Using Channel Hopping to Increase 802.11 Resilience to Jamming Attacks

Vishnu Navda, Aniruddha Bohra, Samrat Ganguly
NEC Laboratories America
{vnavda,bohra,samrat}@nec-labs.com

Dan Rubenstein
Columbia University
danr@cs.columbia.edu

***Abstract*—802.11a, b, and g standards were designed for deployment in cooperative environments, and hence do not include mechanisms to protect from jamming attacks. In this paper, we explore how to protect 802.11 networks from jamming attacks by having the legitimate transmission hop among channels to hide the transmission from the jammer. Using a combination of mathematical analysis and prototype experimentation in an 802.11a environment, we explore how much throughput can be maintained in comparison to the maintainable throughput in a cooperative, jam-free environment. Our experimental and analytical results show that in today's conventional 802.11a networks, we can achieve up to 60% of the original throughput. Our mathematical analysis allows us to extrapolate the throughput that can be maintained when the constraint on the number of orthogonal channels used for both legitimate communication and for jamming is relaxed.**

I. Introduction

802.11 protocols were designed under the assumption that all nodes are interested in transmission of data, and follow the rules of the protocol regarding when to send and when to permit other nodes to send. These protocols were not designed to account for jammers who would seek to interrupt transmissions. As these networks become more prevalent and society becomes more dependent on their use, the potential for their jamming will increase, whether one is simply out for kicks, or for advantage in a military or business setting.

This paper explores how to utilize channel hopping to make 802.11 networks resilient to jamming attacks. While Frequency Hopping Spread Spectrum (FHSS) was available in the original 802.11 standard [1], it was not incorporated into the subsequent, more popular 802.11a,b and g protocols that are our focus.

Our goal is to see how much more resilient we can make 802.11 protocols to jamming attacks. Obviously, resistance to a jammer is not always possible since, in the wireless domain, a jammer with unlimited resources can always successfully jam any wireless transmission by flooding the entire spectrum that could be used by the client. However, there are instances where increased resilience would be beneficial, especially against more naive or resource-constrained jam attacks. Often the jammer may be restricted to a configuration similar to that of a legitimate client. For instance, in many situations, such as in a secure group meeting or in an open field, the equipment used to jam would be visible to the legitimate participants. To remain inconspicuous, the jammer would need to jam with conventional (802.11) hardware such as a single laptop with one or two wireless interfaces. The software can be modified, but under the hardware constraints, the jammer has limited ability to flood the spectrum, which makes mounting a defense possible.

The jammer's protocol and the legitimate communication's channel hopping protocol are designed via a Stackelberg competition, where the legitimate communication is the leader and the jammer is the follower. This means that the jammer knows all the non-randomized details of the legitimate communication's resilience techniques, and can design its jamming protocol taking into account that knowledge. Our objective is then to design the channel hopping protocol that maximizes the legitimate communication's throughput in the presence of this knowledgeable jammer.

In our Stackelberg competition, the channel hopping sequence used by the legitimate communication is a pseudo-random sequence whose sequence of channels is not known to the jammer. A smart jammer who can only transmit and receive on a single channel at a time and has no knowledge of the specific channel hopping sequence must scan channels for legitimate communications. Upon finding an active channel, the jammer transmits back to back packets on that channel to interfere with the legitimate communication until the legitimate communication hops to a new channel. The jammer then repeats this procedure again by scanning for an active channel.

Our experiments reveal that changing channel in conventional hardware takes a few milliseconds. Hence, when the legitimate communication moves to a new channel, it has a short period of time to transmit data while the jammer spends time hopping from channel to channel in search of the legitimate communication. The time between channel hops must be strategically chosen by the legitimate communication: if the times are too small, then the high frequency of outages will reduce throughput. If they are too large, then the jammer will find and successfully jam large fractions of the transmission.

We provide mathematical analysis for the determina-

tion of optimal channel hopping time given the number of available channels. We implement our channel hopping mechanism and optimally configure the channel-scanning jammers using off-the-shelf 802.11 wireless network adapters. Our experimental results show how combinations of jamming and channel hopping impacts the achievable throughput.

II. Problem Formulation

In this paper we consider the communication between an AP and a *legitimate* client over an 802.11 based wireless network. We refer to this communication as a *legitimate* communication to distinguish it from the transmissions sent by the jammer. Let L represent the *number of channels* that can be utilized for this communication. For instance, in 802.11a $L = 12$ when only non-overlapping channels are used for communication.

