# DYNAMICS OF LEARNING ALGORITHMS FOR THE ON-DEMAND SECURE BYZANTINE ROUTING PROTOCOL

Baruch Awerbuch, Reza Curtmola, David Holmer, Herb Rubens[1] and Robert G. Cole[2]

[1] Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA
[2] The JHU Applied Physics Laboratory, Laurel, MD, USA

**Abstract.** We investigate the performance of of several protocol enhancements to the On-Demand Secure Byzantine Routing (ODSBR) [3] protocol in the presence of various Byzantine Attack models. These enhancements include a) Nodal Weighting (in addition to Link Weighting) in the reputation database, b) a network layer retransmission capability and c) modifications to the packet flow rates. These enhancements are meant to improve the learning rate of the protocol in the presence of various Byzantine Attack models. The attack models investigated include previously investigated models [4] and a new and effective attack model, termed the *MAC-Level Attack*. We investigate the dynamics of the ODSBR protocol and its enhancements through analytic models and extensive simulation studies. We find that the protocol enhancements improve the learning times of the ODSBR protocol. The Nodal Weighting enhancement specifically helps in the presence of the various colluding Byzantine Attack models investigated. However, the time to develop a relatively complete reputation database in the MANET context is found to be relatively large.

**Keywords:** {MANET, Security, Byzantine Attacks}

## 1. INTRODUCTION

Developing ubiquitous and secure computer networks is a major challenge in today's world. The Internet has demonstrated itself to be vulnerable to numerous and evolving security intrusions. The growing reliance on wireless networks exacerbates the problem by offering easy access to the communications media. Emergency services, Homeland Security, and the U.S. Military are planning on the extensive reliance on Mobile Ad Hoc Networks (MANETS) for key communications capabilities. It is imperative that the issues of computer network security be addressed within the core networking protocols building the foundations for MANETS and wired networks.

Much effort into hardening networking protocols concentrates on protection against *Outsider Attacks*. This body of work relies on cryptographic mechanisms to ensure the integrity, authenticity and confidentiality of data. Examples of work securing routing protocols with these techniques are [13], [10] and [19]. Less work has addressed *Insider (Byzantine) Attacks*. It is assumed *a priori* that the Byzantine adversary has gained access to one or many of the network nodes and therefore has access to the cryptographic keys associated with the compromised nodes. Several situations where insider attacks are likely include a) tactical, battlefield situations where the opponent has overrun networking assets, b) a computer worm has infected networking components, and c) non-malicious code failure cause components on the network to act in improper ways which disrupt the overall network performance.

In [3], the On-Demand Secure Byzantine Routing (ODSBR) routing protocol was proposed for MANETS. ODSBR is secure against outsider attacks due to presence of cryptographic mechanisms. Notably, ODSBR is also secure in a well defined sense against Byzantine Attacks. It was shown in [3] that ODSBR places a strict bound on the level of damage that a Byzantine node can inflict on the network, independent of the dynamic behavior of the Byzantine node. In [4], the stationary, time averaged performance of ODSBR was evaluated through the development of an extensive simulation model. The simulation model captured the protocol actions of ODSBR and included several Byzantine node attack models. These included both independent and colluding Byzantine nodes.

In this paper, we extend the work in [4] by a) proposing several protocol enhancements to the ODSBR protocol in the context of a new attack and b) analyze their performance impact on the time dependent convergence dynamics of the protocol. The specific protocol enhancements investigated are packet flow rate adjustments, end-to-end retransmission at the network layer and the addition of a nodal weighting component to the reputation database. These enhancements are shown to improve the efficiency of the ODSBR protocol in avoiding Byzantine attackers. Finally, we investigate the voracity of a new MAC-Level attacker in disrupting the network performance. In this context we demonstrate the benefit of incorporating the nodal weighting enhancement into the ODSBR protocol. We investigate these protocol modifications and the impact of the new attack through analytic modeling and extensive simulation studies. We analyze the protocol enhancements with respect to their impact on the average time it takes the protocol to learn the presence of network adversaries and find non-adversarial paths through the network. In this sense, we investigate the temporal dynamics of the ODSBR protocol.

The rest of this paper is organized as follows: The next section reviews previous, related work. Section 3 overviews the ODSBR protocol. Section 4 lists the various attack models addressed in this paper. Section 5 presents an analytic model of the ODSBR dynamics. Section 6 presents our simulation studies of the protocol enhancements to the ODSBR protocol. Section 7 contains conclusions and proposed future work.

