose that A senses the channel at an arbitrary time t. We are interested in the amount of time D that A must, under CSMA, wait before it starts transmission. A will sense the channel until it is idle. Under popular implementations of CSMA, when the channel becomes idle, node A will continue sensing the channel for an additional τ time before transmitting in order to ensure distant transmissions have arrived at A[2]. If the channel becomes busy during that time, then A declares the channel busy and continues sensing until it experiences an idle period that lasts τ units of time.

In order to capture the distribution for D, we will introduce a few auxiliary variables. Let N denote the number of times we visit state 0 before witnessing an idle period of at least τ duration. Also, let us define the random variable $T_{0,k}$ to be the occupancy time in state 0 for the k-th visit to state 0, and similarly $T_{1,k}$ to be the occupancy time in state 1 for the k-th visit to state 1. Observe that, if we are at state 0, then to exit CSMA requires $T_{0,k} \geq \tau$, and hence we must introduce $q = P(T_{0,k} \geq \tau) = e^{-\lambda\tau}$. We may now look at the chain of events, as depicted in Figure 2. If the channel was idle when we started, then we enter at the left-most state 0, whereas if the channel was occupied when we entered, then we enter at the left-most state 1.

The probability density function for D can be shown to be

$$f_D(d) = p_0 \sum_{k=1}^{\infty} P(N=k) f_{0,k}(d) \quad (1)$$

$$+(1-p_0)\left(f_{T_{1,1}} + \sum_{k=1}^{\infty} P(N=k) f_{0,k}(d)\right) \quad (2)$$

$$= (1-p_0) f_{T_{1,1}} + \sum_{k=1}^{\infty} P(N=k) f_{0,k}(d) \quad (3)$$

[1]We note that the validity of our model depends on λ/μ, and that the approximations made become increasingly accurate as $\lambda/\mu \to 0$, c.f. [9].

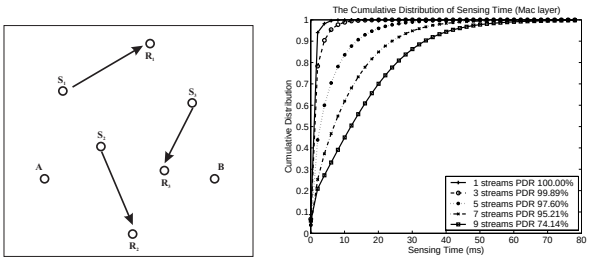
[2]Typically, the propagation delay τ is on the order of 50μseconds.

Here N is geometric with $P(N = k) = (1 - q)^{k-1}q$. The distributions $f_{0,k}(d)$ are the pdfs describing the duration contributed by exiting during the k-th visit to state 0 when we start at state 0. For example, if we exit after the first visit to state 0, then the only contribution comes from the propagation delay τ, that is, $f_{0,1}(d) = \delta(\tau)$, the point mass at $d = \tau$. As another example, let us look at the time, Y, contributed when exiting at the second visit to state 0. Since we did not exit during the first visit to state 0, Y is composed of time contributed by $T_{0,1}$ that is strictly less than τ. We will call this conditional random variable $\tilde{T}_{0,1}$. Next, Y also consists of $T_{1,1}$, and finally the τ needed to exit CSMA. Hence $Y = \tilde{T}_{0,1} + T_{1,1} + \tau$, and thus $f_{0,2}(d) = f_{\tilde{T}_0} * f_{T_1} * \delta(\tau) = f_{0,1} * f_{\tilde{T}_0} * f_{T_1}$, where $*$ denotes convolution and arises due to independence between the random variables involved, and f_{T_j} is the distribution for occupancy time for any state j. Similar recursive representations can be derived for $f_{0,k} = f_{0,k-1} * f_{\tilde{T}_0} * f_{T_1}$. From this theoretical pdf, we may find a threshold α such that $P(D \leq \alpha) = \gamma$, for a confidence level γ.

3.1.2 Empirically Setting the Threshold

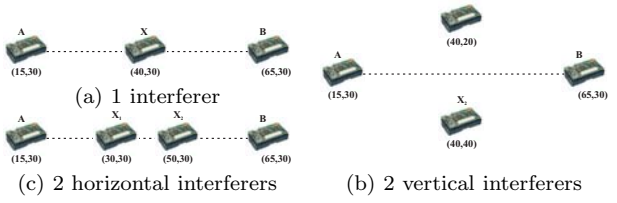
A second approach, which does not rely on constructing a mathematical model for the channel occupancy, involves each network device collecting statistics regarding the amount of time D that a device must wait before it can start transmission during normal, or even somewhat congested, network conditions. With a distribution $f_D(d)$ describing the amount of time spent in sensing before the channel becomes idle during acceptable network conditions, we may classify any new measured sensing time as either normal or anomalous by employing significance testing [11]. In this case, our null hypothesis H_0 is that the measured delay D corresponds to the distribution $f_D(d)$. If we reject the null hypothesis, then we conclude the network is experiencing a DoS. It is not desirable to falsely conclude the presence of a DoS when the network conditions are merely experiencing a glitch. Therefore, using a very low probability of false positive is desirable for determining the threshold value as it yields conservative thresholds.

