Feasibility of Channel Hopping in Jamming Attack

Suman Bhunia, Shamik Sengupta
Department of Computer Science and Engineering, University of Nevada, Reno, USA
sumanbhunia@gmail.com, ssengupta@unr.edu

***Abstract*—Currently deployed WLAN network use cooperative spectrum sharing and is prone to jamming based denial of service attack. In this article we propose a model to avoid jamming that use decentralized channel hopping algorithm. Upon detection of jamming, the network switches its channel of operation in accordance with pseudo-random sequence based rendezvous. We develop a testbed to evaluate the feasibility of this model in standard WLAN devices. Results show that performance is significantly enhanced when using the proposed model.**

***Keywords*—jamming, DoS attack, channel hoping, testbed**

I. INTRODUCTION

The nature of broadcasting electromagnetic signal on shared wireless medium is vulnerable to jamming based denial of service attack. Attacker emits jamming signal to create high interference to the communication of a legitimate wireless network and disrupt it. Advancement in software defined radio (SDR) and dynamic spectrum access (DSA) makes it even easier for smart malicious attacker to efficiently find legitimate communication and disrupt. Jamming strategies are widely analyzed in recent time which makes it more critical to design an efficient anti-jamming scheme.

Many jamming prevention schemes have been proposed in recent years. These schemes are mainly of two types: *spatial retreats* and *channel surfing* [1]. In spatial retreat, Mobile Nodes (MN) relocate themselves to avoid jamming region. In this approach, a network is geographically overlapped by several access-point (AP). Upon detection of jamming, MNs physically move to another position and establish connection with another AP. This approach has several drawbacks. It requires all MNs to be self-relocating. Moreover establishing connection with another AP is a critical job that causes delay. In the channel surfing scheme, network under jamming attack follow migration strategy of channel hopping. Upon detection of attack, the network changes its channel of operation to a new channel decided by the AP.

A smart attacker would not jam a channel randomly. It obtains highest reward by successfully jamming channels with active or most "prioritized" communicating users. Before jamming, a smart attacker scans a channel and if it senses active transmission on the channel, then jams the channel. The attacker also knows that it might be detected by the legitimate network. So, it jams a channel for *jam duration* and then scans again to see if the network is still active on the channel. If it does not find any active user on the current channel for *channel scan duration*, then it switches to a new channel and scans again. The flow chart for attacker is given in Fig. 1.

IEEE 802.11 uses CSMA-CA protocol to avoid collision. Apart from jamming, ISM bands are prone to heavy interference from other networks using the same channel or adjacent channels. Cross channel interference makes it very difficult to distinguish between jamming and interference. Again, introduction of channel fragmentation and/or bonding introduces complexity in model. Quality of Service is strictly dependent on the application type. As an example, Real-time streaming video requires high throughput and less delay but can tolerate packet loss. On the other hand, application like file transfer or web browsing can tolerate delay but can not tolerate packet loss. A network with low load can stay on a channel with certain level of interference. However, with a heavy demand, the network is better of switching to a channel with lower interference in order to guarantee higher data rate and less delay. In this article, we propose an algorithm to take decision of switching channel based upon channel condition and application demand.

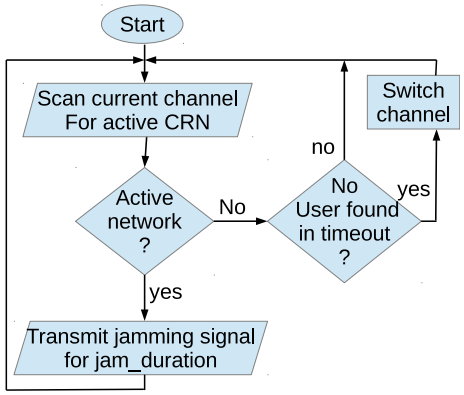

Fig. 1: Flowchart for attacker

The rest of the article is organized as follows. Section II presents the system architecture. Section III describes the testbed. In section IV we discuss about the results. Finally section V concludes the article.

