# Jamming ACK Attack to Wireless Networks and a Mitigation Approach

Zhiguo Zhang[†] Jingqi Wu[†] Jing Deng[‡] Meikang Qiu[*]
[†]Dept. of Computer Science, University of New Orleans, New Orleans, LA 70148, USA
[‡]Dept. of Computer Science, University of North Carolina at Greensboro, Greensboro, NC 27402, USA
[*]Dept. of Electrical Engineering, University of New Orleans, New Orleans, LA 70148, USA

***Abstract*— In many Medium Access Control (MAC) schemes for wireless networks, an Acknowledgment (ACK) packet is transmitted from the data receiver to the data sender to announce the successful reception of the data packet. Such a protocol requirement may become a system weakness when malicious nodes attack these wireless networks. In this paper, we demonstrate the effects of such a Jamming ACK (JACK) attack to networks employing the popular Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme in IEEE 802.11 DCF. Our study shows that a JACK attacker can easily disrupt the traffic flow between two wireless nodes when it sends out JACK packets at the right time. The benefits of such a JACK attack include low energy consumption by the attacker, attack stealthiness, and great damage to the victim nodes. To mitigate the effects of JACK attacks, we propose in this paper an Extended Network Allocation Vector (ENAV) scheme. Our analysis and simulations show that the ENAV scheme recovers a significant portion of the lost throughput and reduces the energy drainage of the attacked nodes to 40%.**

## I. Introduction

One of the main differences between wireline and wireless computer networks is the transmission medium. Rather than keeping signal within cables, wireless communication transceivers use open media. Many transceivers nearby share the same medium. When two signals arrive to a receiver at the same time, the receiver is unlikely to pick up either unless signal capture occurs.

Medium Access Control (MAC) schemes are designed to reduce such collisions and to improve channel usage efficiency. In most of these MAC schemes, an Acknowledgment (ACK) packet will be transmitted from the data receiver to the data sender after the data packet is successfully received. An example is the widely implemented IEEE 802.11 DCF that employs Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) (we will show that most of these discussions apply to other MAC schemes with ACK packets). In CSMA/CA, each pair of hosts will go through the process of Request-To-Send packet, Clear-To-Send packet, Data packet, and ACK packet (RTS/CTS/DATA/ACK) to reserve and to use the medium exclusively. Besides using physical carrier sensing, the CSMA/CA scheme employs a virtual carrier sensing technique with the help of Network Allocation Vector (NAV). NAV indicates how long the sender and the receiver reserve the medium. Neighbors overhearing the NAV information are required to keep silent until the NAV expires.

Many hosts in wireless networks such as Mobile Ad-hoc Networks (MANETs) are powered by batteries. How to make good use of the limited energy is always one of the top concerns. In some applications, such as battlefield, hosts in MANETs will face many adversaries. Besides jamming the medium and preventing targeted hosts from using it, attackers may also try to drain the battery energy of the victim nodes as quickly as possible.

In this paper, we investigate a potential attack, Jamming ACK (JACK) attack, to wireless networks. JACK attackers basically send out packets to collide with legitimate ACK packets so that the data sender will need to reschedule the data transmission. Therefore, JACK attack can be used by adversaries to drain the battery energy of victim nodes and this is achieved with a small amount of energy. In order to mitigate the effect of the JACK attacks, we propose an effective countermeasure called Extended NAV (ENAV). The basic idea of ENAV is to extend the ACK packet transmission window so that it becomes more difficult for the JACK attackers to jam the legitimate ACK packets.

Our paper is organized as follows: In Section II, related work is introduced; The JACK attack and our ENAV mitigation technique are presented in Sections III and IV; In Section V, theoretical evaluation and NS2 simulation are shown on the performance of ENAV; Finally, Section VI concludes our work.

## II. Related Work

The security problem in wireless networks such as MANETs has attracted a lot of research interests [1]–[3].

