This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Defense Against Injecting Traffic Attacks in Cooperative Ad Hoc Networks

Wei Yu and K. J. Ray Liu
Department of Electrical and Computer Engineering and The Institute for Systems Research
University of Maryland, College Park, MD 20742
Email: weiyu, kjrliu@isr.umd.edu

***Abstract*— In this paper we investigate how to defend against injecting traffic attacks in cooperative ad hoc networks. By injecting an overwhelming amount of packets into the network, the attackers can easily consume valuable network resources and reduce nodes' lifetime. Since in cooperative ad hoc networks nodes will usually unconditionally forward packets for other nodes, such networks are extremely vulnerable to injecting traffic attacks, especially those launched by inside attackers. In this paper, the possible types of injecting traffic attackers are studied, and a set of mechanisms are proposed to protect cooperative ad hoc network against such attacks. The performance of the proposed mechanisms is analyzed, which show that from attackers' point of view, the best strategy is not to launch injecting traffic attacks. Simulation studies also confirm the theoretical analysis.**

I. Introduction

Since ad hoc networks can be easily deployed as needed, they have drawn extensive attentions recently. In many situations, such as military or emergency applications, nodes in an ad hoc network belong to the same authority and pursue the common goals. Under such circumstances, *fully cooperative behavior*, such as unconditionally forwarding packets for each other, can be assumed. We refer to such ad hoc networks as *cooperative ad hoc networks*.

However, before ad hoc networks can be successfully deployed, security concerns must be resolved first [1]–[6]. In this paper, we study a class of powerful attacks called injecting traffic attacks. Specifically, attackers inject an overwhelming amount of traffic into the network in attempt to consume valuable network resources, and consequently degrade the network performance, such as causing network congestion and reducing network lifetime. Since in cooperative ad hoc networks, nodes will usually unconditionally forward packets for other nodes, such networks are extremely vulnerable to injecting traffic attacks, especially those launched by inside attackers who have gained access to the network.

Roughly speaking, there are two types of injecting traffic attacks that can be launched in cooperative ad hoc networks: *query-flooding attack* and *injecting data packet attack* (IDPA). Due to mobility, nodes in ad hoc networks may need to frequently perform route updates which may require broadcasting query messages. Attackers can then initiate query messages with a very high frequency to consume valuable network resources, which is called *query-flooding attack*. Attackers can also inject an overwhelming amount of garbage data packets into the network to request other nodes to forward. When other nodes process and forward these packets, their spent resource will be wasted.

To defend against query-flooding attacks, one possible way is to limit the amount of queries that can be initiated by each node in the network. Although this may degrade the network performance in certain degree, such methods can effectively limit the damage that can be caused by query-flooding attacks. However, if nodes in the network cannot know other nodes' data packet injection statistics, such as packet injection rate, then it becomes extremely hard (or impossible) to detect whether some nodes are launching IDPA. Fortunately, in cooperative ad hoc networks, since nodes belong the same authority and pursue the common goals, it is generally true that they can know each other's data packet injection statistics.

In this paper we mainly focus on protecting cooperative ad hoc networks against IDPA, especially those launched by inside attackers. We propose a set of mechanisms which can effectively detect IDPA, even when attackers can use some advanced transmission techniques such as directional antennas in attempt to avoid being detected. The probability that attackers can successfully launch IDPA without being detected is studied, which shows that from the attackers' point of view the best strategy is to conform to their packets injection rate. Meanwhile, the query-flooding attacks are also studied and the tradeoff between limiting query rate and system performance is investigated.

The rest of the paper is organized as follows. Section II describes the system model. Section III proposes a set of mechanisms to defend against injecting traffic attacks. The theoretical performance analysis of the proposed mechanisms are presented in Section IV. Simulation results are presented in Section V. Finally, Section VI concludes this paper.

II. System Model

Nodes in cooperative ad hoc networks can be classified into two types: *good* and *malicious*, in which good nodes will unconditionally help those nodes that have not been detected as malicious, while attackers' objective is to maximize the damage they can cause to the system. Each node is equipped with a battery with limited power supply, communicates with other nodes through wireless connections, and can move freely inside a certain area. Good nodes use omnidirectional transmission techniques, while attackers are also allowed to

This work was supported in part by the Army Research Office under URI Award No. DAAD19-01-1-0494

use directional transmission techniques, such as directional antennas [7], to increase their attacking capabilities.