In order to quantify the validity of MAC-layer DoS detection, we carried out several experimental studies using the 802.11 extensions to the ns-2 simulator. We modified ns-2 by disabling the MAC layer retransmission, so that we could focus our investigation on the channel sensing behavior. In our experiments we have two nodes, A and B, which run our DoS detection algorithm. Once every 19 msecs, node A senses the channel by trying to send out a beacon to node B. We obtained the MAC-layer delay time D by calculating the difference between the time when beacon packets reach



(a) (b)

Figure 3: The MAC-layer sensing time experiment: (a) basic underlying experimental setup, (b) cumulative distributions of D for different traffic scenarios and the corresponding packet delivery ratio.



(a) 1 interferer

(c) 2 horizontal interferers (b) 2 vertical interferers

Figure 4: The locations of the communicators and interferers. The unit of distances is centimeter.

MAC-layer and the time when the MAC successfully senses the channel as idle and sends out RTS. In order to capture the statistical behavior of the delay time, we calculated the cumulative distribution of the delay time for several scenarios involving different levels of background traffic loads. As shown in Figure 3(a), we introduced several streams (from sender S_i to receiver R_i) that are within the radio range of A and B in order to increase the interfering traffic. Each stream's traffic was chosen to represent an MPEG-4 video stream suitable for a wireless video application. We used traffic statistics corresponding to the movie Star Wars IV [12], where each sender transmitted packets with the packet size governed by an exponential distribution with a mean size of 268 bytes, and the packet inter-arrival times following an exponential distribution with mean 40msecs, resulting in each stream having an average traffic rate of 53.6Kbps. The corresponding cumulative distributions of D are shown in Figure 3(b). These observations can be explained as follows. When there are only a few streams, there are few nodes competing for channel, and node A can get the channel quickly with high probability. As the number of streams increases, the competition for channel becomes more intense, thus taking longer for A to acquire the channel.

From this figure, we can observe that when the number of streams is less than 7, the curves approach 1 quickly before D equals 40 msecs. Even in the case of 9 streams, which has an average packet delivery ratio of 74.1% and corresponds to a poor quality of service for each application, over 99% of all observed transmission delays occur within 60 msecs. However, when a DoS attack occurs, the time taken to acquire the channel is very large relative to normal MAC-sensing times, or even the times observed for poor QoS conditions. Therefore, we can select an appropriate threshold for the MAC-sensing time that corresponds to a desired confidence level. For example, if we would like to ensure, with 99% confidence, that our sensing time is a DoS and not a result of a normal background with a PDR of 75%, we should choose the threshold as 60 msecs.

3.2 PHY-layer Detection

A different strategy for detecting denial of service is to perform the detection at the physical layer. The basic idea of PHY-layer detection is to discriminate between normal and abnormal levels of ambient noise in a channel. Since most commodity radio devices do not provide signal strength or noise level measurements that are calibrated (even across devices from the same manufacturer), it is necessary for each device to employ its own empirically gathered statistics in order to make its decisions.

Each device should sample the noise levels many times during a given time interval. By gathering enough noise level measurements during a time period prior to denial of service, network devices can build a statistical model describing usual energy levels in the network. Discrimination between normal noise level measurements and noise levels due to denial of service can be done by exploiting the various features of the data. For example, χ^2-statistics or ψ^2-statistics might

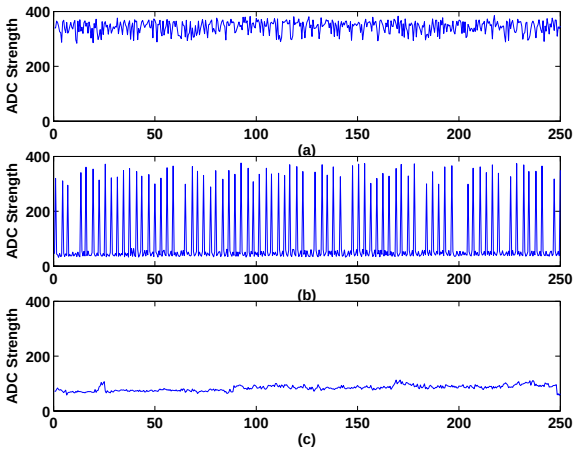


Figure 5: Raw signal strength time series for (a) the scenario with no communicator, (b) the scenario with three communicators, transmitting every 250, 300, and 350 msec respectively, and (c) the scenario with the jammer on.

be powerful tools for capturing and differentiating between time series data from benign and non-benign scenarios [13].

