Neighbor based Channel Hopping Coordination: Practical against Jammer?

Faraz Ahsan[∓], Soufiene Djahel[∓], Farid Naït-Abdesselam[∓] and Sajjad Mohsin[±]

[∓]LIFL – UMR CNRS USTL 8022 – IRCICA
University of Lille, France

[±]COMSAT Institute of Information Technology
Islamabad, Pakistan

***Abstract*— As compared to its wired counterpart, wireless network is relatively new and is exposed to some additional threats specific to the underlying medium. Among these threats the jamming attack which can take place easily due to the open nature of wireless medium. A device or person can continuously emit radio signals to disturb a valid conversation. If it lasts for sometime continuously, it can result in total collapse of a network using single channel.**

In order to evade a jammer in an ad hoc network, we propose a proactive channel hopping scheme based neighbor correspondence. Rather than detect and react we rely on prevention is better than cure. Each node communicates with its neighbors on different channels, coordinated between them dynamically. Furthermore, the control and data channels of each node are separated. This way redundancy at the node-level is provided so that even if nodes on the jammed channel can not be approached but they still are able to contact others by visiting their control channels; avoiding the node on the jammed channel from starvation. Hence, even if the network is exposed to the jammer, a complete failure is prevented. The simulation results show that our scheme is efficient and is able to reduce the jammer's impact significantly, as compared to the scheme presented in [8].

***Keywords* – MANET, Multichannel MAC Protocol, Channel Hopping, Jamming Attack.**

I. Introduction

Wireless networks are becoming increasingly popular for all kinds of futuristic networks in todays world. Unlike wired, wireless networks rely on an open medium and hence face a far larger set of security threats and vulnerabilities than wired ones. As we move on from wireless infrastructure to ad hoc, access point (AP) setup is eliminated and we are left with no central coordinating authority, varying topology, limited battery life, lower transmission rate, and desired cooperation amongst nodes for packet forwarding, etc. Hence the complexity further intensifies posing it as a greater challenge to achieve similar efficiency and effectiveness as compared to the wired infrastructure case.

Due to the open nature of wireless medium, an anomaly can listen to and disrupt an on-going communication by sending continuous or periodic fake messages resulting in collisions. Eventually, this can result in increasing backoff at the node level and extinction of the traffic at the network level. Hence, a wireless device that intentionally launches radio interference attacks in a wireless network is referred to as a jammer [6]. Continuous emission of radio signals are taken as physical layer jamming whereas the MAC layer jamming involves violation of protocols' rules on the said layer. Jamming attack is dealt with active defense strategy which involves either physical escape of the jammed region, like spatial retreats, or logical escape that involves channel switching as described in [3]. The former approach recommends moving away from the jammed region whereas the latter suggests leaving the jammed channel and tune to a new channel. Either way, the network members need to regroup. Active defense is not recommended for smaller devices and passive strategy, like doze for long periods to save energy, is generally opted. Nodes remain physically and logically in the area being jammed and monitor the channel periodically while dozing. Network is restored when the jamming phase is over.

IEEE 802.11 [2] offers multiple channels which can be used simultaneously to reduce a few of the existing deficiencies of wireless mentioned above, however practically it is not being implemented. Usually in an infrastructure based network AP leads to a new channel if required, either with other APs or avoid congestion with other APs or due to the underlying MAC protocol, and nodes follow accordingly. On the other hand, limited battery life and the lack of central coordination in an ad hoc network setup hinders the use of multiple channels. Studies have highlighted that multiple channels can be used to acquire better network performance through simultaneous communication of different channels as shown in Figure 1. Though these studies differ in terms of approaches and required resources but they all emphasize the same concept that is to use unused and available channels in parallel [13]. Among these techniques parallel rendezvous is considered robust as it eliminates the single control channel bottleneck and utilizes all the channels concurrently for data as well as control packets.

In this study we propose the idea of nodes having their designated control channels (CC) to be approached by others. Furthermore, rather than data is exchanged on the control channel of the receiver, both parties negotiate to come up with a new channel for data transfer. Both the nodes hop to the new channel, transfer data and return to their respective channels. Thus, separating control and data channels for every node pair, along with each node communicating with its neighbors on different channels increases redundancy to avoid isolation of a node and collapse of the total network in case of a jamming attack.