In the current system model, data packets are generated by certain nodes and delivered to certain destinations with each packet having a specific delay constraint. We call a source-destination (SD) pair to be *legitimate* if this pair is required by the common system goals. For each legitimate SD pair (s, d) in the network, the number of packets that can be injected by this pair into the network until time t is bounded by $f_{s,d}(t)$. Since nodes belong to the same authority and pursue the common goals, we can assume that every node knows all legitimate SD pairs in the network as well as the associated upper-bounds of the packet injection rates. We assume that all nodes in the network are legitimate, no matter whether they are good or malicious. We assume that each node has a public/private key pair, and a node can know or authenticate other nodes' public keys, but no node will disclose its private key to others unless it has been compromised. To keep the confidentiality and integrity of the transmitted content, we assume that each packet will be encrypted and signed by its source when necessary. Without loss of generality, we assume all data packets have equal size.

III. Defense Mechanisms

A. Route Discovery and Packet Delivery

First, the following security enhancements are incorporated into the baseline DSR [8] to handle possible attacks, such as query flooding attacks.

When a source s initiates a route discovery to the destination d, the following format is used for the route request:

$$\{s, d, id_s(s,d), t_s(s,d), seq_s(s,d), BL_s, sig\},$$

where $id_s(s,d)$ is a unique ID specified by s for this request, $t_s(s,d)$ is the time that s issued this request, BL_s is the subset of s's blacklist that has not been broadcasted by s before, $seq_s(s,d)$ is the sequence number associated to the last data packet that s has sent to d, and sig is the signature generated by s based on message $\{s, d, id_s(s,d), t_s(s,d), seq_s(s,d), BL_s\}$. After broadcasting this request, s should also increase $id_s(s,d)$ by 1.

After a good node x has received a route request originating from s and targeting on d, x first checks the following conditions: 1) the SD pair (s,d) is legitimate; 2) all signatures are valid; 3) $id_x(s,d) < id_s(s,d)$ and $t_x(s,d) < t_s(s,d)$, where $id_x(s,d)$ is the maximum request sequence number corresponding to the pair (s,d) that x has seen before, and $t_x(s,d)$ is the latest time associated to the route requests issued by pair (s,d) that x has seen before; 4) no nodes appended to the route request packet have been marked as malicious by x; 5) less than L_{maxhop} intermediate nodes have been appended to the request packet, where L_{maxhop} is a system-level parameter indicating the maximum number of hops that any route is allowed to have in the network; 6) x has not forwarded any request for pair (s,d) in last T_x^{min} interval, where T_x^{min} is the minimum query forwarding interval specified by x to indicate that x will forward at most 1 route request for any legitimate pair in any T_x^{min} interval.

If all the conditions from 1 to 4 are satisfied, we call such a request as a *valid* request, in this situation x will update its record $BL_x(s)$ using the received information BL_s where $BL_x(s)$ is the subset of s's blacklist known by x, assign the value of $id_s(s,d)$ to $id_x(s,d)$, assign the value of $t_s(s,d)$ to $t_x(s,d)$, and assign the value of $seq_s(s,d)$ to $seq_x(s,d)$. If all of the six conditions can be satisfied and x is not the destination, x will also append its own address to the request packet, sign and rebroadcast the new request. If the request is not valid, x will simply discard this request.

Once a source has decided to send a packet to a certain destination using a certain route, a data packet delivery transaction should be started. In this paper, the data packet delivery works as follows. Suppose that node s is to send a packet with payload msg and sequence number $seq_s(s,d)$ to destination d through the route R. s first generates two signatures sig_h and sig_b, with sig_h being generated based on message $\{R, seq_s(s,d)\}$ and sig_b being generated based on message $\{R, seq_s(s,d), MD(msg)\}$ where $MD()$ is a digest function such as SHA-1 [9]. The final format of the packet to be sent is as follows:

$$\{R, seq_s(s,d), sig_h, msg, sig_b\}.$$

We refer to $\{R, seq_s(s,d), sig_h\}$ as the *packet header*, and refer to $\{msg, sig_b\}$ as the *packet body*. By also using the signature sig_h, lots of energy can be saved when performing traffic monitoring. Next, s will transmit this packet to the next node on route R (e.g., x), increase $seq_s(s,d)$ by 1, and wait for a receipt to be returned by node x.