II. SYSTEM ARCHITECTURE

An intelligent attacker can sniff communication between AP and its MNs. To avoid this problem, we use sequence based rendezvous [2] channel hopping. While connecting with the AP for the first time, each MN receives a *pseudo random* channel hopping sequence for rendezvous. This sequence is not known to the attacker. Attacker being only capable of transmitting and receiving packets on only one channel at a time, doesn't have the knowledge about which channel the network is going to switch. So, the attacker has to scan through all the channels to find out the channel to attack. This channel scanning causes delay for the attacker to attack.

We consider the case of infrastructure based wireless network where multiple mobile nodes (MN) are connected to an AP. All nodes share the same channel in accordance with IEEE 802.11 (WLAN) MAC protocol. AP periodically broadcasts *beacon* signal to check whether all nodes under its supervision can be reached successfully. As AP broadcasts beacons after fixed interval, all MNs know the next beacon broadcast time

This research is supported by NSF CAREER grant CNS #1346600.

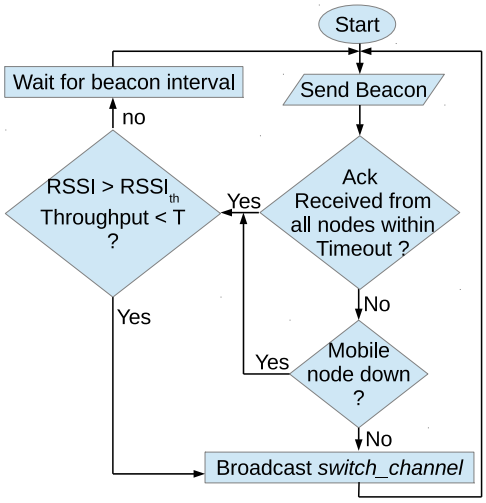

Fig. 2: Flowchart for Access Point

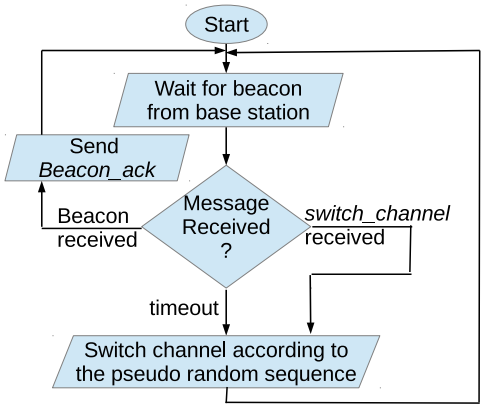

Fig. 3: Flowchart for MNs

Fig. 4: Picture of the testbed

and stay silent during that period. After receiving the beacon, all MNs acknowledge to the beacon. In the beacon message, AP schedules slots for all associated nodes to acknowledge. This eliminates the chance of collision among MNs while sending *beacon_acknowledgment*.

scenario 1: If the AP does not receive acknowledge from a MN within a *beacon timeout* interval, it checks whether this may be caused by a possible jamming attack or the MN is down. AP keeps track of how many times consecutively a MN has not acknowledged beacons. If a MN has not acknowledge the beacon for *ack_timeout* times, AP conceive that the MN is down. Upon detection of jamming, the AP broadcasts *switch_channel* message to all MNs under its supervision. Upon receiving the *switch_channel* message, all MNs switch to a new channel. If a MN does not receive *beacon* for *beacon_timeout* interval, it takes it as jamming and hop to a new channel and wait for the *beacon*. All nodes including AP determine the next channel to switch, based on the channel hopping sequence and the previous history of channel hopping. This keeps the integrity that all legitimate nodes in the network know what would be the next rendezvous channel. An attacker can't know the channel that a network is going to hop even if it sniffs the *channel_switch* message.

scenario 2: Consider the case where attacker can cause selective jamming to confuse the network with interference. Interference cause decrease in effective throughput. In our model the AP initiates channel switching if the current achievable throughput doesn't satisfy the traffic demand. If nodes in the network is silent then received signal strength indication (RSSI) should be low. However if there are collision and packet transmission failures due to interference then RSSI should be high but throughput will fall down. AP tracks this situation and triggers channel switch. Flowcharts for AP and its MNs are depicted in Fig. 2 and Fig. 3 respectively. Note that, these flowcharts provides the schematic overview of the jamming avoidance strategy. Normal packet transfer between MNs follow WLAN MAC protocol.

