Intelligent Jamming in Wireless Networks with Applications to 802.11b and Other Networks

David J. Thuente and Mithun Acharya

Abstract— **It has long been recognized that complete jamming of wireless networks can be realized by generating continuous noise with sufficient power in the vicinity of the wireless network. There are many disadvantages of this approach including high energy requirements and a high probability of detection. The purpose of this paper is to show that similar jamming effectiveness can be achieved with very low energy requirements and low probability of detection. We discuss various measures of performance for jamming and the role of authentication in denial of service attacks. Then we study and simulate, using OPNET 11.5, the effect of periodic jamming on throughput for an 802.11b network. We add *intelligence* to the jammer by using knowledge of the protocol and exploiting crucial timings and control packets. Intelligent jamming is shown to be more efficient than continuous jamming in terms of signal duration. The next approach is to use a node or two to exploit the backoff timer to create a denial of service attack. Finally, we discuss how these attacks can be applied to networks with protocols such as MIL-STD-188-220D.**

Index Terms—Denial of Service, MAC protocol attacks, intelligent wireless jamming.

I. Introduction

Wireless networks now enjoy widespread commercial implementation because of their low cost, ease of use and setup. Wireless access point based 802.11b hotspots are commonplace with more and more mobile users accessing the Internet wirelessly through various high-end mobile devices already available in the market at affordable prices. However, since accessing wireless media is much easier than tapping a wired network, security becomes a serious concern when implementing any wireless network. [3] and [4] are among hundreds of articles that deal mainly with confidentiality and authentication related security attacks. We consider a particular class of Denial of Service (DoS) attacks called *jamming*. For our purposes, jamming is any attack to deny service to legitimate users by generating noise or fake protocol packets or legitimate packets but with spurious timing. We also study DoS attacks that can be created inadvertently by a greedy member of the network trying to get more throughput for itself. The jamming in [7] and in this paper is at the MAC layer. The results in [1] and [2] were presented at OPNETWORKS 2004 and 2005 respectively and are not available in the open literature. The results presented here are extensions and refinements of those results. All of the results presented here are new and provide considerably more insight into the study of jamming. The jamming results in [1] focused primarily on energy conservation and in some cases were heavily dependent on the TCP performance. The results presented here do not depend on any layer above the MAC layer. This is an appropriate way to test MAC layer DoS attacks. The role of TCP in DoS attacks is the subject of another study.

The most trivial way of disrupting a wireless network is by generating a continuous high power noise across the entire bandwidth near the transmitting and/or receiving nodes. The device that generates such a noise is called a *jammer* and the process is called *jamming*. However, jamming can be made more energy efficient and less detectable if the jammer operates using knowledge of the protocol. Energy efficient stealthy jammers are a goal. We will show how the jammer can briefly disrupt selected control packets and reduce network throughput to zero if need be. Such jammers, which jam the network with the knowledge of the protocol, are termed as *protocol aware jammers*. Jamming and its countermeasures have a long history in military applications. Cheap jammer nodes can be scattered across the enemy battlefield with the purpose of generating noise to bring down the enemy network, to disrupt sensor networks, or even to disrupt command and control networks. It becomes very important for such jammer nodes to be very efficient to ensure their longevity and lower their probability of detection. On the other hand, commercial wireless hotspots or even wireless classrooms can be jammed. Jammers can also be used to disrupt critical communications in times of national emergency. In this paper, we explore periodic and protocol aware jamming attacks as well as attacks from within the network itself. Unfortunately, it is much easier to disrupt wireless networks than to defend against such attacks.

David J. Thuente is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA, phone: 919-515-7003, fax: 919-513-1895, email: thuente@csc.ncsu.edu 27695.

Mithun Acharya is currently a research intern at IBM Thomas J. Watson Research Center, Hawthorne, N.Y. 10532 and is an advanced PhD student with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA, e-mail: acharya@csc.ncsu.edu.

Presenting these results is an important step to achieve reliable communications.

II. Characterizing Various Jamming Attacks

A. Authenticated User Attacks vs. Unauthenticated Attacks

Networks with open access such as 802.11b allow attacks that can be more subtle and more difficult to defend. The misbehaving node(s) attack, presented later, is of this variety. One can argue that all safety critical networks must authenticate all users. Unfortunately, this is not the case. We will also present attacks that do not depend on being an authenticated user. Such attacks have a much wider range of applicability and as such have a significant advantage in terms of domains of application in disrupting communication networks. None of the techniques presented early in the paper require that the user be a member of the network. Both the authenticated user and the unauthenticated attacks make significant use of the MAC protocols for the network. The MAC attacks are ultimately timing attacks and use the knowledge of the various control packet sizes and their timings. Hybrid attacks involving both authenticated users and jammers not part of the network can provide a very effective attack that is difficult to detect.