When a node (e.g., x) detects that a certain packet is to be transmitted by another node in its neighborhood, x first decodes and checks the packet header. Assume $\{R, seq_s(s,d), sig_h\}$ is the header of the transmitted packet. x needs to continue receiving and decoding the body of the packet only if all of the following conditions are satisfied: 1) x is on the route R; 2) no nodes on route R has been marked as malicious by x; 3) $seq_s(s,d) > seq_x(s,d)$, where $seq_x(s,d)$ is the sequence number of the last packet with the source being s and the destination being d that x has seen; 4) the signature sig_h is valid; 5) route R has no more than L_{maxhop} hops, where L_{maxhop} is a system-level parameter indicating the maximum number of hops that any route is allowed to traverse in the network. After x has decided to forward the packet and has successfully received and verified the whole data packet, x will forward the packet to the next node.

B. Traffic Monitoring Mechanisms

In this paper, to detect possible injecting traffic attacks, each good node will keep monitoring its neighbors' transmission activities using the proposed *header watcher mechanism*. Specifically, when a good node x detects that a neighbor is transmitting a data packet, no matter whether x is the target of this transmission or not, x will try to receive and decode the transmitted packet header (e.g., $\{R, seq_s(s,d), sig_h\}$). If the

signature of the packet header is valid, x will put the packet header in the set $HL_x(s,d)$, which will be used later to detect whether s has launched injecting traffic attacks.

If all packet headers received by a good node x are recorded, with the increase of x's staying time in the network, more and more storage will be consumed. In this paper, for each legitimate SD pair (s,d) that x knows, only those packet headers received after the last valid route request issued by (s,d) need to be recorded by x. Since the interval between two consecutive route discoveries is usually not long, the storage requirement will become very small.

C. Attacker Detection

Now we consider the detection of injecting traffic attacks. For each set of packet headers $HL_x(s,d)$ in x's records, x will mark s as malicious if any of the following situations happens:

1) The set $HL_x(s,d)$ is not empty and the SD pair (s,d) is illegitimate.
2) For any header $\{R, seq_s(s,d), sig_h\}$ in $HL_x(s,d)$, R has more than L_{maxhop} hops.
3) x detects that in $HL_x(s,d)$ there are two valid packet headers $\{R, seq_s(s,d), sig_h\}$ and $\{R', seq'_s(s,d), sig'_h\}$ with $seq_x(s,d) = seq'_x(s,d)$ while $R \neq R'$,
4) x detects that there exists a sequence number $seq_s(s,d)$ in $HL_x(s,d)$ with $seq_s(s,d) > f_{s,d}(t)$.
5) Let $\{s, d, id_s(s,d), t_s(s,d), seq_s(s,d), sig\}$ be a valid route request received by x which is issued by s. There is a packet header $\{R, seq'_s(s,d), sig_h\}$ in $HL_x(s,d)$ which is received by x at time $t \leq t_s(s,d)$ with $seq_s(s,d) < seq'_s(s,d)$.
6) x has received a route request from an illegitimate SD pair (s,d).

In all these situations, once a good node x has detected that s has launched injecting traffic attacks, x will also notify other nodes in the network by broadcasting an ALERT message which consists of necessary evidence such as the corresponding packet headers. When other good nodes have received the ALERT message, after verification, they will also mark s as malicious.

IV. Performance Analysis

According to the secure route discovery procedure described in Section III-A, a good node x will only forward at most 1 route request in any time interval T_x^{min} for any legitimate SD pair, and will not forward route requests for any illegitimate SD pairs, therefore the total damage that can be caused by attackers launching query flooding attacks is bounded.

Next we analyze the effects of IDPA. Assume that node s is malicious and tries to launch IDPA with d being the destination of the packets injected by s. To avoid being detected immediately, the SD pair (s,d) must be legitimate and d must be malicious too; otherwise, s can be easily detected by d as malicious. There are three possible ways to launch IDPA: simple IDPA, long-route IDPA and multiple-route IDPA.

We first consider *simple IDPA*, where an attacker arbitrarily pick a route to inject a huge amount of packets into the network through this route. According to Section III-A, in order for good nodes to forward packets for s, s has to increase the sequence number $seq_s(s,d)$ by 1 after each packet delivery. Unless all nodes on the selected route are malicious, which makes no sense from the attackers' point of view, the good nodes on route R can easily detect that s is launching IDPA by comparing the received packets' sequence number with $f_{s,d}(t)$. That is, when launching simple IDPA, the attackers can be immediately detected and can cause no damage.