In order to understand the effect that a jammer would have on the received noise levels, we performed an experiment with Berkeley motes. Our baseline scenario involved a single mote A merely measuring noise levels without any other devices present. The second experiment involved three other motes (B, X_1 and X_2) transmitting packets of 31 bytes every 250, 300, and 350 msecs, and arranged as in Figure 4(b). For the third scenario, we introduced an Agilent E4438C waveform generator at $(40, 30)$ and transmitted a carrier wave at 916.7Mhz with a transmit power of -10dBm. The noise levels were monitored by employing the `RSSIADC.getData()` function on the port `TOS_ADC_CC_RSSI_PORT`. The reported values correspond to the raw values following the analog-to-digital conversion of the received voltage levels, and are in inverse relationship to power [14]. We present time series data for each of the three scenarios in Figure 5. From this figure, we find that the noise level time series with the jammer and without are distinctly different. Specifically, the measured noise levels with the jammer exhibit a much lower variation (the time series curve is almost flat) in Figure 5(c) compared to noise levels when the jammer is not present in Figures 5(a) and (b). In particular, the baseline scenario with no communicator, Figure 5(a), has a much higher variation. A more marked contrast exists in Figure 5(b), where normal traffic behavior causes the channel state to alternate between busy and idle, each period with a different duration. As a result, the time series exhibits spikes during idle periods. In contrast, the jammer kills all the channel activity, causing a low variation. These observations suggest that signal discrimination techniques can be employed to differentiate between normal operational scenarios and jammed scenarios.

4. CHANNEL SURFING

The first escape strategy that we present is channel surfing. Typically, when radio devices communicate they operate on a single channel. When an adversary comes in range and blocks the use of a specific channel, it is natural to migrate to another channel. The idea of channel surfing is motivated by a common physical layer technique known as frequency hopping. We assume throughout this section that the adversary blasts on a single channel at a time, and that the adversary cannot pretend to be a valid member of the network (i.e. the adversary does not hold any authentication keys used by the network devices).

4.1 Two-Party Radio Communication

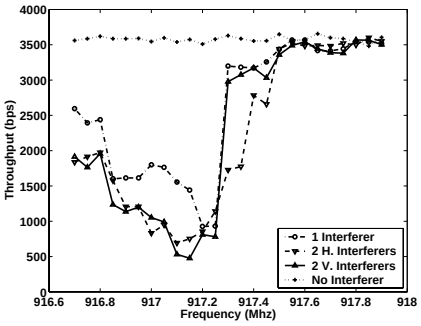
Consider the radio scenario depicted in Figure 1(a). In this scenario, adversary X_1 or X_2 has disrupted communication between A and B. We desire both A and B to change to a new channel in order to avoid X's interference.

In order to facilitate channel surfing, it is necessary to understand the interference behavior between different channels so that A and B can move to a clean channel. If the adversary is using the same radio technology as A and B to interfere, then it is important to know how many orthogonal channels are available for one to switch to. On the other hand, the adversary may not employ the same radio technology, in which case it is desirable to have some notion as to how the adversary is generating the jamming signal, and then determine an appropriate set of *safe* channels.

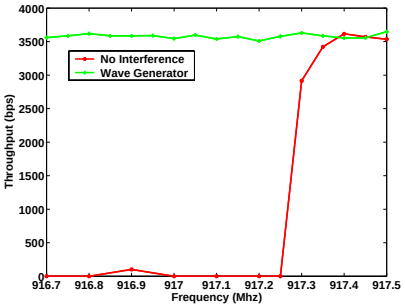
It is well known that the number of orthogonal channels provided by 802.11b is 3 (namely channels 1, 6, and 11), while 802.11a supports 12 orthogonal channels. For many wireless networks, however, the amount of orthogonal channels must be determined, or verified, experimentally. For example, the specifications for the radio employed in Berkeley motes state that a channel separation of 150kHz is recommended in order to prevent cross-channel interference. Experimentally, we have found that 800kHz is a safer value for channel separation in order to maintain *effective network-layer orthogonality*.

In order to demonstrate how one might determine channel orthogonality as well as demonstrate the feasibility of channel surfing, we conducted a set of experiments using Berkeley motes. In these experiments, two motes act as the communicator and receiver, denoted by A and B. A continuously sends out 31-byte packets to B, resulting in a throughput of 3.6Kbps. We then placed interferers or jammers in different locations. In the first set of experiments, we used motes as interferers. We tried three interferer scenarios, which are illustrated in Figures 4(a)-(c). These interferers also continuously send out packets of the same size. The interferers completely followed the default MAC protocol of Berkeley motes. All the motes transmit at the same power level. The default frequency of the motes was 916.7MHz. The results for this set of experiments are summarized in Figure 6(a). When all the motes transmit at the default frequency, the measured throughput between A and B significantly drops compared to the scenario with no interferers. Due to the fact that all devices follow the MAC protocol, the throughput did not become zero. We then incremented the transmission/reception frequency of the communicator and receiver by 50KHz in order to search for an orthogonal channel. As the frequency gap between the communicators and the interferers increases, but before it reaches a threshold, the measured throughput worsens. This is because the transmissions still interfere with each other, yet the MAC protocol is not able to coordinate transmissions across different frequencies, resulting in a much higher collision rate and a lower throughput. Finally, when the communicators increase their frequency to (or above) 917.5MHz, they are no longer interfered with. As a result, the orthogonal channels must be at least 800KHz away.