B. Jamming Attack Metrics

A variety of metrics can be used to compare various jamming attacks. Clearly, the following metrics are all relevant:

1. Energy efficient
2. Low probability of detection
3. Stealthy
4. Strong DoS, complete if so desired
5. Maintain behavior consistent with or close to the protocol standard
6. Authenticated or unauthenticated users
7. Strength against error correction algorithms
8. Strength against physical layer techniques such as FHSS, DSSS, CDMA.

Which of these are most important will depend greatly on the application addressed. Energy efficiency may be the most important metric for jammers of sensor networks that are expected to last a long time. Strong DoS may be the key component if even a few successful messages will compromise your situation such as behind enemy lines. Low probability of detection is crucial if you need to maintain a modestly long-term presence in a hostile area. We will make some comments on the metrics of various techniques in the conclusion.

III. 802.11b MAC Layer

The jamming techniques use the MAC protocols of 802.11b and hence a brief description of the 802.11b MAC protocols is given here. The basic CSMA/CA mechanism is shown in Fig. 1. If the medium is sensed idle for at least the duration of DIFS (with the help of the clear channel signal (CCA) of the physical layer), a node can access the medium at once. If the medium is busy, nodes have to wait for the duration of DIFS before entering a contention phase. Each node now chooses a random backoff time within a contention window and additionally delays the access for this random amount of time. If a station does not get access to the medium in the first cycle, it stops its back off timer, waits for the channel to be idle again for DIFS time and starts the timer again. As soon as the timer expires, the node accesses the medium. Thus, longer waiting stations have the advantage over newly entering stations in that they only have to wait for the remainder of their backoff timer from the previous cycles. If collision occurs, then the station backs off exponentially. ACKs have higher priority over data. So a station wanting to send ACK waits only for SIFS time.

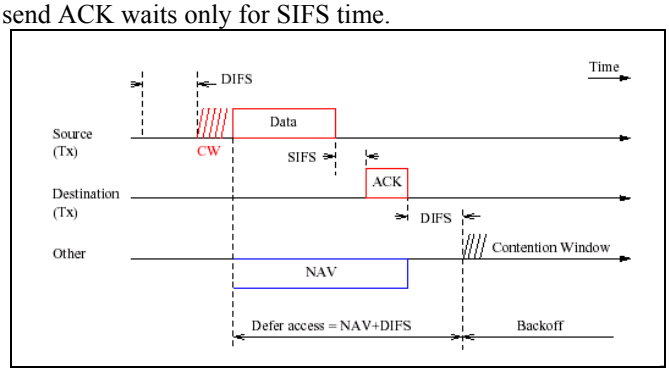

Fig. 1. The basic CSMA/CA in 802.11b networks.

The basic CSMA/CA mechanism cannot solve the *hidden terminal* problem. The problem occurs if one station can receive two others, but those stations cannot receive each other. If both of these stations sense the channel idle and send the data to the station which can see both, collision occurs at the receiver. Fig. 2 illustrates the use of RTS (Request to Send) and CTS (Clear to Send). After waiting for DIFS (plus a random back off time if the medium was busy), the sender can issue a RTS packet. The RTS packet includes the receiver of the anticipated data transmission and the duration of that whole data transmission. This duration specifies the time interval necessary to transmit the whole data frame and the acknowledgment related to it. Every node receiving the RTS now has to set its Net Allocation Vector (NAV) in accordance

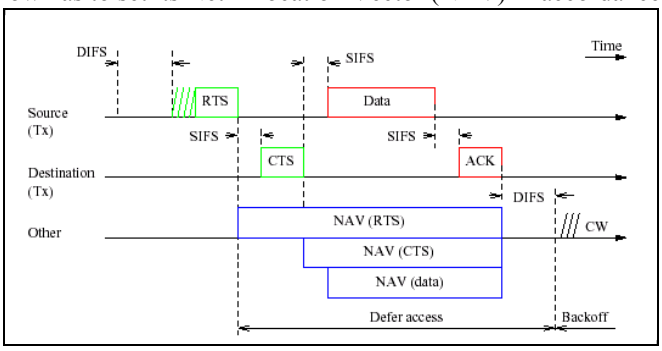

Fig. 2. The RTS/CTS mode in 802.11b networks.

with the duration field. The NAV specifies then the earliest point in time at which the station can try to access the medium again. Following a successful RTS, CTS is sent after a SIFS interval (SIFS < DIFS). After a successful reception of CTS, DATA and ACK follow, with the duration of SIFS between the frames. If a RTS frame undergoes collision, the station backs off exponentially.