III. Testbed Developement

In this article we use unlicensed ISM bands for experiment setup. All nodes have a single communication interface and can switch channel. We use Soekris [3] boards equipped with Ubiquiti-SR2 Hi-performance 2.4 GHz 802.11b/g mini-PCI module [4] that has Atheros, 4th Generation, AR5213 processor. All nodes run Ubuntu 12.04 as operating system with Ath5k [5] as WLAN driver module. WLAN cards are configured to operate on IEEE 802.11-b mode with access to 2.4 GHz ISM band. We set the retry-limit to 7 and RTS-threshold to off. Although we run our experiment on 12 channels, proposed algorithm is equally applicable to devices with access to more channels. Fig. 4 shows the picture of the testbed. The node in the middle is configured as AP and other two nodes are connected to the AP as associated MNs.

For creating a jamming attacker, we can either use another Soekris board with same configuration or a software defined radio. For simplicity we have used AirPcap 802.11 wireless packet capture [6] to create jamming. AirPcap can sense the wireless environment and has the capability of injecting 802.11 packets on desired frequency or channel. The attacker transmits garbage packets with high data rate to create jamming. This makes higher collision rate for legitimate data communication over attacked channel. Eventually, the legitimate communication on the attacked channel can either achieve a very low data rate or loose the communication over that channel. In the current experiment, we create some garbage packets to send. The AirPcap is configured to transmit with 802.11-G protocol with a data rate of 54 Mbps.

We use Chanalyzer equipped with Wi-Spy [7] to observe the power spectral density (PSD) over 2.4 GHz ISM band. This gives us a hint about data transmission and jamming over different channels. For example, when the network is using channel-1 of 802.11 (central frequency is 2.412 GHz with bandwidth of 20 MHz) we observe a PSD chart as given in Fig. 5. Clearly, the energy on channel 1 is very high and there is some energy leakage on adjacent channels. Fig. 6 plots

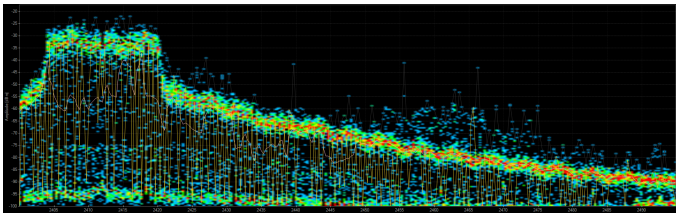

Fig. 5: PSD when network is operating on WLAN channel 1

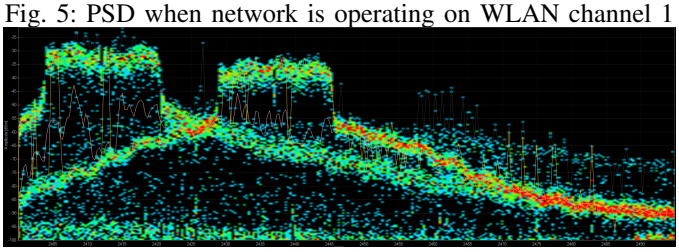

Fig. 6: PSD when attacker is on channel 6 and the legitimate network is operating on channel 1

the PSD when the legitimate network is operating on channel 1 and attacker transmits on channel 6. We can clearly see two main lobes: on channel 1 and on channel 6. If the attacker is successful in attacking the legitimate network, then we would be able to see only one lobe.

IV. Results and Discussions

IEEE 802.11-b can achieve raw throughput (at MAC layer) of 11 Mbps which can deliver UDP traffic at 7Mbps roughly. Keep in mind that our model does not affect the packet transfer protocol for WLAN. This model can be used with WLAN protocol that gives higher throughput such as IEEE 802.11-n. In the current configuration data traffic is going from one MN to another MN through the AP, the actual end-to-end achievable throughput is around 3 Mbps.