## 2. PREVIOUS WORK

Notable work investigating the development of protocols which are resilient to Byzantine Attacks include [11] and [17]. [11] provided an analysis of the Byzantine Generals Problem, i.e. reaching consensus in the presence of malicious participants. [17] studied the general problem of hardening the Networking Layer of a data network against Byzantine attackers. The analysis of two approaches was presented, one based upon a flooding algorithm for path discovery and one based upon a link state method.

Work on securing MANET routing protocols against Byzantine attacks falls into several categories, i.e., Passive Neighbor Monitoring Methods, Active Monitoring Methods and Active Monitoring with Fault Isolation Methods. The work in [9], [7] and [12] investigated passive methods to monitor the behavior of neighboring nodes in order to detect faulty behavior. In these works, if a neighbor is deemed to be misbehaving, the monitoring node suggests or carries out a path reroute around the faulty neighbor. These methods require that the networks rely on omni-directional antennas and nodes that transmit at a single rate, i.e., no multi-rate systems can be used.

[1] investigated the use of active monitoring capabilities, generally in the form of end-to-end monitoring, in their investigation of novel attack scenarios against TCP flows. In the event the path is determined to be faulty, then alternative paths are chosen. The work did not address the issue of identifying the faulty component or avoiding it in the network reroute. Instead, it relied on finding diverse paths when choosing alternate routes in the network. [15] and [16] proposed the use of diverse, multi-path routing as a means to secure data transmissions against malicious nodes within a MANET. Here data packets are segmented, redundancy is added, and transmitted over a set of disjoint paths. The destination then reassembles the original data packets.

Several studies have investigated both active monitoring and fault isolation systems. These are often referred to as ‘Reputation-Based’ systems, because the nodes maintain a picture of the reliability of each component comprising the network. Notable works in this area are [2] and [3]. In [2], a Byzantine resilient protocol was proposed for a Link State protocol in a wired environment. Their scheme relied on end-to-end monitoring and fault isolation to identify faulty links. They analyzed a buffer allocation scheme in the network to ensure data transmission in the presence of congestion. They discussed forms of information sharing between good nodes, as we discuss in later sections of this report. In [3], the ODSBR protocol was proposed for the network layer in a MANET.

## 3. THE ODSBR PROTOCOL

ODSBR [3, 4] is a point-to-point on-demand secure routing protocol for ad hoc wireless networks, designed to be resilient against a wide range of external and Byzantine attacks. It is based on the observation that no matter what attack and how it is executed, the only threat an adversary can pose is to disrupt packet delivery. Data packets and control packets are coupled together, so that adversaries do not go undetected if they start dropping packets. The protocol assumes that while all the network nodes can be authenticated, only the source and destination can be fully trusted. At the highest level, ODSBR operates using three modules: the *Route Discovery Module*, the *Path Monitoring Module*, and the *Component Weighting Module.*

The *Route Discovery Module* returns the least weight route from the source to the destination based upon a reliability metric that captures past history. The metric is represented by a Link Weighting Table that contains weights of links and is maintained by the source node. The Route Discovery Module relies on an on-demand, double flooding mechanism which is based on a combination of the source digitally signing the flood and per node flood verification. Faulty links have a high weight and are avoided in this process.

The *Path Monitoring Module* monitors the quality of the source-routed path over which data packets flow, based on end-to-end acknowledgments for each data packet sent across the MANET. If the packet loss rate exceeds a prescribed *loss threshold* on a given route, the Path Monitoring Module enters a fault isolation state, in which the source uses an adaptive probing technique to locate faulty links on the path to the destination. The source requires secure acknowledgments from intermediate nodes along the route. Due to the structure of the probing scheme, a fault will be attributed to one of the links adjacent to the adversary. The source then updates the Link Weighting Table and initiates a new route discovery. The loss threshold is tracked by maintaining a *sliding window* which holds the recent history of loss events.

The *Component Weighting Module* maintains the node's current view of the reliability of each link in the network. The measure of reliability is developed based upon input from the Path Monitoring Module. A link has its weight increased if it is found faulty. The weight of a faulty link is decreased based upon the source's view of successful data packets delivered to the destination over that faulty link.

When combined together, as long as a fault free path from the source to the destination exists, these three modules bound the number of losses caused by adversaries, even when a majority of the nodes are colluding Byzantine adversaries.

## 4. ATTACK SCENARIOS

The ODSBR protocol is resilient to the following Byzantine and non-Byzantine attack scenarios [3]:

*False Route Attack* where an adversary generates a false route. In ODSBR, the route is built up while the route_req packets are flooded through the network. Each node's contribution to the path is appended to the route and an aggregate hash is added which protects against modifications to the list by adversaries.