IV. Simulation, Traffic, and Jammer Models

The simulation model we used is a heavily modified version of the 802.11 Wireless LAN model (11Mbps) from [9] with the network and transport layers removed. We did not use the network or the transport layers since this interferes with the experiments we are doing with the MAC protocols and the disruption of the network due to MAC level attacks. TCP can strongly decrease the performance of the network due to MAC layer attacks. TCP was used for the preliminary results presented in [1] that showed that wireless networks using TCP were very vulnerable to MAC attacks. Fig. 3 shows the scenario that is used in this paper to study the effects of jamming on network throughput. We have an 802.11b wireless network with nine similar stations and an access point. The access point relays the messages between the nine nodes and is a bottleneck. We chose to do an extension of the model from [9] but with the upper layers removed so as not to skew the experimental results. The Fig. 3 diagram also shows a jammer node placed near the center of the setup.

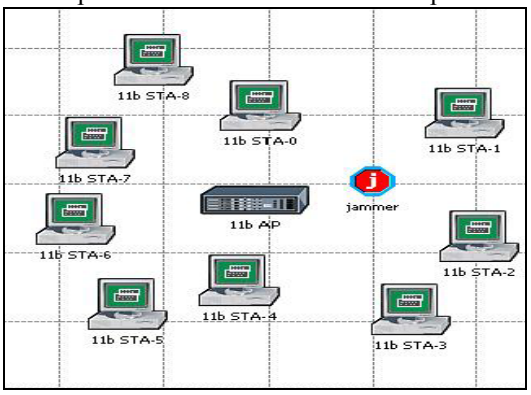

Fig. 3. Basic scenario with jammer.

The node model uses a source and sink module to simulate the higher layers (IP, TCP, Application, etc.). The source model generates packets sent to random destination addresses. The packets received at the destination nodes are discarded at the sink module. The OPNET node model is given in Fig. 4. The parameters for the wireless workstation node model are given in Fig. 5 and are generally self-explanatory.

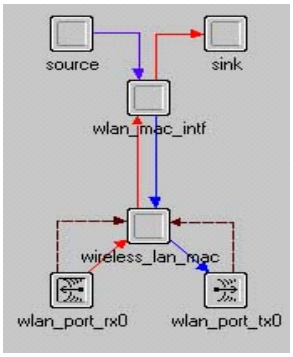

Fig. 4. OPNET node model for the wireless workstation.

The data rate for the network is nominally 11Mbps but the access point functions as a relay point with the workstation nodes randomly assigning any other node as destination node and the access point must relay the message. Hence, the effective throughput must be less than 5.5 Mbps. All packets are transmitted with adequate power so there is never packet loss due to signal strength.

The wireless attributes of a station node are shown in Fig. 5. The packet size distribution is exponential with a mean of 2000 bytes. The interarrival time is exp(.03) for all the nodes unless otherwise specified. The maximum packet size transmitted in a 802.11b network is 2304 bytes and packets over this size are discarded at the source. An exp (2000) packet size distribution has 31% of its packets over the 2304 byte maximum. The RTS threshold is 128 bytes which means that data packets, whose size together with the WLAN MAC header of 28 bytes, exceed this threshold, require a RTS/CTS frame exchange before the transmission of the actual packet over the radio channel. Since the packet size is exponentially distributed with mean of 2000 bytes, RTS/CTS exchange is required for most of the packets; see Fig. 6. As can be seen from Fig. 5, all the wireless station nodes and the access point use Direct Sequence Spread Spectrum at the physical layer. All the nodes employ the DCF basic CSMA/CA access mechanism.

Attribute	Value
Wireless LAN	
Wireless LAN MAC Address	Auto Assigned
Wireless LAN Parameters	(...)
BSS Identifier	Auto Assigned
Access Point Functionality	Disabled
Physical Characteristics	Direct Sequence
Data Rate (bps)	11 Mbps
Channel Settings	Auto Assigned
Transmit Power (W)	0.005
Packet Reception-Power Threshol...	-95
Rts Threshold (bytes)	128
Fragmentation Threshold (bytes)	None
CTS-to-self Option	Enabled
Short Retry Limit	7
Long Retry Limit	4
AP Beacon Interval (secs)	0.02
Max Receive Lifetime (secs)	0.5
Buffer Size (bits)	1024000
Roaming Capability	Disabled
Large Packet Processing	Drop
PCF Parameters	Disabled
HCF Parameters	Not Supported

Fig. 5. Wireless attributes of a station node.