The legitimate communication may change channels over time. When such a channel change occurs, we say that the communication *hops* between channels. We refer to the time that the legitimate communication stays fixed in a particular channel as the *residence time*.

Even in the absence of a jammer, when the legitimate communication hops channels, there is an *outage* period whose time we represent by τ. The outage is due to the time it takes to reconfigure the card to transmit on a new channel, and also any lack of synchrony between the times at which the AP and client respectively switch channels. During the outage, the throughput drops to zero.

A separate *jammer* client attempts to block the legitimate communication. As mentioned in the introduction, we assume the jammer's configuration is identical to that of the legitimate client.

A. Channel Hopping

If the legitimate client and AP were to communicate on a static channel, the jammer could scan through all channels and identify the channel used for communication. It could then proceed to block this channel indefinitely. Clearly, if the legitimate communication wishes to continue, it must hop to a new channel.

One approach, as is taken in [2], [3], is *reactive*: the legitimate communication hops channels only after the legitimate participants have identified that the current channel they are using is being jammed. A second approach, taken by us, is to *proactively* hop channels without attempting to verify the status of the channels being hopped from or hopped to. The benefit of reactive hopping is the minimization of the number of hops per unit time. This is done by reactive switching. Proactive switching will likely hop more often than is necessary. However, the benefit of proactive hopping is the lack of a need for the legitimate communication to detect the presence of a jammer. Obtaining an accurate estimate of a channel's status in a short period of time is not easy. For example, the DOMINO system [4] as reported requires several seconds to make an accurate determination of a "greedy station". In [5], the accuracy of the method is not evaluated with respect to the time it is allowed to sample the network traffic, as the concern is identifying selfish users in a steady-state system.

B. Pseudo-random Channel Hopping Sequence

In order to implement channel hopping in a manner that is secure from the jammer and minimize the throughput loss, we assume that the channel hopping sequence is pseudo-random, and that this sequence is known by only the AP and the legitimate client. For instance, the AP can decide the method for generating the pseudo-random sequence and using public key encryption, encrypt the description of the method using the client's public key. Methods for generating and exchanging secure pseudo-random sequences are well known [6] and NIST offers several possible standards that can be used [7]. If such pseudo-random channel hopping sequences are applied, even if the jammer knows the past history of channels used, its ability to determine the next channel in the sequence is no better than a random guess.

C. Jammer Behavior

We assume that the jammer is aware of our channel hopping protocol and will do as much as it can, given its limited set of hardware resources, to interrupt the legitimate communication. Since the hopping sequence follows a random progression that is unknown to the jammer, the best remaining jam strategy is to quickly scan the entire spectrum of channels looking for the legitimate communication to jam. When a channel is found containing this communication (e.g., a packet with MAC addresses of nodes whose communication the jammer wishes to disrupt), the jammer occupies the channel completely by transmitting back to back packets that interfere with this other communication.

Our results about the resilience of our hopping solution depend upon our claim that we examine the "best" jammer against such a solution. Our work does make the following assumptions about our jammer:

- We assume (conservatively) that when the jammer transmits, it does in fact block the legitimate communication. Packet capture [8], [9] and hidden terminal [10] phenomena may reduce the effectiveness of a jammer whose geographical position is not ideal for jamming.
- The jammer must periodically stop jamming and listen for a brief period of time on the channel to determine if the legitimate communication has already hopped to another channel.

III. Analysis

In this section, we provide analysis of the optimal residence time. We derive formulas that determine the

optimal time as a function of the number of channels and the maximum hopping rate of conventional 802.11 cards (used by both the legitimate client and the jammer).

Let τ be the time required to hop, and that no data can be sent while the hop is occurring. If the channel is switched after transmitting on the current channel for $s*\tau$ time units, without an jammer, the throughput is clearly a fraction $f(s) = s/(s+1)$ of what it would be without hopping.

Suppose legitimate communication utilizes a single channel at a time, and the jammer can listen to and jam a single channel at a time, and suppose the jammer also requires time τ to hop channels. Furthermore, suppose the jammer can determine instantaneously when it has hopped to a channel on which legitimate communication exists and immediately jams this channel. Furthermore, let us assume that the jammer knows precisely when the channel ceases to be used by the legitimate communication (i.e., the client and AP are in the process of hopping). Last, to simplify presentation, we normalize the throughput rate without channel hopping to one. Hence, any throughput rates stated below indicate the fractional rate with respect to no channel hopping.