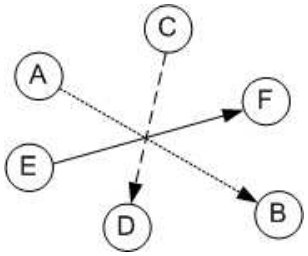

Figure 1: An ad hoc network with simultaneous multichannel communication

II. Related Work

Recent studies suggest channel hopping as a logical escape in case of jammed channel. If a valid communication is not heard for a period of time on the common channel, nodes initiate jamming attack detection, individually. If the said channel is detected as jammed, nodes switch channel to locate other nodes and resume communication as a reactive mechanism. [10] proposes that once a node switches its channel, it needs to notify its neighbors by broadcasting this change on other channels. This strategy can result in heavy traffic in case of a large number of affected nodes or if the jammer is intelligent enough to scan other channels forcing legitimate nodes to hop frequently.

Based on jamming detection, authors in [9] suggest a protocol suite with frame masking, frequent hopping, fragmentation and redundant transmission of packets to overcome a jamming attack. Depending upon the strategy selected by the jammer, each one of them is incorporated step-by-step, leaving the jammer with the option to jam a single channel, as the best strategy. The protocol suite, when applied as a whole for every affected node in a dense environment, increases computation and communication delay and can affect the network performance.

[7] surveys about the options to escape a jammer which includes frequency hopping on the basis of a pre-shared secret function between legitimate nodes. However, they suggest in [3] to use coordinated channel switching where nodes move to next channel when current channel is found to be jammed. Besides this, they have suggested another strategy based on jammed area detection where only the affected nodes switch to the new channel instead of whole network. Boundary nodes are defined just outside the jammed area that bridge the communication between affected and non-affected nodes by switching between the two channels periodically. This way, not all the network undergoes change and only nodes in the jammed region need to be reorganized. However, this scheme is tedious and generates additional network traffic for coordination between nodes. First, the detection of a jammed region in a distributed manner generates reasonable computational and communication overhead. Second, the selection of boundary nodes has a similar impact. Finally the delay incurred and cost involved due to periodic channel switching by the boundary nodes are also significant.

The work presented in [8] differs from the above in two important points. First, rather than reactively switching channels it takes into account the proactive channel hopping. Secondly, it considers a knowledgeable jammer which has all the information to target a valid communication except the seed shared between the node and the AP. Thus, legitimate nodes hop channels periodically on a set pattern and jammer tries to follow them by scanning different channels, randomly. Channel hop is initiated by the AP using a special broadcast message. If the node does not receive this message, it hops to the next channel when it times out. However, they have not considered scalability of the network and taken only two legitimate nodes into consideration; one AP and a single node. Our proposed solution differs in a sense that an ad hoc network is chosen, having considerable number of nodes, with the provision that nodes can join or leave the network any time. Additionally, instead of a set of hopping sequence, dynamic coordination between nodes exists for selection of next channel. Besides, it is easy to stay in a single channel and send a burst of data to overcome channel hopping cost, as chosen by them. Yet, we restrict to single packet exchange per visit, that can be modified to multiple data packets anytime, to increase overall throughput.

While using multiple channels, the channel assignment strategy is critical. [11] proposes a cross-layer approach for channel assignment using a multichannel MAC protocol, with the help of two transceivers on each node. [3] describes the already fed function to be a simple linear formula, like selection of next channel as the new channel. The resistance to jammers does not lie in the complexity of the function rather in the available channels and the effectiveness of channel hopping drops if more than one jammer exist [12]. Theoretically, for a wireless network using 'k' channels, there must be at least 'k' jammers to break down the network completely; one for each channel.

III. The proposed solution

As it is said that prevention is better than cure, similar is our proposed solution to mitigate a jamming attack using proactive channel hopping in an ad hoc network. Every node selects its control channel through a predefined function which is known to network participants. Furthermore, data transfer takes place on a different channel coordinated dynamically between each pair of nodes. Thus, every node communicates with each of its neighbors on different channels. From the network level point of view, every channel can be used for control and data packets simultaneously. In the sequel, we discuss the major design aspects of our proposed scheme.