Attribute	Value
name	11b STA-0
model	wlan_station_adv
trajectory	NONE
Destination Address	Random
Traffic Generation Parameters	(...)
Start Time (seconds)	constant (5)
ON State Time (seconds)	constant (1200)
OFF State Time (seconds)	constant (0)
Packet Generation Arguments	(...)
Interarrival Time (seconds)	exponential (.03)
Packet Size (bytes)	exponential (2000)
Segmentation Size (bytes)	No Segmentation
Stop Time (seconds)	Never
Traffic Type of Service	Best Effort (0)
Wireless LAN	

Fig. 6. Traffic model for wireless stations.

The offered load for the network with nine nodes is: (2000 bytes + 28 byte header) * 100/3 pkts/sec. * 8 bits/pkt * 9 nodes * 0.68 (% packets under 2304) = 3.31 Mbps.

Since all packets must be relayed this is a saturated network. In fact, the baseline throughput for this load for the network is just over 1.5Mbps and is given in Fig. 7.

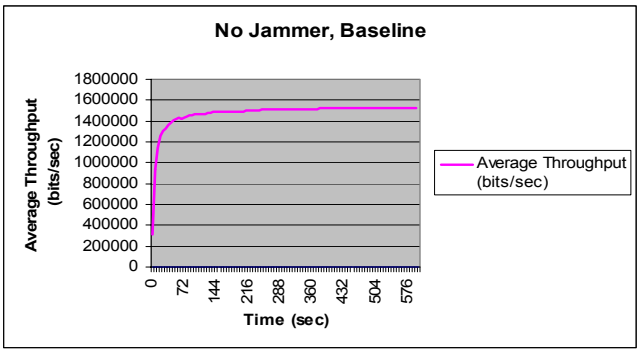

Fig. 7. Baseline throughput for scenario with no jamming.

The throughput is reduced because most packets are over the RTS threshold of 128 bytes and require a four way hand shake both to and from the access point. We will see that we can reduce the throughput to near zero with a very modest amount of jamming.

The jammer module is just a transmitter of noise packets in the 802.11b spectrum. We use the jammer module available in OPNET 11.5 and it is shown in Figure 3. The jammer node model represents a pulsed jammer which can be deployed as a fixed, mobile, or satellite node. The jammer provides transmission on a single fixed frequency band which is masked by a periodic pulse train in time. The source creates and transmits packets for the duration of a pulse (here 1 µsec. or 11 bits). For OPNET simulations it is important to place the jammer and access point at a small positive altitude since if all the nodes are on the surface of the earth, the curvature of the earth will prevent line of sight communication between them. The jammer and access point are placed at an altitude of 10 meters. The jammer has a pulse width which specifies the length of time (in seconds) a pulse is transmitted and a silence width specifies the interval (in seconds) between pulses. Jammer bandwidth specifies the bandwidth (in kHz) of the transmitting channel. Jammer band base frequency specifies the base frequency (in MHz) of the transmitting channel. Finally, jammer transmitter power specifies the transmission power (in Watts) allocated to packets transmitted through the channel. The jammer attributes are shown in Fig. 8.

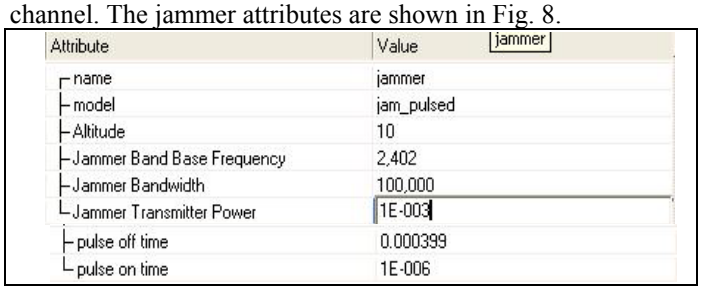

Fig. 8. Jammer Attributers

V. Results for Trivial, Simple and Intelligent Jamming

Since the jammer functions as just described, jamming can be executed without being a member of the network and hence it is easily perpetrated but may not be the most stealthy. It can be easily set up and, we show, that if the behavior of the protocol is used intelligently, it is very powerful. The various types of jamming by an external jammer can be broken into three categories: trivial jamming, periodic jamming and intelligent jamming. In trivial jamming, continuous noise is generated to bring down the network. This is the least energy efficient or stealthy of all the jamming methods. In periodic jamming, a periodic noise pulse is generated irrespective of the packets that are put on the network. The power levels, the pulse duration and periods of silence are the parameters of periodic jamming that can be modified. One form of periodic jamming is busy jamming where a short pulse is generated every DIFS interval (50 µsec.) so that the stations always find the network busy. Since the nodes always find the network busy there is no throughput with this attack. However, the attack can be much less intense and still have complete or nearly complete DoS as