We use Iperf [8] for measuring throughput for all nodes in network. We configure one MN as UDP sink while another MN is sending UDP traffic to the sink at a speed of 3 Mbps. Without jamming, the sink can receive traffic at 3 Mbps. Now, in presence of jamming, the data rate falls very abruptly. The measured data rate while jammed is in the order of Kbps and is not stable. Achieved data rate in presence of attacker depends on the attacker's packet sending interval and distance of attacker from the AP. Jamming effect or decrease in data rate is proportional to the distance of the attacker and the receiver node or AP. Now, when we run our algorithm, the network itself switches to a new channel to avoid the jamming. In Fig. 7, we can see that the network was operating on channel 1. Now when the AP observes an attack on this channel (not receiving beacon acknowledgment or decrease in achieved data rate), it switches its channel of operation to channel 11. After switching to a new channel, the achieved data rate is 3 Mbps. However if the network doesn't use channel switching mechanism (as in the conventional network) then data transferred would blocked by the attack. Fig. 8 depicts throughput comparison for a simulation scenario.

For the current simulation we take RSSI threshold as $-45dBm$. Throughput threshold is kept at a value of 75% of achievable throughput without interference. So, the throughput threshold in this case is 2Mbps. When we configure the attacker not to jam all packets but jam some packets,

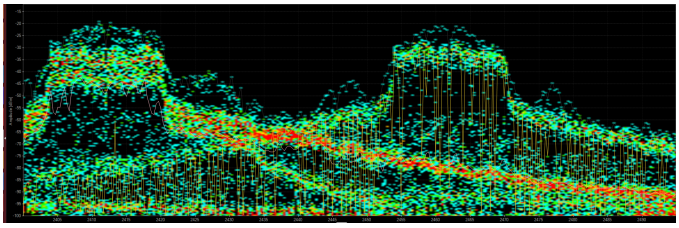

Fig. 7: PSD when attacker attacks on channel 1 and the network switch its channel of operation to 11

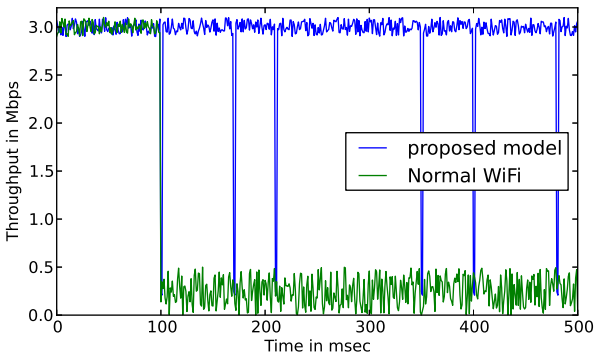

Fig. 8: Throughput for one simulation

it creates high interference to the legitimate communication but the channel is not completely blocked. In this case, the network achieves a throughput lower than the desired value. Now, when AP observes throughput lower than throughput threshold while the RSSI value is greater than RSSI threshold, AP triggers channel switching. After channel switching, the network observes full network throughput.

V. Conclusion and Future Work

In this article we emulate channel hopping mechanism to avoid jamming based Denial of service attack in WLAN. With successful testbed deployment we observe that channel hopping is useful in obtaining better performance in presence of jamming attack. We need to further investigate optimal system parameter suchs as *ack_timeout*, $RSSI_{th}$ etc. This model can be further enhanced with Dynamic spectrum access and with larger spectrum access range.

References

[1] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: defenses against wireless denial of service," in *Proceedings of the 3rd ACM workshop on Wireless security*, pp. 80–89, ACM, 2004.

[2] L. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pp. 1–7, 2008.

[3] "Soekris board." http://soekris.com/products/net5501.html.

[4] "Ubiquiti-sr2 hi-performance 2.4 ghz 802.11b/g mini-pci module." http://www.ubnt.com/sr2.

[5] "Ath5k wireless driver." http://wireless.kernel.org/en/users/Drivers/ath5k.

[6] "AirPcap 802.11 wireless packet capture tool." http://www.riverbed.com/products-solutions/products/network-performance-management/wireshark-enhancement-products/Wireless-Traffic-Packet-Capture.html.

[7] "Wi-spy spectrum analyzer." http://www.metageek.net/products/wi-spy/.

[8] "Iperf: Network performance measuring tool." http://iperf.fr.