Next we consider *long-route IDPA*, where an attacker pick a very long route to inject data packets into the network with the injection rate conforming to the legitimate rate. If s launches long-route IDPA, since much more good nodes will be involved, s can cause similar damage as launching simple IDPA. However, as described in Section III-A, the maximum allowable number of hops per route is bounded by L_{maxhop}, and good nodes will drop all packets with the associated number of hops more than L_{maxhop}, therefore the damage is upper-bounded by $L_{maxhop} f_{s,d}(t)$ by time t.

Finally we consider the *multiple-route IDPA*, that is, the attacker picks multiple routes to simultaneously inject packets via these routes. To avoid being detected immediately, the packet injection rate to each route must conform to $f_{s,d}(t)$, and the selected routes must be node-disjoint, that is, no selected routes should share any common good node; otherwise, if a good node x lies in more than one route from s to d, it can easily detect whether s and d have launched multiple-route IDPA. Meanwhile, the packets passing through the same route should have different sequence numbers in order for good nodes on the route to forward them. Based on whether s allows packets in different routes to share the same sequence numbers and what transmission techniques s will use, there are three cases:

- Case 1: s dose not allow packets on different routes to share the same sequence numbers. Since $seq_s(s,d) \leq f_{s,d}(t)$ is required to let s avoid being detected immediately, in this case s has no extra gain compared with launching simple IDPA.
- Case 2: s allows packets on different routes to share the same sequence numbers, and transmits packets omnidirectionally. Since s's neighbors will keep monitoring s's packets transmission, they can easily detect that some packets sent by s through different routes use the same sequence number, which indicates that s is launching IDPA. Therefore if s can only transmit packets omnidirectionally, s should not launch multiple-route IDPA.
- Case 3: s allows packets on different routes to use the same sequence numbers, and can transmit packets using directional transmission techniques. Since now s's neighbors cannot receive s's transmission not targeting on them, they have little chance to directly detect that s is launching IDPA. However, since good nodes in the network use omnidirectional transmission techniques, the

probability that s can successfully launch multiple-route IDPA without being detected still approaches to 0, as to be shown next.

Next we derive the upper-bounds for the probability that s is able to successfully pick n node-disjoint routes to inject data packets without being detected immediately, as illustrated in Case 3. We consider the most general situation that the destination d does not know the exact locations of those nodes within its transmission range. Suppose that N good nodes are randomly deployed inside a large area of S. Suppose that all of these N nodes use omnidirectional transmission techniques and r is their common maximum transmission distance. Suppose that the SD pair (s, d) collude to launch IDPA with s using directional transmission technique and s and d not knowing the exact location of the nodes inside d's receiving range (which is r). If the defending mechanisms described in Section III are used by good nodes, then we can show that the probability $P(n, r)$ that the two attackers can successfully pick n node-disjoint routes to launch multiple-route IDPA without being detected immediately is upper-bounded by

$$\left(\frac{3\sqrt{3}}{4\pi}\right)^{\binom{n}{2}} \sum_{k=n}^{N} \binom{N}{k} \left(\frac{\pi r^2}{S}\right)^k \left(1 - \frac{\pi r^2}{S}\right)^{N-k} \left(n\left(\frac{3\sqrt{3}}{4\pi}\right)^{\binom{n-1}{2}}\right)^{k-n}. \tag{1}$$

Further, we can also show that the probability that two colluding attackers s and d can successfully pick 6 or more node-disjoint routes to launch multiple-route IDPA without being detected immediately is 0.

We have also evaluated through experiments the upper-bounds of the success ratio for two colluding attackers s and d to launch multiple-route IDPA with s using directional transmission technique. Given a rectangular area of $20r \times 20r$, we put d in the center of the area. At each round of experiment, we randomly deploy $400r^2\rho$ nodes inside the area and then randomly pick n nodes inside d's receiving range, where ρ is referred to as the node density. For each configuration of route number n and node density ρ, 10^7 experiments have been conducted.