In the second set of experiments, we used a waveform generator as the jammer between the two communicators. Their positions are the same as shown in Figure 4(a). The waveform generator continuously emitted a narrow AM signal at 916.7MHz frequency and with an amplitude of -10dBm. Unlike the interferers in the above experiment, the jammer does not follow MAC rules and can completely take the



(a)



(b)

Figure 6: Experimental results indicating throughput versus channel assignment for (a) MAC-compliant Mote, (b) Jammer continuously emitting an AM signal.

channel so that the communicators do not have a chance to transmit. The results are shown in Figure 6(b). Before the communicators move out of the spectral interference range, the measured throughput is 0. As soon as the communicators move to (or above) 917.4MHz, they can transmit without any interference. The reason for a narrower gap in this scenario compared to the mote interferer cases is that the spectral width of the waveform generator's signal is narrower than the spectrum of a Berkeley motes' signal.

Using one of the DoS detection techniques discussed in Section 3 or an application-level mechanism, once A and B have detected a DoS, they will change channels. There are several strategies for changing channels. Suppose there are M total orthogonal channels and that the current channel is $C(n)$. Then a natural channel surfing strategy is to change the channel according to $C(n+1) = C(n) + 1 \pmod{M}$.

However, if we assume the adversary will periodically stop its interference and check to see whether A and B are still attempting to transmit, then the adversary will be able to detect that he is no longer interfering with their communication. Consequently, the adversary will change its channel. The adversary should successively sample each channel for a short duration T_s to see if there is any communication occurring on other channels, and then switch to that channel to resume interfering. The shorter T_s, the more likely that the adversary will miss A to B communication, while a longer T_s means that A and B can communicate longer before interference resumes.

For A and B, using $C(n+1) = C(n) + 1 \pmod{M}$ has drawbacks since an adversary will only have to sample one channel before concluding that A and B are there. In order to make scanning more difficult for the adversary, A and B should generate $C(n+1)$ pseudo-randomly by employing a shared key K. If ever a case where $C(n+1) = C(n)$ is encountered, then both A and B should proceed to the next

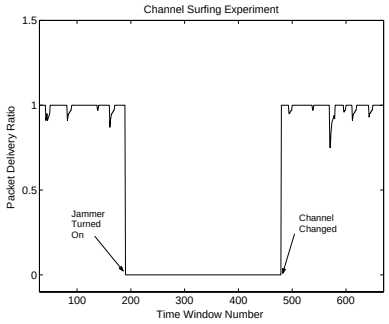


Figure 7: Packet delivery measurements from the Mote channel surfing prototype.

output of the pseudo-random generator until a new channel value is provided.

One issue that naturally arises in employing such a channel changing strategy is whether or not one should continually change channels regardless of whether the adversary is blocking the current channel. Although physical layer frequency hopping employs a strategy of constantly changing the underlying frequency, there are reasons why this might not be desirable at the link-layer. In particular, although changing the frequency of the carrier wave is easy to accomplish in the case of frequency hopping spread spectrum, changing channels at the link-layer is more involved as it requires synchronization between both parties, which necessitates additional time cost.

Prototype: We built a proof-of-concept prototype system using two Berkeley motes A and B. The application running on these two motes involved A sending out a packet to B every 200 msecs. Each packet contained a sequence number starting at 1. We partitioned the time axis into windows, and B kept track of how many packets it received in each window (n_{recv}). It can also determine how many packets A has attempted to send in each window by looking at the sequence number of the last message it receives (n_{send}). In order to capture the quality-of-service of the application, we employed the *packet delivery ratio* $r = n_{recv}/n_{send}$.

In the experiment, we used the waveform generator as the jammer X. As soon as the jammer is turned on, A cannot access the channel, and so no packets can be sent out. As a result, the packet delivery ratio becomes 0. In order for the application to survive the DoS attack, both A and B should incorporate a DoS detection and defense strategy. Our prototype detection algorithm works as follows. At the application level, each mote sets a DoS check timer (30 seconds). Each time the timer expires, it attempts to send out a beacon by making a `SendMsg.Send()` call. The send call will return SUCCESS if the channel monitor component identifies an idle period so that the message send can start. (Later on, a notification will be sent to the application after the MAC-layer ACK is received from the receiver.) If the channel monitor component cannot sense the channel as idle after a long time (e.g., by using the threshold we obtain from the empirical study in Section 3), the send call will return FAILURE. After it returns FAILURE, the mote can conclude that it is under a DoS attack. As soon as the attack is detected, the mote will change its frequency to an orthogonal channel (e.g, from 916.7MHz to 917.6MHz). In order to avoid collision, A and B should not send beacons at the same time. The code is included below:

```
task void checkDoS(){
    sent = call SendMsg.send(TOS_BCAST_ADDR,
                             sizeof(uint16_t),
                             &beacon_packet);
```

```
Algorithm: Infrastructured Channel Surfing
if DETECT_DOS(Self)==TRUE then
    Change_Channel()
else
    if AMI_AP == True then
        Calc_ChildrenLastCalledHome()
        Calc_NegligentChildren()
        RESPONSES = Probe_NegligentChildren()
        if Any(RESPONSES == NULL) then
            Broadcast_ChangeChannelCommand()
            Change_Channel()
        end
    else
        ListenForBeacons()
        if TimeToLastBeacon() > BIG then
            Change_Channel()
        end
    end
end
```

Algorithm 1: Channel surfing for wireless infrastructured networks. This algorithm runs on each network device.

```
    if(!sent){
        if(failures++ < thresh)
            post checkDoS();
        else post changeChan();
     }else{
        failures = 0;
    }
}
```

After both A and B change their frequencies, they can resume their application behavior, and the packet delivery ratio will go up again. We present measurements for the prototype channel surfing experiment in Figure 7. The DoS detection and channel surfing strategies works as expected.