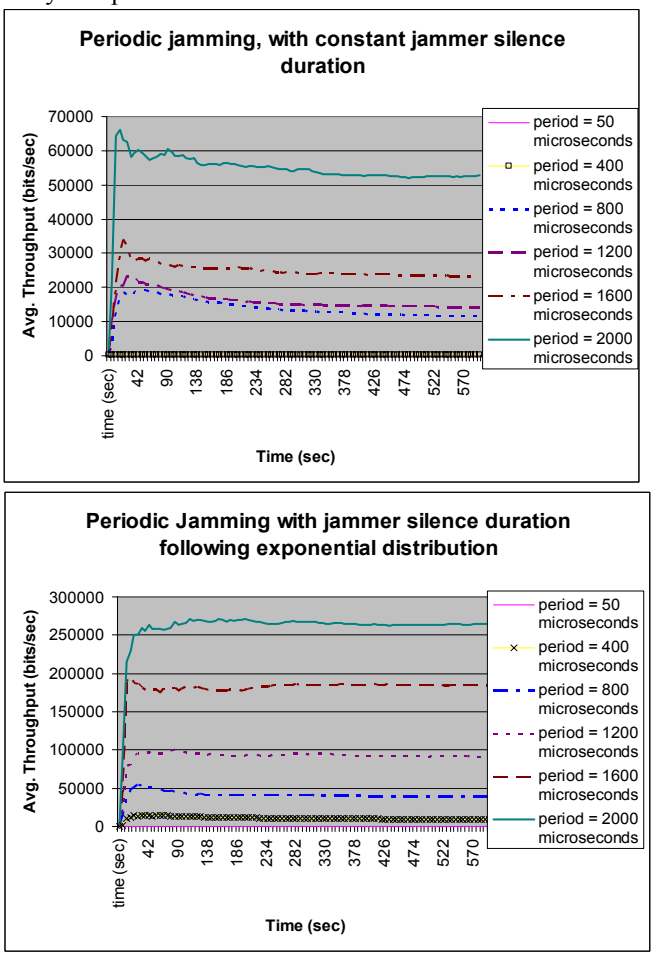

Fig. 9. Periodic jamming – constant and exponential.

shown in Fig.9. The attack is more difficult to detect if it is not regular and hence results for an exponential time delay between attacks is generated and presented again in Fig. 9. Jamming on random exponential periods is a simple but effective DoS attack.

It would be reasonable to suspect that the jammer is so effective even for the long silence periods because of the large packets being transmitted but simulations with packet sizes varying from 200 bytes to 1000 bytes show similar patterns.

For smaller packets, the initial network is less efficient in bps transmitted but the jamming continues to be effective.

In intelligent jamming, the jammer transmits with the knowledge of the protocol. The jammer continuously listens for energy on the network. The jammer can distinguish between the control packets and the data packets by analyzing the length of packets and the inter-packet timings (DIFS, SIFS, Slot time of 50, 10, and 20 μsec. respectively and RTS, CTS and ACK of 20, 14, and 14 bytes respectively.) We identified four types of intelligent jamming. In CTS corrupt jamming, the jammer waits for any arbitrary RTS packet. For every RTS packet the jammer sees, it starts counting down for a period of SIFS from the end of RTS packet and then issues a short jamming pulse that disrupts the CTS packet that would follow the RTS packet. Since the CTS packet does not get through, only packets of less than the RTS threshold of 128 bytes ever are transferred. This attack is among the most efficient (energy use) of the methods presented here as the jammer is active for only the short interval of time necessary to disrupt the CTS packet. The results are presented in Fig. 10.

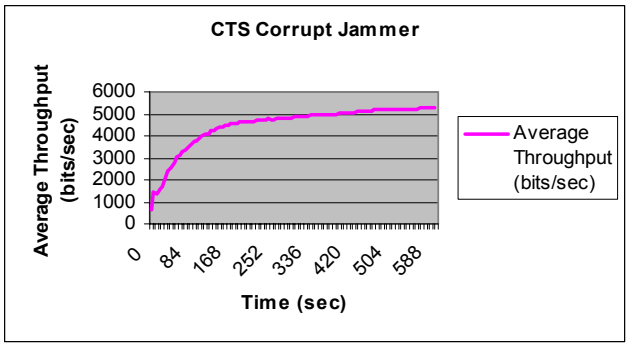

Fig. 10. Throughput results for CTS jamming.