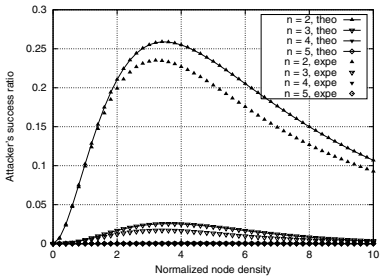


Fig. 1. Upper bounds of attackers' success probability

Both experimental and theoretical upper-bounds are plotted in Fig. 1, where "theo" denotes the theoretical upper-bounds obtained using (1), "expe" denotes the experimental upper-bounds obtained through experiments described above, and "n" denotes the number of node-disjoint routes to be picked by the malicious SD pair (s, d). In Fig. 1, the normalized node density is defined as the average number of nodes inside an area of πr^2. Since both the theoretical and experimental upper-bounds corresponding to $n = 4$ and $n = 5$ are almost equal to 0 across all illustrated node densities (e.g., for $n = 4$, all values are less than 2×10^{-3}), the four curves associated to $n = 4, 5$ have almost overlapped into one single curve, which is the lowest curve illustrated in Fig. 1. For $n = 2, 3$, we can see that the success ratio increases first with the increase of node density until it arrives at a peak, then decreases with the further increase of node density, which is consistent with (1).

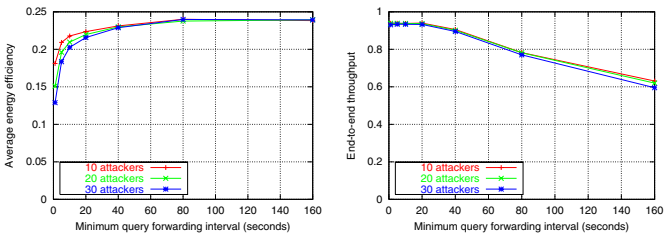
The above upper bounds are evaluated based on a fixed topology. However, due to node mobility, s needs to frequently update routes. Then after several route updates, the probability that s still has not been detected as malicious will be very small. For example, assume that each route update is independent, after 5 times of route updates, even for $n = 2$, the probability that s has not been detected as malicious is less than 0.06%. That is, attackers has negligible chance to flee. In summary, when the malicious SD pair (s, d) tries to launch IDPA, to avoid being detected and to maximize the damage, the optimal strategy is to use only one route to inject data packets by conforming to both the maximum hop number L_{maxhop} and the legitimate rate $f_{s,d}(t)$, which is equivalent to say that the optimal strategy is not to launch IDPA.

V. Simulation Studies

In our simulation, nodes are randomly deployed inside a rectangular area of $1000m \times 1000m$, and each node moves according to the *random waypoint* model [8]. The physical layer assumes that two nodes can directly communicate with each other successfully only if they are in each other's transmission range. The MAC layer protocol simulates the IEEE 802.11 DCF with a four-way handshaking mechanism [10]. The transmission range is fixed to be 250m.

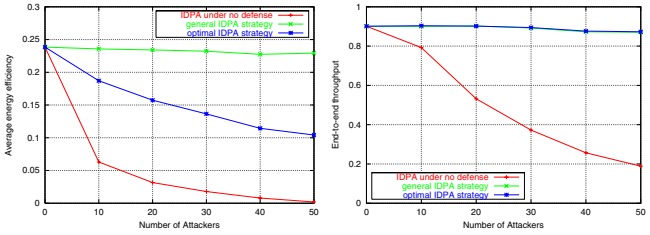
In the simulations, the number of good nodes is 100 and the number of inside attackers varies from 0 to 50. For each round of simulation, 50 good nodes are selected as the packet generators, and each will randomly pick a good node to send packets. For each attacker who launches IDPA, it will also randomly pick another attacker as the destination to inject data packets. For each SD pair, packets are generated according to a traffic rate of one packet per second, which is known by all nodes. For attackers who launch injecting traffic attacks, they will increase the average packet injection rate by 10 times. All data packets have the same size.

In our simulations, the result are averaged over 50 rounds of simulations. For each round, the simulation time is set to be 5000 seconds. When we calculate the energy efficiency, only transmission energy consumption has been considered. One reason is that transmission energy consumption plays a major role in overall energy consumption, and another reason is that receiving energy consumption may vary dramatically over different communication systems due to their different implementations. However, both data and route request packets have been considered. We assume that the transmission energy needed per data packet is normalized to be 1.