4.2 Infrastructured Network

Now consider an infrastructured wireless network as depicted in Figure 1(b). Here, we have an access point AP_0, which has four wireless devices A, B, C, and D connected to it. There are two main scenarios for a denial of service against the access point, corresponding to adversaries X_0 and X_1. In the first scenario, adversary X_0 interferes with AP_0, A, B and C, but does not interfere with node D since it is outside of X_0's radio range. In the second scenario, adversary X_1 interferes with A and B, but not with AP_0 or any of the other nodes. The main difference between these two scenarios lies in the fact that one has the access point blocked by the adversary, while the other does not.

We need a strategy for changing channels whereby *all* nodes connected to the access point will change channels with the access point. We do not want to have scenarios where some devices are on the old channel while some are on the new channel. Algorithm 1 presents the sequence of events needed for a network device to determine whether to change channels.

Periodically, the algorithm checks to see whether the device has been blocked from communicating by an adversary. This can be done using the methods described in Section 3. If a DoS has been detected, then the device will change channels. Otherwise, the device checks to see whether it is an access point. Access points will examine their list of children to see which devices have not communicated recently. Those devices whose last communication with the AP was greater than some threshold amount of time will be probed by the AP to ascertain whether they have left the channel due to a DoS. If any device does not respond to their probe, the AP will conclude that the device has disappeared due to a DoS.

```
Algorithm: Basic Ad Hoc Channel Surfing
if DETECT_DOS(Self)==TRUE then
    Change_Channel()
else
    RESPONSES = Random_ProbeNeighbors()
    if Any(RESPONSES == NULL) then
        STRANDED = Test_NextChannel()
        if Any(STRANDED)==TRUE then
            Broadcast_ChangeChannelCommand()
            Change_Channel()
        end
    end
end
```

Algorithm 2: Basic channel surfing for wireless ad hoc networks.

The rationale behind this is that, during normal operation of an infrastructured wireless network, if a network node wishes to leave the network it will perform a disassociation request, allowing the AP to free up any resources allocated to managing that network device. Further, when a network node moves to another access point, the device will perform reassociation with the new access point. The new access point will relay this information to the old access point, and acquire any data that might be buffered at the old access point. In both cases, the access point will know when a user legitimately leaves its domain.

If the AP concludes that the device disappeared due to a DoS, then it will broadcast an emergency change channel packet that is signed by the AP's private key. This packet can be authenticated by each of the AP's children that are not blocked by the adversary. Following the issuance of the change channel command, the AP will change its channel and commence beaconing on the new channel in hopes to elicit associations from its children.

If the device is not an access point, then it will check to see if it has not heard its access point's beacons in a long time. It may even probe the access point. Either way, the child device will decide that it needs to catch up with its parent and change channels. In Algorithm 1, the default condition is to remain on the same channel.

4.3 Ad Hoc Network

When an adversary performs a DoS on an ad hoc network, he severs many of the links between network devices and can possibly cause network partitioning. Channel surfing can counteract such network faults by having the network or regions of the network switch to a new channel and re-establish network connectivity. In order to use channel surfing to address DoS for ad hoc networks, we assume that each network device keeps a neighbor list. However, since we are operating in ad hoc mode, we do not assume that if a device moves that it will inform its neighbors of its intent to relocate. Also, during unhindered network operation, we assume that no network partitions arise due to network mobility.

Algorithm 2 presents an outline of the channel surfing operations that run on each device. First, devices check to see if they have been blocked by a DoS. If so, they change channels and monitor the new channel to assist in reforming the ad hoc network on the new channel.

If a device has not been blocked by a DoS, then there is a chance that its neighbors have been blocked. Therefore, at random times a node will probe its neighbors to see that they are still nearby. There are several reasons that a neighbor node might not be present: it might have moved to a different part of the network, or it might have been blocked by a DoS. The device checks to see if a DoS has occurred by

```
Algorithm: Dual-Radio Ad Hoc Channel Surfing
if DETECT_DOS(Self)==TRUE then
    Change_Channel()
    InformNeighbors()
    EstablishNewLinks()
end
```

Algorithm 3: Dual-radio channel surfing for wireless ad hoc networks.

testing to see if any devices are stranded on the next channel. If the test returns positive, then the device returns to the original channel and broadcasts a signed change channel command to its neighbors, which is flooded through the rest of the network. It will then change channels and assist in reforming the ad hoc network on the new channel. Other devices will authenticate the command, and switch channels. If the test returned negative, then the device will assume that its absent neighbor has merely migrated to a different portion of the network and remove it from the neighbor list.