A. Determining Control Channel (CC)

In the formation of an ad hoc network every node selects its own control channel, docks itself there and waits for other nodes to visit it (if it has no packets to send). The control channel is selected via a pre-loaded function based on the node identity which is shared among nodes and is kept secret. This way, each node not only selects its control

channel but also learns about the control channel where the intended destination is residing, if a transmission needs to be initiated. To avoid an outsider from targeting a particular node or legitimate communication, the function can be a high level polynomial which is hard to break by overhearing the traffic. However, for the sake of simplicity we are incorporating a simple function. Hence, a neighboring node which wants to initiate communication with node 'A' can determine its control channel using the following function:

$$CC(A) = I_A \, mod \, k \tag{1}$$

where k being the total number of channels and I denotes the identity of the intended receiver. This way, n nodes will be distributed over k channels evenly, having (n/k) nodes on the same channel, on average. To minimize the computation overhead and avoid the re-computation of the same function, once determined, the resultant channel is stored in a CC-table for future correspondences. Additionally, since nodes need to visit other channels which are the control channels of intended destinations. Therefore, each node will maintain its own control channel in the CC-table as well which will be referred while returning after a successful data transfer. Hence, sender node first checks a corresponding entry for the intended destination in CC-table. If not found, the corresponding control channel is calculated using equation-1 only once for each node. We are not taking energy consumption into account in this study, but due to frequent channel hopping, which has its own delay, the computation delay is reduced. Thus, to get an estimation of how much computation is saved, we consider that if node 'A' intends to communicate with its neighbor node B for m packets and that only a single packet is exchanged in each visit. Then (m-1) computations for locating the control channel of node B and similar number of computations for returning to its own control channel are avoided by node A. If we extend the situation to multiple destinations, say node A sends m packets to j neighbors then A saves the amount of computation for (j $\times$ m) packets

$$(j \times (m-1)) + (j \times m - 1)$$

where the first half of the equation refers to the CC of the destination node and second half is representing return to its own control channel after each sent packet. So, for n nodes having large 'm' packets to send in the network, quite a computation overhead is diminished.

Once the node knows the receiver's control channel, it hops to the corresponding channel where both nodes agree upon a new channel chosen for data exchange. Since, a single channel can be used for data and control messages by different pair of nodes. So, the newly arriving node on the control channel of intended receiver may disrupt an on going communication. Therefore, it needs to contend for the medium in the next slot to initiate its communication formally. Similar situation is depicted in Figure 2 where node D hops to CC(C), where already a communication between node A and B is in progress. Node D senses the medium busy and consequently it keeps

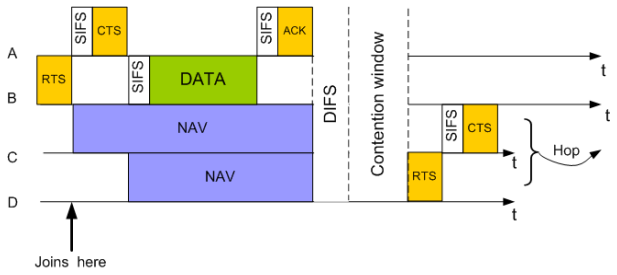

Figure 2: Scenario stating how node D would initiate communication with node C

silence till the end of the on-going transmission. Later, it contends for the medium with other nodes.

B. Data Channel (DC) Coordination

Once the sender hops to the control channel of the intended receiver, rather than they initiate data exchange both coordinate for a data channel. To have different data channels between each node pair, the channel is selected using the identities of both parties. As earlier for equation 2, the complexity of the function will not yield much difference except the increase of the computational time only, due to the limited number of channels. Therefore, the current channel (i.e. control channel of the receiver) is also taken into account. This way, it becomes hard for an attacker to guess and target a particular communication having knowledge of the node identities. Moving a step further, since the nodes need to coordinate data channel over an insecure medium, we tailor a key exchange scheme to incorporate our desired coordination, securely. For this purpose, the data channel coordination is based on Diffie-Hellman algorithm [1] as shown in Figure 3 and described as follows:

The sender initiates the coordination by choosing a secret random exponent 'a' and yields

$$X = g^a \, mod \, p \tag{2}$$

where g is a publicly known constant [15] and p is a prime number selected using the identities of sender and receiver along with the current channel number in use as follows:

$$p > f(ID_{receiver}, ID_{sender}, CC_{receiver}) \tag{3}$$

X is then sent to the receiver by piggybacking it to the RTS frame. On receiving this RTS, the receiver generates Y, same as the sender yielded X, with the help of its secret random exponent b. The receiver then responds by sending CTS piggybacked with Y. Both parties then apply their respective secret random exponent on the information received from the other, to yield the same value:

$$Y^a = X^b = Z \, mod \, k \tag{4}$$

Thus, Z is the newly selected channel for data exchange. Both nodes store it in a DC-table for future reference and switch their transceivers accordingly to initiate data transfer

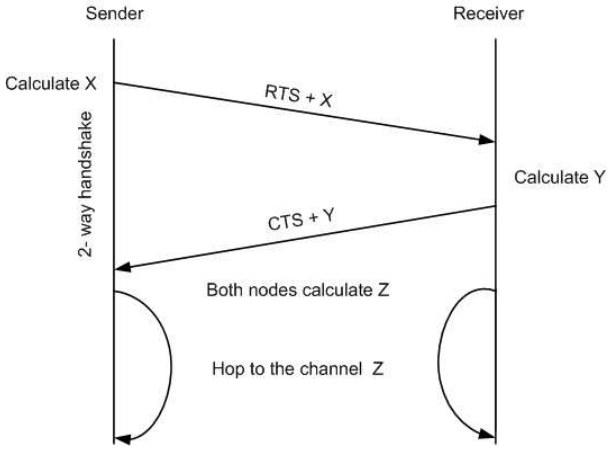

Figure 3: Elementary Negotiation for a DC between two nodes

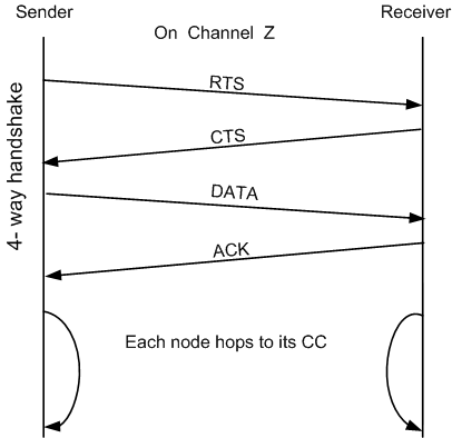

Figure 4: Communication Sequence on Data Channel between a node-pair

on the new channel. Regardless, the data is exchanged successfully or it times out due to unavailability of the medium, in either case nodes will return to their respective control channels.

When the nodes hop to channel Z for data exchange to avoid hidden terminal problem on the new channel, they need to exchange RTS-CTS once again [4]. If successfully exchanged, data and acknowledgement follow as shown in Figure 4.

For each neighbor, the above sequence is followed only once and the coordinated data channel is stored in DC-table. For the subsequent visits, only simple RTS-CTS are exchanged, both parties refer their DC-table and hop for data exchange. If a node has more than one visitor on its control channel, it will choose and accord with one only. The others will wait for receiver to return to its control channel for their turn, till they time out.

IV. Simulations and Analysis

This section presents the simulation scenario and results obtained using the OPNET network simulator [14]. The simulation parameters are summarized in table I. We consider an ad hoc network consisting of 24 wireless stations with single transceiver only. All nodes are in the transmission range of each other, i.e. 1-hop neighbors. To incorporate a saturated case, the traffic load at the source nodes is 200 pps. whereas the packet size is chosen to be 512 bytes, each. Only a single packet is exchanged in each visit between a node-pair. The physical layer standard taken into consideration is 802.11a which offers 12 channels in 5 GHz band. For channel hopping the cost currently considered by different studies is between 40 to 80 μs, so we opted for 80 μs. delay, similar to [11] and [5]. Besides these, the jammer is located in the center and all nodes are in the jamming range and we assume that no communication takes place in this range on the jammed channel, thus the intensity of jammer is set accordingly. Since, the jammer is considered as an outsider and on any channel it tunes into, it is able to listen to legitimate traffic; either control or data packets. Therefore, the underlying assumption is that the jammer sticks to such a channel where its intention to block lawful conversation is fulfilled and by doing so it does not scan other channels. Hence, in our simulation environment the jamming attack is launched in the form of a constant jammer which sticks to a single frequency.