In [4], Kyasanur and Vaidya investigated the misbehavior of selfish nodes that intentionally disobey the MAC protocol rules in IEEE 802.11 networks. These misbehaving hosts may wait for smaller back-off intervals to gain unfair share of the channel compared to other well-behave hosts. A protection scheme was presented to detect and penalize any selfish misbehavior. In this scheme, the receiver selects a random back-off value and sends it in the CTS and the ACK packets to the sender. The sender must use this assigned back-off value in its next transmission to the receiver. The receiver observes the back-off time between consecutive transmissions from the same sender and judges whether the sender is deviating from the protocol. The application of the proposed scheme

in networks with more than one receiver is however more difficult.

Virtual carrier-sense is used to determine the availability of the shared medium in the IEEE 802.11 MAC protocol. In [5], Chen et al. investigated the vulnerabilities that a misbehaving node may exploit to block neighboring nodes from accessing medium for an extended period of time. Two potential virtual jamming attacks were discovered. A backward-compatible solution, NAV Validation, was designed to overcome these vulnerabilities. The main idea of NAV Validation is to set two MAC-layer timers: one timer monitors the duration between RTS packet and DATA packet; the other monitors the duration between CTS packet and ACK packet. The two timers help to double-check whether the DATA and ACK packets appear as expected. Besides the virtual jamming attacks, the hosts may also suffer from physical jamming attacks such as the Jamming ACK attack that we discuss in this paper.

In [6], Gupta et al. showed an IEEE 802.11 MAC protocol weakness and how it may be exploited to launch Denial-of-Service (DoS) attacks in wireless Ad Hoc environment in different ways. They investigated different types of attacks at the routing layer and data link layer. At routing layer, the attacks include simply dropping a certain number of the data packets, transmitting falsified route updates, replaying stale updates, and reducing the Time-to-Live (TTL) field in the IP header. At the MAC layer, they focused on the attacks such as keeping the channel busy in the vicinity of a host and draining the battery energy of a host by continuously asking it to relay spurious data. They concluded that MAC layer fairness is insufficient to alleviate the effects of various types of DoS attacks.

Xu et al. [7] investigated DoS attacks at MAC layer in wireless networks. Four types of attacks were categorized: constant jammer, deceptive jammer, random jammer, and reactive jammer. A constant jammer continuously emits a radio signal. It is effective but inefficient. A deceptive jammer constantly injects regular packets without any gap. A random jammer alternates between sleeping and jamming. A reactive jammer stays quiet when the channel is idle, but starts transmitting a radio signal as soon as it senses activity on the channel. In [7], several detection techniques were proposed and evaluated. The JACK attack that we focus on can be classified as from reactive jammers. The difference is that the JACK attackers do not jam the data packets but just the expected ACK packets.

In [8], Goldsmith and Wicker summarized several wireless attacks discovered by other researchers and implemented them with the help of an "aux port", an unbuffered unsynchronized raw memory access interface for debug purpose. Most of the implemented attacks were shown to be successful, underlining the necessities to detect any of these attacks when such networks are employed in mission critical applications.

In [9], Acharya et al. categorized jamming attacks into three categories: trivial jamming, simple periodic jamming and intelligent jamming, where the ACK corruption jamming was classified as intelligent jamming. The jamming efficiency was compared: the simple periodic jamming can be three to four orders of magnitude more effective than trivial jamming while intelligent jamming can be five orders of magnitude more effective than the trivial jamming.

## III. The JACK Attack

We illustrate the JACK attacks in this section. Our explanations are based on IEEE 802.11 DCF MAC, CSMA/CA, but they fit most MAC schemes for wireless networks.