One unfortunate drawback of channel surfing for ad hoc networks is that it requires the use of flooding messages to promptly initiate a channel change across the entire ad hoc network. A simpler and more efficient alternative channel surfing strategy is possible if the network devices employ a dual-radio interface, that is they are capable of operating on two channels simultaneously.

Algorithm 3 describes the operations that run on each network device when a dual-radio interface is available. The default operation of the ad hoc network is for devices to employ one radio channel for communication, yet monitor both channels. When a device detects that it has been blocked, it will switch to the next channel. Once on the new channel, the device will contact its neighbors via the new channel to inform them of its new channel policy, warn of possible DoS, and establish new links in order to maintain network connectivity. In addition to the usual routing information, each network device must maintain an additional channel assignment field for each of its neighbors. The end result is that a network will consist of some links on the old channel and some links on the new channel, as depicted in Figure 8.

5. SPATIAL RETREATS

The second escape strategy that we propose is *spatial retreats*. The rationale behind this strategy is that when mobile nodes are interfered with, they should simply move to a safe location. Spatial retreat is often a desirable defense strategy to employ since most wireless networks involve mobile participants, such as users with cell phones or WLAN-enabled laptops. The key to the success of this strategy is to decide where the participants should move and how should they coordinate their movements. In this paper, the discussion on spatial retreats primarily focuses on a simple adversarial scenario in which the adversary is stationary. This adversarial model might arise in cases where the adversary is unknowingly or unintentionally jamming the communication. More powerful adversarial models where the adversary is mobile and can stalk the communicating devices is currently being investigated and will be presented in subsequent work.

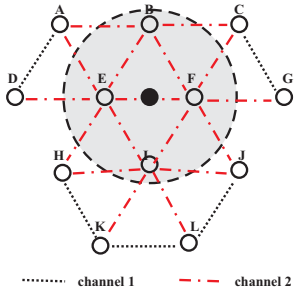


Figure 8: Channel surfing for an ad hoc network consisting of dual radio devices.

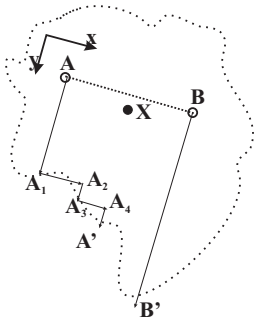


Figure 9: The spatial retreat strategy for a two-party communication scenario. The region depicted by the dotted line is the interference range of the adversary.

5.1 Two-Party Radio Communication

Let us again start by examining the two-party communication scenario. We present an example jamming scenario in Figure 9, where the adversary X interferes with both A and B so that these two nodes cannot communicate with each other. In a spatial retreat, as soon as the communicating parties (i.e., A and B) detect the DoS scenario, they try to move away from the adversary. It is a daunting task, however, to decide on a retreat plan as both parties must agree on the direction of the retreat and how far to retreat. This task is complicated by the fact that A and B cannot communicate with each other while they are within the adversary's broadcast radio range. Further, even after they leave the adversary's radio range, they may not remain within each other's radio range due to the lack of synchronization between them and the irregularity of the interference region.

Considering the above factors, any functional retreat plan must satisfy the following two conditions: (1) it must ensure that both parties leave the adversary's interference range; and (2) it must ensure that the two parties stay within each other's radio range. In order to accomplish these two requirements, we propose a three-stage protocol:

1. *Establish Local coordinates:* We assume the two parties know each other's initial position prior to the introduction of the adversary. This assumption is reasonable since it is becoming increasingly popular to incorporate positioning capabilities in mobile devices [15]. Using both parties' positions, we can decide on a local coordinate system (for example, we may define the x axis of our local coordinate system to be aligned with the segment $\overline{AB}$, as shown in Figure 9, and determine the y axis accordingly).

2. *Exit the Interference Region:* After the local coordinates are established, both parties move along the y-axis. While they move, they periodically check the interference level using the techniques discussed in Section 3. As soon as a node detects that it is out of the interference range, it stops moving. We would like to emphasize that, in practice, the two parties will stop asynchronously (as shown in Figure 9) because the ra-

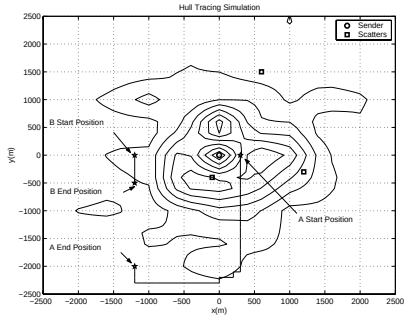


Figure 10: Simulated hull tracing scenario in which B is the Master and A is the Slave. Upon escaping the radio region of the adversary, A seeks to get within 1000 meters of B.

dio range of the adversary is irregular in shape and the two parties cannot talk to each other before they move out of the range. In the example, A stops at location A_1, while B stops at B'.