Initially, we compare single channel with multi-channel proactive hopping for throughput and later with the presence of jammer in both cases. The network is divided such that half of the nodes are traffic sources and the rest are treated as sinks. In a single channel environment, nodes and the jammer are situated on the same channel. However, for multichannel scenario sink nodes are evenly distributed so that one on each channel resides and the remaining are selected as source. Thus, one sender and receiver reside in each channel, but the communication pair are chosen from different control channels to incorporate channel hopping even for the exchange of control packets. For 100 seconds simulation time, the jammer is active from 20 to 80 seconds. Figure 5 shows that in a single channel scenario no legitimate communication is observed during jamming phase. However for multi-channel setup with our proposed scheme a couple of node pairs out of 12 are affected, i.e. approximately 17% degradation in overall network performance is observed when a single channel is exposed to jammer. The degradation is due to the effect on node-pairs having their control or data channel being jammed, depending upon the channel selected by the jammer. The peak found in the curve is due to those packets that are not discarded by that time and were successfully retransmitted after the jamming phase is over. It varies depending upon the number of nodes and traffic load.

Figure 6 shows the average throughput at the sinks in our scheme. Nodes having the jammed channel as control and data channel face nearly the same degradation in throughput on average and are therefore represented using different colors. However, if the jamming intensity is decreased or rather than constant a periodic jammer is incorporated then the difference will be more evident. For this reason, a pulse jammer, which disrupts the communication periodically for sometime and sleeps during the two jamming intervals, is substituted. The

Simulation parameters	Parameter value
Physical Layer Standard	802.11a
Number of Channels	12 (in 5 GHz band)
Traffic type	CBR
Packet Size	512 Bytes
Traffic Load	200 packets/sec (pps)
Simulation Time	100 sec.
Jammer Type	Constant Jammer
Jamming Period	20 – 80 sec.

Table I: Simulation settings

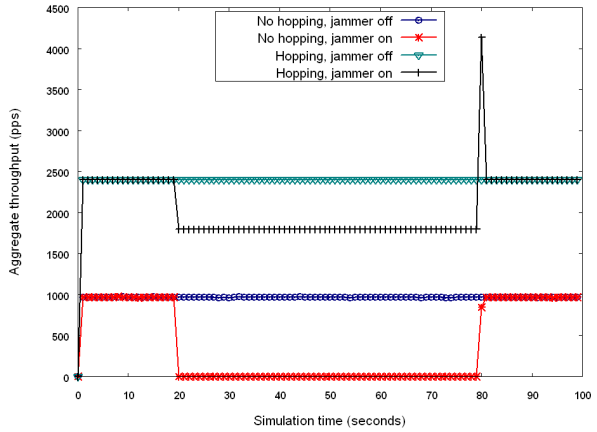

Figure 5: Single channel compared with the proposed scheme using 12 Node-pairs with traffic load 200 pps, where the jammer is active during 20–80 seconds

jamming period chosen is 100 ms. and sleep time as 2 seconds, alternatively. For a 100 seconds of simulation the effect of pulse jammer on the affected nodes is shown in Figure 7. This figure gives a view of data packets corrupted and control packets targeted by pulse jammer. The difference is evident in terms of plunges in the curves, found more in case of control channel being jammed. However, with the increase in jamming intensity the difference is diminished and both nodes may starve, as presented earlier.