Other types of jamming discussed in [1] include ACK corruption jamming wherein an ACK packet is destroyed after DATA is sent. Also DATA corruption jamming wherein a DATA packet is corrupted after the CTS is received or a jamming signal is sent because the data packets is longer than the control packets and hence is easily identified. With this extension DATA corruption works for both RTS/CTS as well as the basic CSMA/CA. Another jamming technique tested was DIFS-wait jamming, wherein a short pulse is sent by the jammer after sensing the medium idle for DIFS time to disrupt either a RTS packet or a CSMA/CA DATA packet. In a network with high traffic, it is very likely that if the medium is idle for DIFS time, a transmission occurs immediately after that. Intelligent jamming can be shown to be several orders of magnitude more effective (in terms of total duration of jamming signal) than trivial jamming of the same power. In summary of unauthenticated jamming, we have looked at:

1. Trivial Jamming
2. Periodic Jamming
    a. Busy Jamming
    b. Constant and exponential silence periods
3. Intelligent Jamming
    a. CTS Corruption Jamming
    b. ACK Corruption Jamming
    c. DATA Corruption Jamming
    d. DIFS Wait Jamming

and all have proven effective with CTS, ACK, and DATA, being theoretically the most efficient but exponential silence periods also being very effective and more stealthy. There is not enough room to include all of the simulation results but some similar work has been presented in [1].

VI. Node Misbehavior Jamming

For our considerations, a misbehaving node is a network node that unilaterally has modified its MAC protocol to attempt to gain an unfair advantage in transmitting its packets or in transmitting packets to deny others the legitimate use of the network. The interesting and somewhat surprising result is that if the misbehaving node wants to gain an extra share of the network bandwidth then it is possible that its behavior will result in lesser throughput for itself as well as all of the other nodes in the network.

One could easily imagine a network that has not been as responsive to command and control functions and a node manager on the network reconfiguring its MAC to improve its chances to gain access. A supervisor might even suggest such a change for a heavily loaded network. What we will see is that the performance suffers severe deterioration for all the nodes on the network as well as the node that was intending to gain an advantage.

A user in an access point based 802.11b hotspot might selfishly choose not to adhere to the MAC protocol to get an unfair share of the bandwidth. An easy way to accomplish this is by not adhering to the random exponential backoff-algorithm. The user might decide to either use smaller backoffs or choose no backoff at all. [5] presents the results of the effect of misbehaving nodes that select smaller backoff times on the total network throughput and proposes a solution to solve the misbehavior attack. This solution requires a significant modification to the 802.11b protocol (in short, instead of the nodes selecting the random back off times, the access point selects the *next* random backoff time and conveys it to the sending nodes via CTS packets) and hence is not realistically viable as a countermeasure. In this section, we use different and more effective misbehaving patterns than those proposed in [5]. We also show that two misbehaving nodes are more effective than just one. We consider the effect of misbehaving node(s) that waits for DIFS time and just a single slot time before entering the contention phase. The random exponential backoff is not used. If the misbehaving node(s) find the medium busy, they wait until the medium becomes free. The node enters the contention phase only after waiting for a period of DIFS and chooses a single slot time for its backoff.

We also show that two misbehaving nodes are able to cause a more severe degradation of network performance than just one such node. Moreover, the energy required for two such nodes is only slightly increased from just one such node. In addition, two misbehaving nodes are able to accomplish this in a stealthier manner as will be explained later. No one has previously proposed this two node misbehavior DoS. All of the results hold whether the misbehavior is motivated by greed for bandwidth or by trying to create DoS attacks.

The node misbehavior attack has some similarities to the RTS faking attack [7]. In the RTS faking attack, a jammer sends out fake RTS packets whenever it finds the medium idle and reserves the channel for maximum time duration possible thereby degrading the network throughput with very little effort. The misbehaving attack we discuss in this section, on the other hand, captures the medium only when it has data to send by not adhering to the backoff protocol. Hence, other nodes will get a chance to transmit and the misbehavior attack can be less effective in terms of degrading the network throughput as compared to the intelligent jammer attacks. This misbehavior can be implemented unilaterally on a node and the other nodes in the network will not be equipped to even detect the misbehavior, much less to offer any countermeasures. In this way, this DoS attack is superior to many other attacks including the fake RTS attack that can be instantly recognized by all nodes that are listening to the network. For a heavily loaded network, the extent of the decrease in network throughput is severe. This attack creates some serious questions about the stability of the 802.11b networks to withstand even minimal intelligent attacks.

To establish a benchmark, we measure the total throughput for the access point based wireless network without any jammers and without any misbehaving nodes. The results are shown in Figure 7. We use the same traffic profile as shown in Fig. 6 for all our experiments unless otherwise specified. Recall that without any disruption, the average throughput is about 1.5 Mbps at the end of 600 seconds.

The single or double node(s) to misbehave are chosen randomly and the simulation is run for 600 seconds. For the misbehaving nodes, we choose a packet size of uniform (140-160) bits with an interarrival time of exp(.005).

The offered load for the network with one misbehaving node is then:

Misbehaving node:

(150 bytes + 28 byte header) * 200 pkts/sec. * 8 bits/pkt = 284,800 bps.