3. *Move Into Radio Range:* There is a possibility, after exiting the adversary's radio range, that the two parties will be outside of each other's radio range, as shown in Figure 9. In this scenario, the two parties must move closer to each other so that they can resume their communication. If we let both parties move around, then they may not be able to find each other. Rather than giving both nodes the freedom to move in the third phase, we propose that one entity act as the Master, who will remain stationary, while the other entity acts as the Slave, who will move in search of the Master. In our figure, B is the Master and stays at B' while A moves to find B. Since every node knows the other node's initial location, A can move along the x-axis to approach B. One issue that comes up is that A may enter the interference range while searching for B. As soon as A detects that it is in the interference range, it must stop moving along the x-axis and return to moving along the y-axis to exit the interference range. While moving along the x-axis, A will not move beyond B's x-position, and if A's x-coordinate ever equals B's, A will move towards B directly.

This protocol achieves the two necessary conditions, and can easily be modified to handle scenarios where only one node is blocked by the interferer.

We studied the behavior of the proposed spatial retreat strategy by conducting a simulated radio communication scenario involving an adversary emitting a jamming signal in the 916.7 Mhz unlicensed band. The two entities A and B were initially located at $(300, 0)$ and $(-1200, 0)$ meters respectively. In order to capture a realistic non-isotropic radio pattern for the interferer, we placed three scatterers at $(600, 1500)$, $(1200, -300)$, and $(-100, -400)$ meters. The radio environment was simulated through ray tracing [16]. The scatterers were assumed to introduce random phase shifting in the transmitted signal. The hull tracing algorithm presented above was employed, with entity B acting as the Master while entity A acted as the Slave. Both A and B were assumed to know each other's initial position, and that they could measure the energy emitted by the adversary. We present the results of the simulation in Figure 10. In this figure, we have presented contours of equal energy for different (x, y) locations relative to the adversary. The paths taken by entities A and B are depicted. As both A and B escape the radio region of the adversary, A moves too far from B and cannot maintain radio connectivity. Therefore, A performs the *Move Into Radio Range* portion of the procedure.

5.2 Infrastructured Network

In the infrastructured scenario, there are several access points $AP_0, AP_1, \cdots, AP_N$ that are connected via a backbone. Wireless devices, R_j, connect to access points and perform communication between themselves (or with devices on the Internet) via routing through the APs.

As noted earlier, during a DoS in the infrastructured network, the adversary can either block the access point from the receivers, or block the receivers from communicating with the access point, or do both. A spatial retreat for an infrastructured wireless network must be a strategy that allows the user R_j to survive all three situations. The basic idea of spatial retreat in this context is that a mobile device will move to a new access point and reconnect to the network under its new access point. We note that it is not necessary for the access point to participate in a spatial retreat as access points are typically fixed infrastructure and not usually capable of mobility.

All three situations described above can be detected by an appropriate DoS detection strategy, as discussed in Section 3. In order to perform a spatial retreat for an infrastructured wireless network, we assume that each mobile device has an *Emergency Access Point List* assigned to it, and that the mobile device knows how to move in order to reach its Emergency Access Point. The Emergency Access Point can be assigned to each device by its current Access Point prior to the denial of service.

When a device R_j detects a denial of service (either it cannot communicate, or it cannot receive beacons from its access point), it will begin to move towards its Emergency Access Point. While moving, it will occasionally pause and attempt to re-establish communication with its home access point. This is done in order to avoid any unnecessary handovers to other access points, and arises in scenarios where the adversary only blocks the user and not the access point. However, if R_j is not able to re-establish communication with its original AP, it will continue to move towards its Emergency Access Point. When R_j receives beacons from the new access point, it will initiate access point handoff. The purpose of access point handoff is to perform mutual authentication and establish authorization to use the new access point's services. There are many variations of authenticated handoff that can be employed, such as [17, 18].

We note that one problem that might result from spatial retreats in the infrastructured network is that all the mobile devices under an AP might move to the same Emergency AP. In order to prevent the resulting congestion at other APs, it is wise for the current AP to assign the Emergency Access Point lists in such a way as to divide the load across all of its neighboring APs.

5.3 Ad Hoc Network

It is much harder to design a spatial retreat strategy for ad hoc network scenarios because each node is not only involved in the communication it initiates but is also involved in forwarding packets. For ad hoc networks, it is critical to maintain network connectivity and if a node must leave its original position as a response to DoS attacks, it should move to a new location that minimizes degradation to network connectivity. For this preliminary study, we assume that only those nodes who are interfered with by the DoS attack need to escape, while other nodes should stay where they are. A globally optimized topology reformation strategy is beyond the scope of this paper [19].

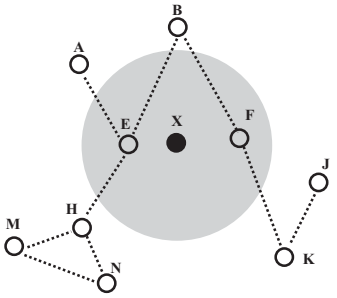


Figure 11: Scenarios for spatial retreat strategies in an ad hoc network setting. The adversary is marked by X.