In this model, the jammer can "check" j channels in time τj. Since there is a total of L channels, the probability that the jammer has hit the right (pseudo-randomly selected) channel after checking j channels is j/L for all $0 \leq j \leq L$.

Hence, if the legitimate communication has a residence time of $s\tau$ on a channel (putting time $(s+1)\tau$ between hops), with $s < L$, the expected throughput is $\frac{\sum_{j=1}^{s}(1-j/L)}{s+1}$ which reduces to

$$g(s) \quad = \quad \frac{s}{s+1} - \frac{s}{2L}. \qquad (1)$$

For a poor choice $s \geq L$, the throughput is $g(s) = \frac{L-1}{2(s+1)}$.

Note a simple generalization, where it takes the jammer time τ to switch channels and identify the channel's use by the legitimate connection, but the legitimate communication takes time $\alpha\tau$ to complete its hop. α can increase due to imperfect synchronization between the AP and client, but may also decrease due to the time required by the jammer to identify traffic on the channel it has just moved to. When $s < L$, Equation (1) then extends to:

$$g(s,\alpha) \quad = \quad \frac{2Ls - s(s+1)}{2L(s+\alpha)}. \qquad (2)$$

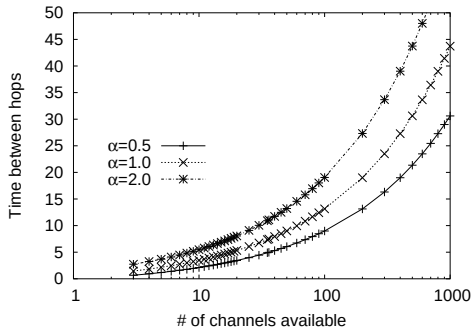
For $s \geq L$, the throughput is $\frac{L-1}{2(s+\alpha)}$.

The optimal residence time is easily determined solving the derivative with respect to s equal to 0, yielding $s = \sqrt{\alpha^2 + (2L-1)\alpha} - 1$. For $\alpha = 1$, this reduces to $s = \sqrt{2L} - 1$.

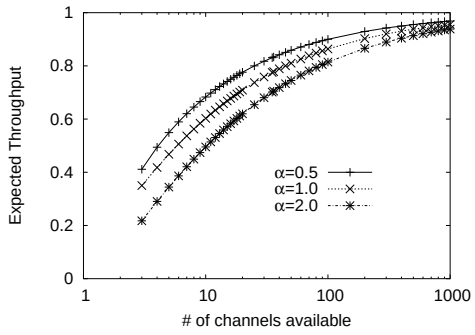
The maximum attainable throughput for general α can be obtained by plugging in the above value for s into (2). For $\alpha = 1$, this reduces to

$$1 - \frac{2\sqrt{2L} - 1}{2L}.$$

For example, when $\alpha = 1$, $L = 3$ (e.g., 802.11b orthogonal) yields a throughput of approximately 0.35, and when $L = 12$ (e.g., 802.11a orthogonal), the throughput is approximately 0.63.



(a) Optimal time between hopping



(b) Throughput

Fig. 1. Optimal wait time and expected throughput with jammer sequencing through channels

Figures 1(a) and 1(b) respectively plot the optimal residence time between hopping and optimal throughput as the number of channels, L, is varied along the x-axis. The different curves plot these values for different values of α. Note that both optimal time and optimal throughput are relatively insensitive to changes in α. In Figure 1(a), the time is in multiples of the time it takes the card to switch channels. Noting that the x-axis is log-scale, the time is sub-linear in the number of channels. We see that for conventional 802.11, the client must switch at a rather high rate and cannot spend significant time on a channel. In Figure 1(b), the expected throughput is normalized to what can be achieved in a jam-free setting without hopping. Conventional 802.11 standards see significant throughput degradation. For instance, if only the 3 orthogonal channels in 802.11b are used, when

$\alpha = 1$, one can expect less than half of the throughput when there is no hopping and no jamming. While this is undesirable, the results are rather impressive, considering there are only 3 channels to hop upon. With 11 channels, we can achieve over sixty percent of the throughput when there is no jammer, and with 36 channels, almost 80% of the jam-free throughput.

IV. Design and Implementation

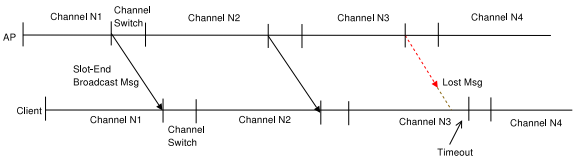


Fig. 2. Channel Switching Protocol.