*Incrimination Attack* where an adversary tries to incriminating other nodes by tampering with the end-to-end acknowledgments. Because of the aggregate integrity mechanism [6] used in ODSBR, a given node on the path is not able to modify an upstream acknowledgment to incriminate the downstream node.

*Black Hole Attack* where the adversarial node correctly participates in the route discovery protocol, but then behaves errantly during data transport. The ODSBR Path Monitoring methods will discover this behavior and isolate the offending link.

*Flood Rush and Worm Hole Attack* where the adversarial nodes act to encourage routes to be setup through them and then behave errantly during data transport. Methods include expediting route discovery packets to speed delivery of their path information to the source (i.e., Flood Rush) or building tunnels between colluding nodes to imply that shorter paths exist through them to the source (i.e., Worm Hole and Super Worm Hole attacks). The ODSBR protocol will learn through Path Monitoring and Link Weighting to avoid these spurious links. Further, because the ODSBR source node will accept route_resp packets with path weights smaller than any prior route_resp packet, Flood Rush attacks are ineffective.

*Adaptive Packet Dropping* where a Black Hole attacker could adapt its packet dropping behavior as an attempt to defeat the Path Monitoring module. It is shown in [3] that regardless of the dynamics of the packet dropping algorithm employed by the attacker, the ODSBR protocol bounds the total loss rate to a reasonably small value proportional to the loss threshold employed.

*MAC-Level Attack* where a pair of protocol passive adversaries with radio repeaters create the perception of numerous false links. Each device monitors its local radio channel and transmits signals down a tunnel to its remote mate. The remote mate then transmits the signal it receives through the tunnel out onto its local radio channel. The effect is to make all nodes within radio range of one repeater think they are a single hop from all nodes within radio range of the remote mate. This is extremely effective in pulling in routes and, when implemented in conjunction with a Black Hole Attack, can cause severe havoc on the network performance. This is a new attack, first analyzed and simulated in this paper. The ODSBR Path Monitoring methods will discover this behavior and isolate the adversarial links.

The resilience of the ODSBR protocol to these various attack scenarios was discussed extensively in [3]. In [4], simulation studies of the ODSBR protocol under various simulated attack scenarios and mobility conditions were presented. It was shown that the performance of the ODSBR protocol is robust against the attack scenarios discussed above (not including the last attack model).

## 5. ODSBR DYNAMICS

In this section we develop a qualitative model of the dynamics of the ODSBR protocol. The model results are validated against our simulation studies. In [5], we present a more thorough discussion of our modeling and its assumptions.

When a source node in the ODSBR wishes to establish a data flow in the network, it performs a relatively quick route discovery. Once a route is found, the source node begins data transmission to the destination, while performing end-to-end monitoring of the flow. If the route contains one or more adversaries, then the source will detect and isolate a faulty link and perform a reroute. It may go through several reroutes prior to finding a good path through the network. Eventually, the existing path will break due to nodal mobility. Once again the source must perform a relatively quick route discovery and re-establish a data path to the destination. Prior to the route breaking due to nodal mobility, the source node learned about the integrity of some of the links and recorded this information in its Link Weighting Table. This information will reduce the probability of initially hitting a route with an adversary in future route discoveries. This behavior repeats as the protocol switches between monitoring a route and searching for a new good route when the current route breaks due to mobility. This picture assumes the relative motion of the nodes is slow compared to the response of the route discovery and monitoring functions. If this assumption does not hold, then the ODSBR's Path Monitoring and fault isolation functions will fail to have sufficient time to discover adversarial behavior along the current route.

Hence, our model of the ODSBR protocol dynamics is as follows: Imagine the MANET system stepping through a series of static topology cases. Each static case has a lifetime equal to the mean lifetime of a given path through the network. As the system steps to a new topology state, it inherits the knowledge about the integrity of the links learned from the previous topology cases. Upon entering a new topology, the source node initiates a new route request. Thus two time scales exist; a short time scale related to the time required within a static topology for the source node to find a good route and a long time scale related to the time for a given source node to build up its Link Weight Table. The dynamics of the short time scale learning are studied within the context of a static topology case. The dynamics of the long time scale learning are studied within the context of information built up by progressing through the series of static topology cases. Figure 1 contains a pictorial representation of the model. The figure shows the progression through a series of static topology states. The system jumps from state-to-state each $\tau_{topology}$ seconds. Associated with each static topology state, the figure shows a corresponding plot representing the expected behavior of the probability of adversarial drops versus time within the static topology state.