Figure 11 illustrates a spatial retreat scenario in an ad hoc network setting, which attempts to minimize the network connectivity degradation. In Figure 11, node E originally connects to nodes A, B, and H, thus participating in flows $\overrightarrow{AEB}$, $\overrightarrow{AEH}$, and $\overrightarrow{HEB}$. After adversary X starts jamming the channel, E decides to move away. As shown in the figure, it is impossible for E to find a new location where it can avoid X but still maintain connection to A, B, and H. It has two choices: (1) move closer to A and B, or (2) move closer to A and H. (It cannot maintain $\overrightarrow{HEB}$ any more.) It compares these two options, and chooses the one which leads to a smaller loss in local network behavior by trying to maintain the local flow with the most value. Suppose the local flow $\overrightarrow{AEB}$ had a much higher traffic rate than the local flow $\overrightarrow{AEH}$. In this scenario, E decides to move to a new location between A and B in order to maintain the high-valued flow $\overrightarrow{AEB}$. In the other example shown in the same figure, F connected B and K before jamming occurs. No matter where F moves to, it cannot connect both B and K. Network partitioning cannot be avoided in this case, and F should move to a location away from the adversary where it will be able serve as the endpoint for the most traffic.

We assumed that every node knows the location of its neighbors. This assumption can be realized by using equipment such as GPS. In addition, every node must keep track of the traffic rate of each stream it connects. Following a DoS attack, each node will escape to a location where it can avoid the adversary, and continue to serve as much traffic as possible. Tracking the traffic rates is not an expensive operation, and can be accomplished by adding an additional column to the neighbor table for recording traffic measurements. This does not incur noticeable energy or memory overhead.

6. CONCLUSION

Due to the shared nature of the wireless medium, it is an easy feat for adversaries to perform a jamming-style denial of service against wireless networks by either continuously sending packets ignoring the MAC-layer protocols or just emitting jamming signals. In this paper, we have presented two different strategies that may be employed to mitigate the effects of this type of DoS attacks. The rationale behind both strategies is that legitimate wireless users should avoid the interference as much as possible because there is no way to combat the adversary. The first strategy involves changing the transmission frequency to a range where there is no interference from the adversary. The second strategy involves wireless users moving to a new location where there is no interference. We examined both strategies for three general classes of wireless networks: a generic two-party radio communication scenario, infrastructured wireless networks, and multi-hop ad hoc networks.

Additionally, we proposed two approaches that a single node may employ to effectively detecting a DoS attack. The first operates at the MAC layer by monitoring the sensing time before a channel becomes idle and the other operates at the PHY layer by observing the noise levels in the channel. Both these statistics will exhibit distinct behaviors for normal network scenarios and DoS attacked scenarios. We have validated these detection strategies using both simulations and experimental studies.

Acknowledgements: The authors would like to acknowledge Badri Nath, Rich Howard, Kishore Ramchandran, Zang Li, and Ivan Seskar for valuable discussions during this project.

7. REFERENCES

[1] B. Potter, "Wireless security's future," *IEEE Security and Privacy Magazine*, vol. 1, no. 4, pp. 68–72, 2003.

[2] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[3] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *8th ACM International Conference on Mobile Computing and Networking*, September 2002.

[4] P. Papadimittratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulations Conference (CNDS 2002), San Antonio*, 2002.

[5] J. Kong, H. Luo, K. Xu, D. Gu, M. Gerla, and S. Lu, "Adaptive security for multi-layer ad-hoc networks," *Special Issue of Wireless Communications and Mobile Computing*, 2002.

[6] Y.C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proceedings of IEEE Infocom 2003*, 2003, pp. 1976–1986.

[7] Q. Huang, H. Kobayashi, and B. Liu, "Modeling of distributed denial of service attacks in wireless networks," 2003, vol. 1, pp. 41–44.

[8] AusCERT, "Aa-2004.02 - denial of service vulnerability in ieee 802.11 wireless devices," http://www.auscert.org.

[9] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. on Communications*, vol. 23, no. 12, pp. 1400 – 1416, 1975.

[10] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*, John Wiley & Sons, 1976.

[11] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer Verlag, 2nd edition, 1994.

[12] F.H.P. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, November/December 2002.

[13] B. Kedem, *Time Series Analysis by Higher Order Crossings*, IEEE Press, 1994.

[14] Chipcon, "Chipcon cc1000 radio's datasheet," http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf.

[15] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, August 2000.

[16] Andrea Goldsmith, "Stanford University EE 359 Wireless Communications Course Notes," http://www.stanford.edu/class/ee359/.

[17] S. Pack and Y. Choi, "Pre-authenticated fast handoff in a public wireless lan based on ieee 802.1x model," in *Proceedings of the IFIP TC6/WG6.8 Working Conference on Personal Wireless Communications*. 2002, pp. 175–182, Kluwer, B.V.

[18] X. Fu, T. Chen, A. Festag, H. Karl, G. Schäfer, and C. Fan, "Secure, QoS-enabled mobility support for IP-based networks," in *Proc. IP Based Cellular Network Conference (IPCN)*, Paris, France, 2003.

[19] A. Wood, J. Stankovic, and S. Son, "JAM: A jammed-area mapping service for sensor networks," 2003, pp. 286 – 297.