Based on the operational rules of the MAC schemes in MANETs, such as IEEE 802.11 DCF, all data packets need to be acknowledged before they are cleared from the queue. An attacker simply sends wireless signal to jam ACK messages in the network. Such jamming signal ruins the reception of ACK message at the sender of the data packet as long as the attacker locates within the sender's range. This is illustrated in Fig. 1. When the ACK packet is jammed by attackers, data retransmissions will be scheduled. Such retransmissions will fail in a similar fashion. Let $N_{t\_max}$ $(N_{t\_max} \geq 1)$ denote the maximum transmission limit of one DATA packet, which means the sender will retransmit it for at most $N_{t\_max} - 1$ times. Data packets will simply be dropped once the sender reaches the retransmission limit.

An interesting observation to the victims of such an attack is that they consume more energy in vain in order to make sure that the data packets are transmitted and acknowledged successfully, i.e., ACK packets are expected to arrive after successful data transmissions.[1] Hence, the attacks effectively cost the victims extra energy by jamming a short control packet, the ACK packet. In addition, such retransmissions block the channel from transmitting other data packets, reducing maximum achievable throughput. Note that the data receiver has already received the DATA packet even as the ACK packet suffers from collisions. We term the potential attack Jamming ACK (JACK) attack. The adverse effects of the JACK attacks can be summarized as follows:

- Higher energy cost of the data sender;
- Lower maximum achievable throughput;
- Difficulty to detect attacks and identify attackers.

One way to launch the JACK attack is the following: the attacker tries to overhear on the shared channel and wait for any DATA packet from the sender. Once a DATA packet is overheard, it waits for a period of Short InterFrame Spacing (SIFS) and sends out the JACK packet. In fact, any packet sent by the JACK attacker ruins the reception of the legitimate ACK packet. There are two reasons why we consider JACK attackers sending only one full ACK packet.

- The size of ACK is relatively small, and it costs small amount of energy for the attacker. In fact, the ACK packet has the smallest size among the RTS/CTS/DATA/ACK packets (CTS packet is also the same size).
- Repeated transmission of ACK packets, partial packets, or jamming signals from the JACK attacker may raise suspicion from other nodes. For instance, two consecutive

[1]Senders of broadcast messages do not wait for ACK packets. Therefore, this attack will not affect broadcast transmissions.

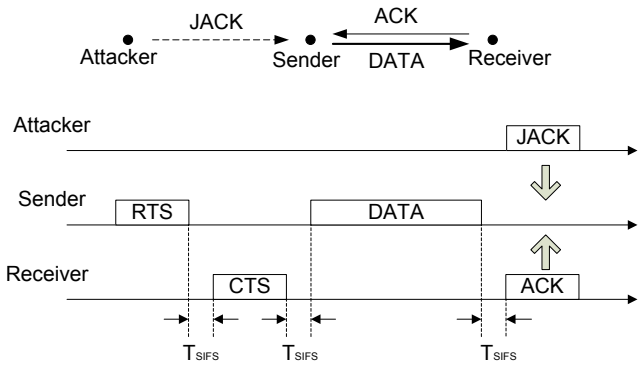


Fig. 1. Illustration of the JACK attack. After sending a data packet to the receiver, the sender expects to receive an ACK packet from it. The attacker sends a JACK packet to collide with the ACK packet at the data sender. $T_{SIFS}$ represents the Short InterFrame Spacing time.

ACK packets from the same receiver usually have an interval of at least one DATA packet transmission time.

Note that jamming other packets may not lead to such a high energy drainage from the victim nodes. For instance, jamming the RTS/CTS packets only leads to retransmission of such control packets, which is less effective compared to forcing data retransmissions.

## IV. The ENAV Scheme

In this section, we introduce a scheme to mitigate the adverse effect of JACK attacks. The main idea of our scheme is an extension of the ACK transmission window and random transmission time over this period. This technique is termed Extended NAV (ENAV), illustrated in Fig. 2. In the ENAV scheme, the ACK transmission window is extended from $T_{ACK}$ to $R \cdot T_{ACK}$, where $T_{ACK}$ is the ACK transmission time. By extending the window of sending/receiving the ACK packet, the data sender has a better chance of receiving the ACK packet from the data receiver. Obviously, when R = 1, a MAC scheme implementing the ENAV scheme degenerates to the original MAC scheme. Note that a larger $R$ increases channel reservation time for the same data packet transmission while it improves the ACK reception chance. We analyze optimal $R$ in Section V.