(a) Energy efficiency (b) End-to-end throughput

Fig. 2. Limiting route request rate vs. system performance



(a) Energy efficiency (b) End-to-end throughput

Fig. 3. Effects of IDPA under different configurations

We first investigate the tradeoff between limiting the route request rate and system performance. Although the performance also depends on other factors such as the mobility pattern, the number of nodes in the network, the average number of hops per route, etc., to better illustrate the tradeoff between limiting the route request rate and system performance, the other parameters are set to be fixed. However, similar results can also be obtained by changing these parameters.

Fig. 2 illustrates the tradeoff between limiting the route request rate and network performance. In this set of simulations, all attackers will only inject route request packets and will not inject any data packets. We assume that all good nodes have the same minimum route request forwarding interval denoted by T^{min}, but all attackers will set their route request rate to be 1 per second. From Fig. 2(a) we can see that with the increase of T^{min} from 1 to 80 seconds, the energy efficiency of good nodes also increases, and keeps almost unchanged from 80 to 160 seconds. The reason is that when T^{min} is small, attackers can waste good nodes' energy through injecting a lot of route request packets to request others to forward. Fig. 2(b) shows that with the increase of T^{min} from 1 second to 20 seconds, the end-to-end throughput of good nodes keeps almost unchanged, while with the increase of T_{min} from 80 seconds to 160 seconds, the end-to-end throughput of good nodes drops almost linearly. These results also motivate us to pick T^{min} to be 40 seconds in the following simulations.

Fig. 3 shows the simulation results under various types of IDPA. In Fig. 3, "IDPA under no defense" denotes the case that attackers launched simple IDPA and the underlying system has not launched any defending mechanism; "general IDPA strategy" denotes the case that attackers launch IDPA but the mechanisms described in Section III have been launched, where both multiple-route IDPA and long-route IDPA have been simulated; "optimal IDPA strategy" denotes the case that attackers will use only one route to inject data packets which conforms both to the maximum hop number $L_{maxhop} = 10$ and to the legitimate maximum packet injection rate and the mechanisms described in Section III have been launched.

From Fig. 3(a) we can see that when there is no defending mechanisms for IDPA, even simple IDPA can dramatically degrade the energy efficiency of good nodes. When the defending mechanisms described in Section III are employed, from attackers' point of view, launching IDPA has no any gain in decreasing the energy efficiency of good nodes. However, if attackers apply the optimal IDPA strategy, they can still degrade the energy efficiency of good nodes. From Fig. 3(b) we can see that without employing necessary defending mechanisms, with the increase of the number of attackers, even simple IDPA can dramatically degrade the end-to-end throughput of good nodes due to the congestion they caused. When the defending mechanisms described in Section III are employed, launching IDPA has almost no effects on the performance of good nodes' end-to-end throughput.

VI. Conclusion

In this paper we have studied the possible injecting traffic attacks that can be launched in cooperative ad hoc networks and proposed a set of mechanisms to defend against such attacks. Both query flooding attacks and injecting general data packets attacks have been investigated. Furthermore, for injecting general data packets attacks, the situations that attackers may use some advanced transmission techniques such as directional antennas to avoid being detected have also been studied. Our theoretical analysis has shown that when the proposed mechanisms are used, the best strategy for attackers is not to launch injecting traffic attacks. Extensive simulation studies have also agreed with our theoretical analysis.

References

[1] L. Zhou and Z. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. Nov./Dec., 1999.

[2] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Mobicom 2000*, August 2000, pp. 255–265.

[3] J. P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," in *MobiHOC 2001*, May 2001.

[4] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," in *MobiCom 2002*, Atlanta, GA, USA, Sep. 2002.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols," in *WiSe*, San Diego, CA, USA, Sep. 2003.

[6] I. Aad, J. P. Hubaux, and E. Knightly, "Denial of service resilience in ad hoc networks," in *ACM MobiCom*, Philadelphia, PA, September 2004.

[7] J. D. Kraus and R. J. Marhefka, *Antennas: for All Applications*, McGraw-Hill, New York, 3rd edition, 2002.

[8] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing," In *Moible Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[9] "Secure Hadh Standard," Federal Information Processing Standards Publication 180-1, 1995.

[10] IEEE Computer Society LAN MAN Standards Committee, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1007," The Institue of Electrical and Electrics Engineers.