Our prototype setup consists of three nodes - a client, an AP and a jammer node. Each node is a dell laptop that is equipped with an Atheros a/b/g wireless minipci card and it runs Linux-2.4.26 kernel. We modified the open source Madwifi wireless card driver to implement the channel hopping protocol and used it on the AP and client nodes. We implemented the jamming protocol as a user level program and we did not modify the Madwifi driver at the jammer node.

In our system, both AP and client use the same algorithm to generate a pseudo-random sequence. The generated random number is mapped to a channel number using a modulo operation. To maintain synchronous channel hopping the AP broadcasts a special End-of-Slot message to signal the end of its residence in the current channel. On receiving the End-of-Slot message, the client immediately switches to the new channel. Figure 2 depicts an instance of our channel hopping protocol in action. In this example, the first two channel hops are initiated at the client on receiving the End-of-Slot message. For the third hop, this message is lost due to bad channel conditions or due to interference caused by the jammer. As a result, a timeout is triggered at the end of the residence time, which initiates a channel switch operation at the client.

Our experimental jammer listens on a channel for an amount of time bounded by a constant that we call the *Listen Interval* to detect ongoing traffic. If the jammer receives a packet containing the MAC address of a participant in the legitimate connection it seeks to jam, it has found its channel to jam. The jammer immediately starts sending large broadcast packets as fast as possible for a fixed amount of time that we call the *Jam Interval*. We can ensure that these packets are always sent out no matter what the state of the channel is by disabling carrier sensing at the jammer. We did not modify carrier sensing functionality as we discovered during our experiments that our simple scheme is effective in blocking the legitimate communication completely. After the jam interval is complete, the jammer returns to its listening state on the same channel, and the process repeats. The goal is to create interference in the channel that is utilized by a legitimate communication for as much time as possible.

V. Evaluation

In this section, we evaluate the performance observed by a client with various configurations of channel hopping both in presence and in absence of a jammer.

A. Experimental Setup

In our experiments, we restrict the channels used by the legitimate communication to a set of 12 non-overlapping channels in the 802.11a, 5 GHz band. Our jammer is aware of this limited spectrum utilization, and therefore only needs to do a sweep of these 12 channels in order to detect the channel used by the legitimate communication. We note that this restriction favors the jammer, as there are fewer channels to scan and the jammer is assured of a direct hit when it locates the channel with traffic. Unless specified otherwise, the residence time in a channel for the clients and the AP is 100 ms.

1) Channel Switching Latency: We first evaluate the latency of switching the operating channel of the wireless card. This operation involves draining the transmit queue, which contains packets waiting to be transmitted. This is followed by a software reset of the card. We measured the latencies for 500 channel switch operations with a channel residence time of 50 ms. We find that the average latency of a channel switch for the Atheros chipset cards is 7.6 ms.

2) Jamming Parameters: Next, we empirically determine the best jamming parameters that minimize the throughput of a legitimate channel hopping communication. The AP is configured to transmit a constant bitrate UDP traffic at 30 Mbps to the client, and the channel hopping residence time is set to 100 ms. Figure 3 depicts the throughput of the legitimate connection as the listen and jam intervals are varied at the jammer. We find that the impact on the throughput for different intervals is not significant. However, a listen time of 5 ms and a jam time of 50 ms has the best impact. We also note that 1 ms interval is too short a time to detect any transmissions, and 20 ms interval is an overkill.

3) Jammer Performance: We evaluate the performance of our jammer on a legitimate communication with and without channel hopping. Figure 4 depicts throughput achieved for the two cases. The jammer uses 5 ms listen interval and 50 ms jam interval and it is active between 10 seconds and 20 seconds from the start of the experiment. The throughput drops significantly to 2.0 Mbps when channel hopping is not used, while it only drops to 18 Mbps when it is employed. In addition, when the jammer is inactive, the channel hopping overhead is minimal.

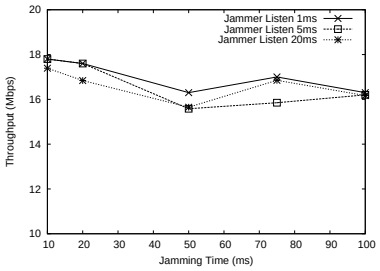


Fig. 3. Throughput for different Jamming Parameters.