While the NAV values carried on the RTS and the CTS packets change from one transmission to another due to the variable DATA packet length, the NAV value carried on DATA packets is usually fixed at $T_{SIFS} + T_{ACK}$. A JACK attacker may notice the extension of the NAV value and hence try to send its JACK packet to collide with the ACK packet in the extended period.[2] Since the JACK attacker cannot guess when the ACK packet will be sent from the data receiver, the best option that it has is to send at a randomly-chosen time between $[0, (R-1) \cdot T_{ACK}]$. With ENAV, the receiver will delay for a random period within $[0, (R-1) \cdot T_{ACK}]$ after the complete reception of DATA packet and a $T_{SIFS}$ period.

[2]An extension of JACK packet will run the risk of exposing the attacker. It will also consume more energy from the attacker.

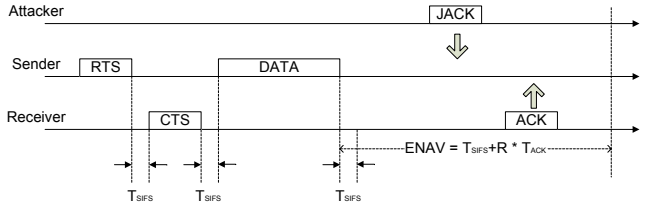


Fig. 2. Illustration of the ENAV scheme. The ACK transmission/reception window is extended from $T_{SIFS} + T_{ACK}$ to $T_{SIFS} + R \cdot T_{ACK}$, where $R > 1$.

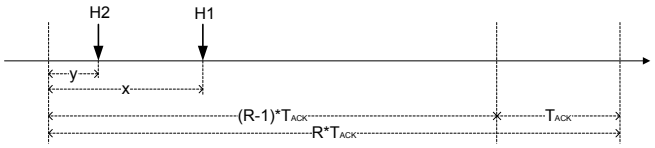


Fig. 3. Derivation of collision probability in the ENAV scheme, $P(R)$. H1 and H2 represents the beginning of the ACK and JACK packets. If $|H1 - H2| < T_{ACK}$, collisions occur.

## V. ENAV Performance Evaluation

In this section, we analyze the effect of $R$ in the ENAV scheme on throughput. These results are then compared to simulation results from NS2.

### A. Analysis

Denote the probability that an ACK packet collides with a JACK packet $P(R)$ for a given $R$. Clearly, $P(R) = 1$ if $R < 2$. So we focus on the situation where $R \geq 2$.

Assuming that both the ACK and the JACK packets are to be transmitted randomly within this $R \cdot T_{ACK}$ period of time, the beginning of the ACK packet, H1, and the beginning of the JACK packet, H2, will be randomly chosen from the period between 0 and $(R-1) \cdot T_{ACK}$, as shown in Fig 3. The Probability Density Function (PDF) of H1, $x$, and H2, $y$, is:

$$f(x) = f(y) = \frac{1}{(R-1)T_{ACK}} \tag{1}$$

where $0 \leq x, y \leq (R-1)T_{ACK}$.

Collisions of the ACK and the JACK packets occur if

$$|H1 - H2| = |x - y| < T_{ACK} \tag{2}$$

Therefore, the probability that these two packets collide with each other can be calculated as (3). We assume $R > 3$ (the $2 \leq R \leq 3$ case is similar but omitted due to space limit).

$$\begin{aligned} P(R) = & \int_0^{T_{ACK}} f(x) \int_0^{x+T_{ACK}} f(y)dydx \\ & + \int_{T_{ACK}}^{(R-2)T_{ACK}} f(x) \int_{x-T_{ACK}}^{x+T_{ACK}} f(y)dydx \\ & + \int_{(R-2)T_{ACK}}^{(R-1)T_{ACK}} f(x) \int_{x-T_{ACK}}^{(R-1)T_{ACK}} f(y)dydx \\ = & \frac{2R-3}{(R-1)^2} \end{aligned} \tag{3}$$