Sum of regular nodes:

(2000 bytes + 28 byte header) * 100/3 pkts/sec. * 8 bits/pkt * 8 nodes = 4,326,400 bps.

The offered load for the misbehaving node is a small fraction of the total load of the other nodes and roughly 50% of that of the any other single individual node. As shown in Fig. 11, the throughput decreases by 73% from 1.5 Mbps to 0.4 Mbps in the presence of a single misbehaving node. Two misbehaving nodes reduce the throughput to near zero. This is clearly a very effective DoS attack and one that is hard to detect.

One might expect that much of the throughput for this case would be from the misbehaving node since it transmits shortly after the DIFS time and before most regular nodes gets a chance to transmit. From Fig. 12 it appears that very little misbehaving node traffic is actually being sent. Somewhat more regular node traffic is sent but it is also very limited. Since the misbehaving node is not actually sending much traffic, its chances of being detected are greatly reduced. It is clear that the network has reached a state where the collisions in the MAC layer are disabling the network.

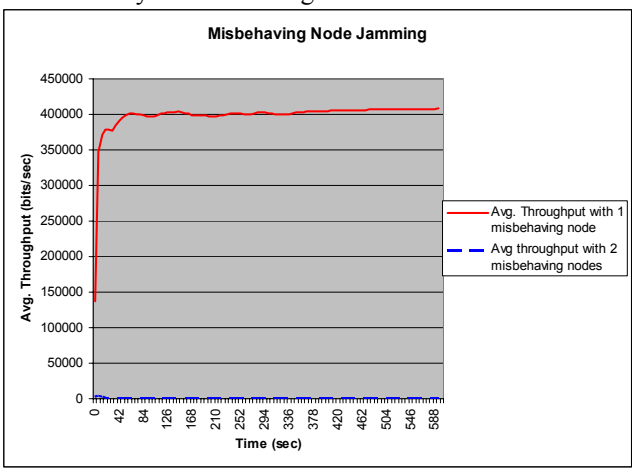

Fig. 11. Throughput for one and two misbehaving nodes.

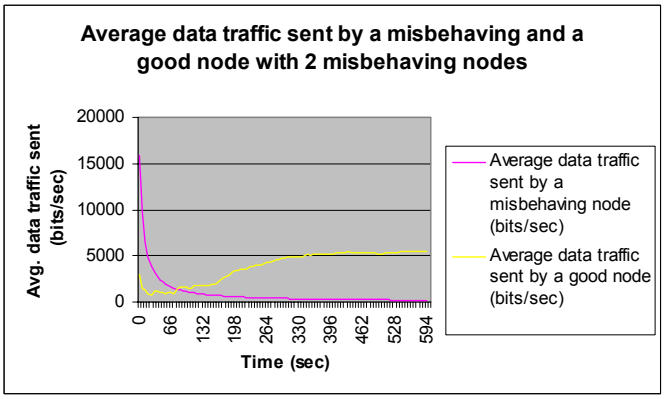

Fig. 12. Traffic sent for one and two misbehaving nodes.

It is clear that the misbehaving node attack is very strong. Since for any individual transmission, it conforms to the 802.11b protocol, it is relatively hard to detect. Moreover, it does not need to send many packets so that the attack becomes nearly invisible. The constraint on this attack is that the misbehaving nodes are required to be part of the 802.11b network and hence might be recognized as being an intruder after transmitting one or two packets. Most 802.11b networks are not configured to detect intruders so it is generally an extremely potent attack.

VII. Hybrid Jamming

We have presented different types of jamming attacks from periodic jamming with exponential silence periods to CTS jamming, discussed ACK jamming, and simulated them in OPNET. Protocol aware jamming is quite easy in 802.11b because the timing, size, and sequences of control and data packets are simple and public. If the underlying MAC does not require authentication, then we saw that two misbehaving nodes can mount an effective DoS attack. Simple periodic jamming relies on sending periodic short pulses while CTS corrupt, ACK corrupt, DATA corrupt jamming rely on disrupting selected control or data packets. These attacks are possible without the node being part of the network.

There are many ways to combine the intelligent jamming techniques that were presented above to be more effective and more stealthy than individual techniques. We saw previously that periodic jamming was very effective and exponential periodic jamming somewhat less so. However, any regular jamming signal will be more easily detected than a random signal. Hence, to decrease the likelihood of being detected we recommend exponential periodic jamming with a relatively large period. It will happen that packets get through but we could still jam the ACK. We can always recognize a data packet since its length is longer (28 byte header) than the control packets and hence we can recognize it and the end of the packet transmitted. The jammer waits for SIFS time and then jams the network and the corresponding ACK.