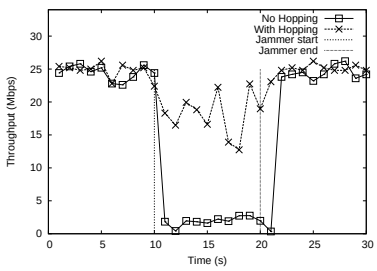


Fig. 4. Throughput trace over time. Jammer is active between 10th and 20th second.

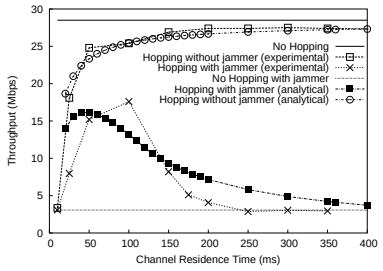


Fig. 5. Throughput Vs Switching Frequency. Jammer operating at listen time of 5 ms and and jam time of 50 ms.

Figure 5 depicts the average throughput values obtained as the channel residence time of the legitimate communication is varied. For comparison, the values obtained through mathematical analysis is also plotted. The uppermost, curve at $y = 28$ plots the throughput of the transmission when there is no jammer and channel hopping is not employed. The curves that increase monotonically are the experimental and analytical ($f(s)$) results for the non-jammed transmission. The curves that drop for large x are the experimental and analytical results ($g(s, \alpha)$) with $\alpha = 1.5$.when the jammer is implemented.[1]

We see that by channel hopping with the right residence time, the throughput achieved is approximately 60% of a jam-free, hop-free transmission. We note the analytical accuracy of hopping with no jammer is very precise, while there is some noticeable difference in the experimental result and the analytical model. While $\alpha = 1.5$ was the best fit (in our opinion), we did not directly measure α (ratio of jammer listen period to legitimate transmission outage period), and may not be using the right α. Also, recall that the analysis assumes a "perfect" jammer who knows precisely when the legitimate transmission switches channels. The combination of these two inaccuracies may explain this difference.

VI. Conclusion

In this paper, we explored the feasibility of implementing channel hopping within 802.11 to protect a legitimate communication from jamming attempts. We begin by evaluating the best channel scanning and jamming strategy that a jammer can implement to reduce the throughput of a legitimate communication that uses channel hopping to resist jamming, and then explore how to best tune the channel hopping strategy to resist such a smart jammer. We then experimented with this channel hopping strategy in the presence of the jammer.

Our experimental results showed that, in practice, while there is a degradation in throughput, this degradation is minimal in the absence of a jammer. Furthermore, in the presence of a jammer, throughput can be maintained at more than half the rate that exists in the absence of a jammer. In future, we plan to extend our analytical model and our software prototype to handle simultaneous communication over multiple channels to further improve resilience to jamming attacks.

[1]Note that $f(s)$ and $g(s, \alpha)$ represent the fraction of rate in comparison to the full throughput, and hence these equations must be multiplied by 28. Also, note that since the average switching time is 10 ms, $s = 10x$.

References

[1] U. Varshney, "The status and future of 802.11-based WLANs," *IEEE Computer*, vol. 36, no. 6, June 2003.

[2] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: defenses against wireless denial of service," in *Proceedings of the ACM Workshop on Wireless Security (WiSe'04)*, Philadelphia, PA, October 2004.

[3] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks," in *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Urbana-Champaign, IL, 2005.

[4] M. Raya, J. Hubaux, and I. Aad, "Domino: A system to detect greedy behavior in ieee 802.11 hotspots," in *Proceedings of Mobisys*, Boston, MA, June 2004.

[5] P. Kyasanur and N. H. Vaidya, "Selfish mac layer misbehavior in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 5, September/October 2005.

[6] A. Shamir, "On the Generation of Cryptographically Strong Pseudorandom Sequences," *ACM Transactions on Computer Systems*, vol. 1, no. 1, February 1983.

[7] http://csrc.nist.gov/CryptoToolkit/tkrng.html.

[8] M. Soroushnejad and E. Geraniotis, "Probability of capture and rejection of primary multiple access interference in spread spectrum networks," *IEEE Transactions on Communications*, vol. 39, no. 6, 1991.

[9] A. Kochut, A. Vasan, A. U. Shankar, and A. Agrawala, "Sniffing out the correct physical layer capture model in 802.11b," in *ICNP '04: Proceedings of the Network Protocols, 12th IEEE International Conference on (ICNP'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 252–261.

[10] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, 1975.