TABLE I

SIMULATION PARAMETERS

| | | | |
|---|---|---|---|
| NIC:dataRate | 11 Mbps | $B(1)$ | $320\mu s$ |
| NIC:basicRate | 4 Mbps | $B(2)$ | $640\mu s$ |
| CBR:rate | 4 Mbps | $B(3)$ | $1280\mu s$ |
| CBR packetSize | 1 Kbytes | $B(4)$ | $2560\mu s$ |
| $N_{t\_max}$ | 4 | $T_{RTS}$ | $232\mu s$ |
| $T_{slot}$ | $20\mu s$ | $T_{DATA}$ | $954\mu s$ |
| $T_{SIFS}$ | $10\mu s$ | $T_{CTS}$ | $220\mu s$ |
| $T_{DIFS}$ | $50\mu s$ | $T_{ACK}$ | $220\mu s$ |

We explain the calculation as follows: the overall possible $(R-1)T_{ACK}$ transmission window is divided into three parts: $[0, T_{ACK}], [T_{ACK}, (R-2)T_{ACK}]$, and $[(R-2)T_{ACK}, (R-1)T_{ACK}]$. The three terms in (3) calculate the chance that $x$ falls in the three parts, respectively, and (2) is satisfied.

When $n$ (re)transmissions are sent for one data packet, the overall transmission time of this data packet can be calculated as T(R, n):

$$T(R,n) = T(R,n-1) + B(n) + T_1 + R \cdot T_{ACK} \quad (4)$$

where $n \geq 1$, $T_1 = T_{DIFS} + T_{RTS} + T_{CTS} + T_{DATA} + 3T_{SIFS}$, $B(n)$ is the average back-off time in the $n$-th (re)transmission, and $T(R,0) = 0$.

Considering the probability of collisions, the average overall transmission time of a data packet is $T(R)$ ($N_{t\_max}$=4):

$$T(R) = \sum_{n=1}^{3} T(R,n) \cdot [P(R)]^{n-1} \cdot [1-P(R)] + T(R,4) \cdot [P(R)]^3 \quad (5)$$

We need to derive $B(n)$ in (4). $B(n)$ represents the average back-off time of each (re)transmission. Since the back-off timers are chosen randomly from the Contention Window (CW), $B(n) = CW(n)/2$.

The throughput of a network with ENAV employed can be estimated based on packet length and $T(R)$. One interesting observation of such a network is that data packet is successful in all (re)transmissions. It is because of the JACK attacks and the ACK packet collisions that prompt the sender to retransmit. For example, when the data packet length is $L$ bytes ($8L$ bits), the throughput can be expressed as:

$$S(R) \approx \frac{8L}{T(R)} \quad (6)$$

A numerical optimization of the throughput based on $R$ is possible. Our derivations show that a maximum throughput may be achieved with $R = 7.5$. This number will be shown to match well with our simulation results in Section V-B.

### B. NS2 Simulation

In order to show the different effects of the JACK attack and the ENAV scheme. We carried out simulations for the following 4 scenarios.

- (*normal*) network without JACK attack and ENAV scheme.
- (*JACK only*) network with JACK attack but without ENAV scheme.
- (*JACK+ENAV*) network with both JACK attack and ENAV scheme.
- (*ENAV only*) network with ENAV scheme but without JACK attack.