Next, to have a more realistic scenario rather than divide the topology into active source and sinks, the status of all nodes is modified so that each one is sender and receiver at the same time. However, while node A is seeking for node B, B may be visiting some other nodes or A at the same time. Such a situation can give rise to a deadlock and increased packet drops and is thus considered as worst case. To incorporate the worst case scenario and analyze its effects, the earlier sinks are therefore changed to senders for different traffic generation rates, but others are kept unchanged. i.e. all nodes send and receive simultaneously, as generally observed in manets. Initially, the traffic load on new sources is kept low to have a better view of network traffic degradation due to synchronization issues, when both nodes try to reach another to deliver their packets. Thus, we start the traffic load on new sources from 10 pps and gradually increase it to 100 pps, so that all nodes generate similar number of packets.

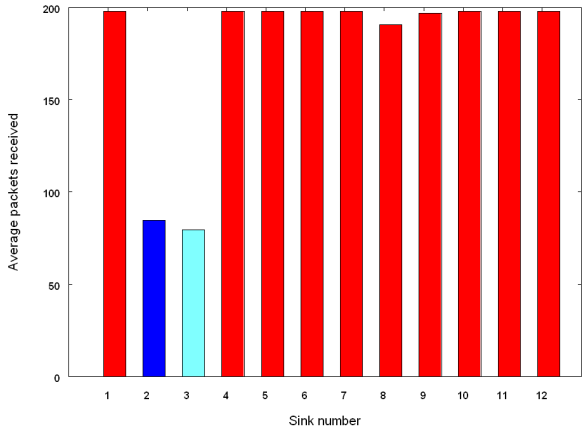

Figure 6: Sink Status on each channel – Nodewise distribution

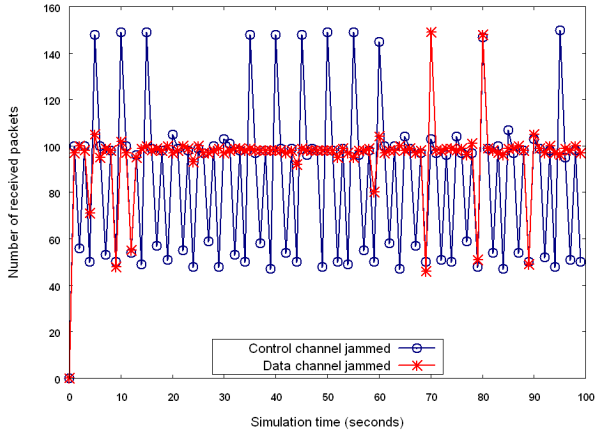

Figure 7: Effect of Pulse jamming on nodes having jammed control and data channels

For a traffic load between 10 to 100 pps for the new source nodes, the network along with the jammer was tested as shown in Figure 8. In the worst case scenario, approximately 20% decline is observed as compared to simpler scenario of a single channel only as considered in Figure 5. However, the overall throughput decreases with increase in traffic load and only 40% of legitimate communication is successful when all nodes have similar configuration, which declines further in the presence of jammer to 25% of the generated traffic (only 600 out of 2400 pps are successfully received). But the jamming phase added pain to the sickness as the lost packets are doubled from the earlier scenario, to nearly 35%. Above all, we treat the lastly presented scenario where all nodes are the senders as worst case due to the fact that when node A is sending to B the data channel would be different than B is sending to A. Thus, increasing the number of affected data channels and nodes.

As compared to other proactive channel hopping schemes like the scheme proposed in [8], the performance drop was reported as 60% due to jamming, in WLAN context. Moreover,it is observed that the performance is estimated to decrease

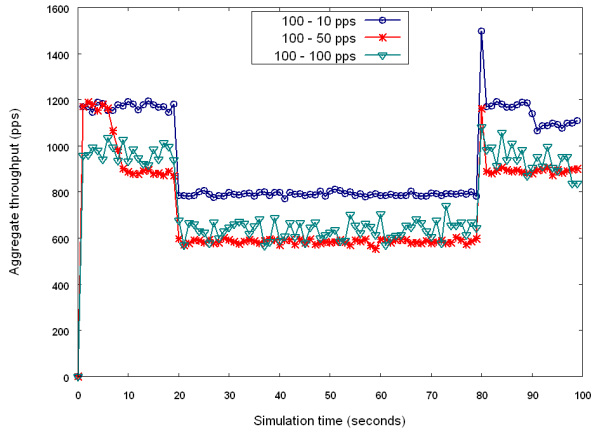

Figure 8: Two way communication between each node-pair with varied traffic generation rates. Jammer is active from 20 to 80 sec.

further in the case of an ad hoc network. However, with our proposed scheme in worst case scenario 65% of the network performance is still retained. Additionally, if we apply the similar jammer configuration with listen and jam intervals, slight improvement in the achieved results are expected. Further improvement is expected in our scheme if burst of packets, rather than only one packet, are exchanged in each meeting between each node pair. Yet, the intention of this study is to explore the jamming effects on proactive channel hopping only and analyze future directions for its mitigation.