Another hybrid attack for intelligent jamming could consist of just combining CTS, DATA, and ACK jamming with CTS jamming deployed with probability p. If the CTS is not jammed, the DATA is jammed with probability q and finally, if neither the CTS nor the DATA is jammed then the ACK is jammed with probability r where r = 1 guarantees no throughput.

We saw how effective jamming attacks with two misbehaving nodes could be. However, if the backoff is always set to 1 slot, a network monitor might be able to recognize the pattern and become suspicious. We can also do hybrid attacks with two misbehaving nodes using the standard DIFS period but instead of using a single backoff slot time vary the backoff between the two misbehaving nodes. Formally:

Node 1: with probability p use backoff = 1 slot
with probability 1-p use backoff in [cw/2, cw]
Node 2: with probability 1-p use backoff = 1 slot
with probability p use backoff in [cw/2, cw].

The effect will be to make the nodes appear closer to normal backoff and to still maintain nearly the same level of effectiveness for DoS.

It is also possible to combine intelligent jamming with misbehaving node(s) but we will not pursue that here since combining authenticated and unauthenticated nodes increases the complexity decreases the range of applicability.

VIII. Other newtwork attacks

These type of attacks with unauthenticated nodes are possible for any network with public protocols but few are as easy to implement as for 802.11b. Any protocol with fixed timing between packets of known length is very vulnerable to this type of jamming attack. This is true even with error correction algorithms and time dispersal coding. The jamming is more difficult but the basic approach is still the same. Military implementations of a protocol such as MIL-STD-188-220D are protected with encryption, error correction and time dispersal coding. However, if we know the structure of the packet and the error correction coding, certain parts of the packet can be shown to be vulnerable. An OPNET model can be built that includes the error correction algorithm and the vulnerable points of the packets can be determined. If there is a deterministic MAC protocol then we may know the necessary timing of the protocol attack. We will not discuss the fine points of such a DoS attack since it might be sensitive even though all of the information is in the public domain.

We have focused on a network using an access point but most of the results could have been applied to an ad-hoc network. We are starting work on the ad-hoc network DoS jamming attack.

IX. Conclusion

We studied various protocol aware jamming attacks that can be launched in an access point based 802.11b network. We started by presenting the various jamming attacks ranging from trivial jamming to intelligent jamming attacks such as CTS corrupt jamming. We then presented simulation results showing the effect of misbehaving nodes that do not adhere to the underlying MAC protocol. The network throughput suffered drastically even in the presence of a single misbehaving node and more so with two misbehaving nodes. We then presented several hybrid attacks that increase the effectiveness of the attack or the decrease the probability of detection of the attack.

Some of these results can be repeated directly for IEEE 802.11a/e/g. We would like to investigate the error correction algorithms used for the various implementations and incorporate these into the OPNET model. Then the attacks could be structured to counteract the error correction algorithms. Error correction algorithms will make DoS attacks more difficult but not inherently impossible. In this paper, the experiments assumed a base station oriented network. We would like to conduct experiments wherein the IEEE 802.11b is functioning in the ad-hoc mode. The results for ad-hoc networks will definitely be different from the base station oriented network. There is an opportunity for more forms of intelligent jamming in ad-hoc networks. We may explicitly extend the results here to other wireless protocols such as MIL-STD-188-220C. We believe that the jamming attacks presented in this paper are a solid first step.

References

[1] Acharya, M., T. Sharma, D. Thuente, D. Sizemore, “Intelligent Jamming in 802.11b Wireless Networks”, OPNETWORK 2004, August 2004.
[2] Acharya, M., and D. Thuente, “Intelligent Jamming Attacks, Counterattacks and (Counter)^2 Attacks in 802.11b Wireless Networks”, OPNETWORK 2005, September 2005.
[3] Bellardo, J., S. Savage, “802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions”, USENIX 2003.
[4] Fluhrer, S., I. Martin, I., A. Shamir, “Weakness in the key scheduling Algorithm of RC4”, LNCS, 2259, 2001.
[5] Kyasanur, P., N. Vaidya, “Detection and Handling of MAC Layer Misbehavior in Wireless Networks”, DSN 2003.
[6] Leon-Garcia, A., I. Widjaja, “Communication Networks,” McGraw Hill, Boston, 2000.
[7] Negi, R., A. Rajeswaran, “DoS Analysis of Reservation Based MAC Protocols”, ICC 2005.
[8] Schiller, J., “Mobile Communications”, Addison-Wesley Longman Publishing, Boston, 1999.
[9] Lab for Session 1332: Planning and Analyzing Wireless LANs, OPNETWORK 2005.
[10] IEEE Std 802.11b-1999/Cor 1-2001 Standard for wireless LAN medium access control (MAC)