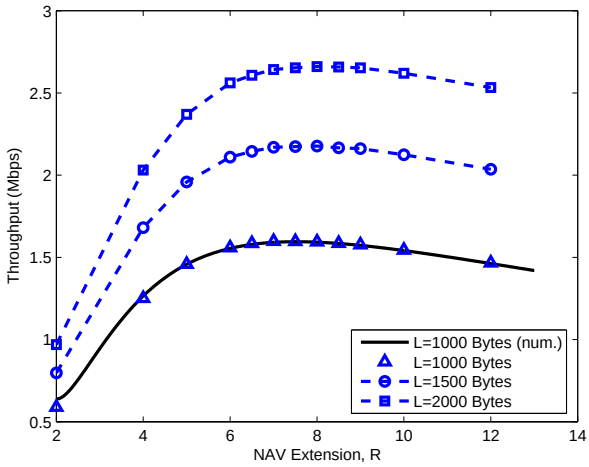


Fig. 4. Throughput comparison of a network suffering JACK attacks and with the ENAV scheme implemented. The solid line represents our numerical results based on Table I and (6).

All the remaining network parameters are shown in Table I. Unless specified otherwise, the Constant Bit Rate (CBR) packet size is 2000 bytes and $R$ in the ENAV scheme is 7.

We show the throughput performance of the simple three-node network in Fig. 4. In the three-node network, a pair of nodes serve as the sender and receiver. The third node is only neighboring to the sender and it serves as the JACK attacker (see Fig. 1). The throughput of the network with JACK+ENAV is shown in Fig. 4 as a function of different $R$ in the ENAV scheme.

A solid line in Fig. 4 represents our numerical results based on Table I and (6). The numerical results match well with NS2 simulation results. We also present the simulation results of data packet length of 1500 bytes and 2000 bytes. As data packet length increases, the network throughput improves. Based on the simulation results, we can observe that the optimal value of $R$ is about 7.5 in the network that we studied. Such an optimal $R$ that maximizes the maximum achievable throughput remains the same for different data packet lengths.

The simulation results of different traffic load are shown in Fig. 5. The maximum throughput of a normal network is 6 Mbps. When the network is under JACK attacks, the maximum throughput reduces to 1 Mbps ($15\%$ of the throughput of the normal network). With the help of the ENAV scheme, the maximum achievable throughput is recovered to the level of 2.5 Mbps. For comparison purposes, we also show the throughput of a network when the ENAV scheme (with $R = 7$) is implemented. The throughput of such an ENAV only network is about 4 Mbps. The lowered throughput is due to the

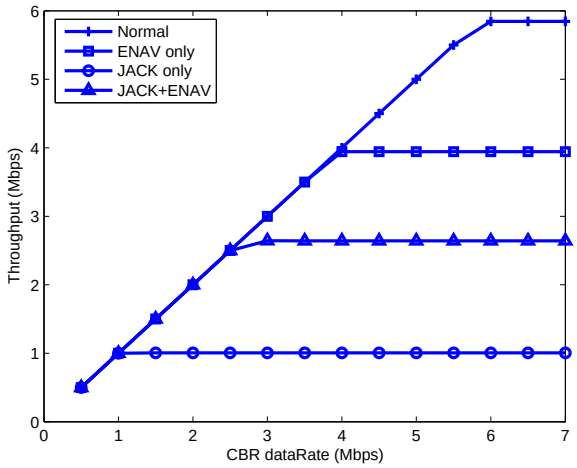


Fig. 5. Throughput comparison of normal, JACK only, ENAV only, and JACK+ENAV networks. As the traffic load (CBR datarate) increases, the throughput increases until reaching the maximum achievable throughput in each scenario.

TABLE II
LUCENT IEEE 802.11 WAVELAN PC CARD

| | |
|---|---|
| Transmission Speed | 11Mbps |
| Power Supply | 4.74V |
| Sleep Mode Current | 10mA |
| Sleep Mode Power | 47.4mW |
| Idle Mode Current | 156mA |
| Idle Mode Power | 739.44mW |
| Receive Mode Current | 190mA |
| Receive Mode Power | 900.6mW |
| Transmit Mode Current | 284mA |
| Transmit Mode Power | 1346.16mW |

additional channel usage during the extended NAV period in ACK packet transmission/reception. Note that this throughput can be improved with a lowered $R$, which is possible when the sender/receiver notice no ACK packet loss.