V. Conclusion

Channel hopping is considered a logical escape from the jammer, either in a reactive or proactive manner. The proposed channel hopping scheme differs from the already existing solutions in a sense that separate control and data channels exist and neighbors coordinate for their corresponding data channels. Neighborhood communication of a node can be described as flower-petals, each of different color representing a distinct channel for each neighbor. Above all, the scheme is proactive in nature which reduces the impact of a jamming attack without using any detection mechanism by providing already existed escape doors for a node. Initially, we have incorporated simpler scenarios for the ease of analysis and highlighting the jamming effect on our scheme and then gradually moving to worst case scenario, involving synchronization issues along with the jamming phase. The obtained results show that our scheme is efficient for an ad hoc network, as compared to other proactive schemes. Yet, the focus of this study is to analyze proposed scheme in terms of jamming attack. It will help us in developing a robust solution to counter the jammer more effectively in the future.

References

[1] W. Stallings, "Cryptography and Network Security: Principles and Practice", 3_{rd} Edition, Prentice Hall, 1998.

[2] IEEE 802.11, 1999 Edition (ISO/IEC 8802-11:1999). IEEE Standards for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[3] W. Xu, T. Wood, W. Trappe and Y. Zhang, "Channel surfing and Spatial Retreats: Defenses against Wireless Denial of Service", *in Proc. of the ACM Workshop on Wireless Security*, Philadelphia, PA, USA, Oct. 2004.

[4] J. So and N. H. Vaidya, "MultiChannel MAC for Ad Hoc Networks: Handling MultiChannel Hidden Terminals Using A Single Transceiver", *in Proc. of ACM MobiHoc*, Tokyo, Japan, May 2004.

[5] P. Bahl, R. Chandra and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks", *in Proc. of ACM MobiCom*, Phliadelphia, Pennsylvania, USA, Sep. 2004.

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

[7] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming Sensor Networks: Attack and Defense Strategies", *IEEE Networks: Special Issue on Sensor Networks*, Vol. 20, No. 3, pp. 41-47, May/June 2006.

[8] V. Navda, A. Bohra, S. Ganguly and D. Rubenstein, "Using Channel Hopping to Increase 802.11 Resilience to Jamming Attacks", *in Proc. of the* 26^{th} *IEEE International Conference on Computer Communications*, Anchorage, Alaska, USA, May 2007.

[9] A. D. Wood, J. A. Stankovic and G. Zhou, "DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.1-based Wireless Networks", *in Proc. of the* 4^{th} *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'07)*, San Diego, California, USA, Jun. 2007.

[10] L. Ma, C.C. Shen and B. Ryu, "Single-Radio Adaptive Channel Algorithm for Spectrum Agile Wireless Ad Hoc Networks", *in Proc. of* 2^{nd} *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN '07)*, Dublin, Ireland, April 2007.

[11] Michelle X. Gong, Scott F. Midkiff and Shiwen Mao, "A Cross-layer Approach to Channel Assignment in Wireless Ad Hoc Networks", *Journal of Mobile Networks and Applications*, Vol. 12, No. 1, pp. 43-56, Feb. 2007.

[12] K. Bicakci and B. Tavli, "Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks", *Elsevier journal of Computer Standards & Interfaces*, Article in Press.

[13] J. Mo, H.-S. W. So, and J. Walrand, "Comparison of multi-channel MAC protocols", *IEEE Transactions on mobile Computing*, vol. 7, no. 1, pp. 50-65, Jan. 2008.

[14] OPNET Modeller, *http://www.opnet.com*.

[15] *http://en.wikipedia.org/wiki/Diffie-Hellman*.