We studied the energy consumption of the nodes in the above four scenarios. The wireless interface cards of sender, receiver, and attacker are assumed to have the specifications as shown in Table II [10]. Table III compares the average energy consumption for each CBR packet in normal, ENAV only, JACK only, and JACK+ENAV networks. The energy consumption for each packet is the lowest in the normal network since there is no attack or extended NAV. When a network suffers from JACK attacks, the sender and the receiver increase the energy consumption to more than 5 times. In the JACK+ENAV network, the energy consumption of the attacked nodes is reduced to $40\%$ of that of the JACK only network. In the ENAV only network, the energy consumption of the sender and receiver increases slightly from the normal network.

TABLE III
ENERGY CONSUMPTION OF EACH DATA PACKET TRANSMISSION (INCLUDING RETRANSMISSIONS). THE UNIT IS IN MJ.

| | normal | ENAV only | JACK only | JACK+ENAV |
|---|---|---|---|---|
| Sender | 3.28 | 4.27 | 16.86 | 6.34 |
| Receiver | 2.63 | 3.62 | 14.23 | 5.39 |
| Attacker | N/A | N/A | 13.67 | 5.25 |

## VI. CONCLUSION

Wireless networks can be under many attacks, especially at the MAC layer. At the MAC layer, wireless networks are more vulnerable than wireline networks. We have investigated the Jamming ACK (JACK) attack to MAC schemes that require the data receiver to return ACK packets to acknowledge the success of data reception. Such JACK attacks may be launched by adversaries to lower the achievable network throughput and to increase the energy consumption by the victim nodes. Our study has shown that a JACK attacker can easily raise the energy consumption of a victim sender by 5 times and reduce its achievable throughput to $15\%$.

We have proposed a solution, termed Extended Network Allocator Vector (ENAV), to mitigate the impact of JACK attacks. With the help of the extended NAV period, the ENAV scheme provides a flexible period for the data receiver to send the ACK packet, significantly reducing the chance of being collided by the JACK attacker. Our analysis and simulations show that the ENAV scheme recovers a significant portion of the network throughput and reduces the energy consumption by the victim nodes to $40\%$.

In our future work, we will investigate JACK and ENAV in networks with more nodes, multiple attackers, and multi-hop networks. Techniques to dynamically change $R$ based on retransmission observation will also be developed and studied.

## REFERENCES

[1] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, November/December 1999.

[2] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues in ad-hoc wireless networks," in *Security Protocols, 7th International Workshop Proceedings*, B. Christianson, B. Crispo, and M. Roe, Eds. 1999, Springer-Verlag.

[3] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," in *Wireless Communication*, Feb. 2004, vol. 11, pp. 38–47.

[4] P. Kyasanur and N.H. Vaidya, "Detection and handling of MAC layer misbehavior in wireless networks," in *Dependable Systems and Networks, 2003*, Jun. 2003, pp. 173–182.

[5] D. Chen, J. Deng, and P. K. Varshney, "Protecting wireless networks against a denial of service attack based on virtual jamming," in *ACM MobiCom '03 Poster Session*, San Diego, CA, USA, September 14-19 2003.

[6] V. Gupta, S. Krishnamurthy, and M. Faloutsos, "Denial of service attacks at the MAC layer in wireless ad hoc networks," in *Proc. of MILCOM '02*, Oct. 2002.

[7] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. of MobiHoc '05*, May 2005, pp. 46–57.

[8] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proc. of 12th USENIX Security Symposium*, Aug. 2003.

[9] M. Acharya, T. Sharma, D. Thuente, and D. Sizemore, "Intelligent jamming in 802.11b wireless networks," in *Porc. of OPNETWORK-2004*, Aug. 2004.

[10] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. of IEEE Infocom '01*, 2001, vol. 3, pp. 1548–1557.