### 5.1. The Short Time Scale Model

The dynamics of the probability of packet drops in each static topology state will be described by the Markov model [8] shown in Figure 2. With probability $P_{NP}$, the protocol finds that no path exists and all packets are discarded. With probability $1 - P_{NP}$, the protocol finds itself within a static state where at least

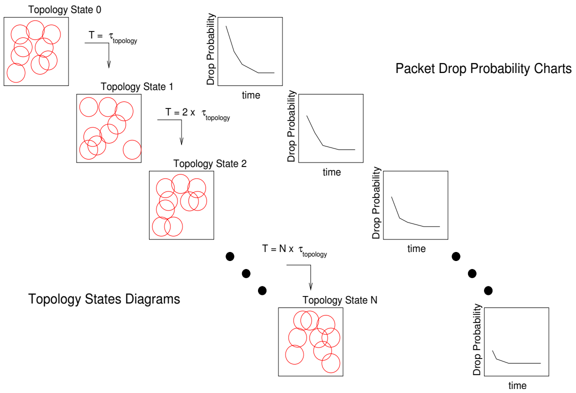

**Fig. 1**. The ODSBR dynamics as a progression through a series of static topology states.

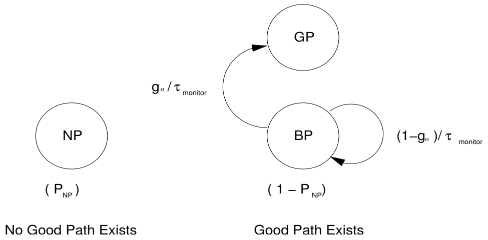

**Fig. 2**. A Markov model of the short time scale dynamics of the ODSBR protocol.

one good path exists. Here, the protocol exists in one of two states, a bad state with probability $P_{BP}(t)$ where the path contains an adversary, and a good state with probability $P_{GP}(t) = 1 - P_{BP}(t)$ where the path contains no adversaries. In the bad state, the ODSBR Path Monitoring function will spend some time to identify the presence of an adversarial node and then institute a reroute to pick a new path. It then transitions to one of the two states, i.e., the good state or the bad state. The transition probability from the bad state to the good state is equal to the product of $g_0$, the probability of randomly choosing a good path in a static state with good paths, and $\tau_{monitor}^{-1}$, the mean rate for the Path Monitoring Module to locate a faulty link in the current path and initiate a reroute.

From the state diagram in Figure 2, we obtain an expression for the probability of adversarial packet drops, $P_{PD}(t)$, as a function of time, i.e.,

$$P_{PD}(t) = P_{NP} + (1 - P_{NP})(1 - g_o)e^{-(\frac{g_o}{\tau_{monitor}})t} \tag{1}$$

This expression accounts for topology constraints and packet drops by adversarial nodes. This expression does not address buffer overflows due to congestion or lost packets due to noisy radio channels. This behavior is illustrated in Figure 3.

The ODSBR protocol relies on a loss threshold to determine a link fault. It maintains a sliding window of size $W$ packets which holds a record of the packet delivery success over the route. When the ratio of lost packets to window size exceeds the loss threshold, ODSBR declares a route fault and enters its fault isolation phase. It then isolates the offending link by observing an additional loss threshold exception. Assuming that a Byzantine node drops all data packets (as we assume in the simulation modeling below), then ODSBR must monitor a total of $2TW$ packets on a route before isolating a faulty link and rerouting. Here $T$ is the lost threshold, e.g., 0.1, and $W$ is the sliding window, e.g., 100 packets. Given a flow rate of $R$ packets per second, it then takes ODSBR protocol $(2TW/R)$ seconds to reroute away from a bad route, i.e.,

$$\tau_{monitor} = 2TW/R \tag{2}$$

## 5.2. The Long Time Scale Model

The model of the of the system's long time scale behavior is similar to the short time scale model, but it additionally incorporates a state dependent $g(t_n)$, the probability of picking a good path during a route discovery phase. Here, $t_n$ is the time at which the static topology state changes for the nth time, i.e., $t_n \approx n\tau_{topology}$. The protocol is designed such that $g(t_n)$ will increase as time progresses, since ODSBR nodes are better able to avoid bad paths based upon previous network measurements. Thus, the system will roughly evolve as illustrated in Figure 4. Here, the initial probability of packet drops within each new static topology state decreases as the ODSBR nodes learn more about their environment.

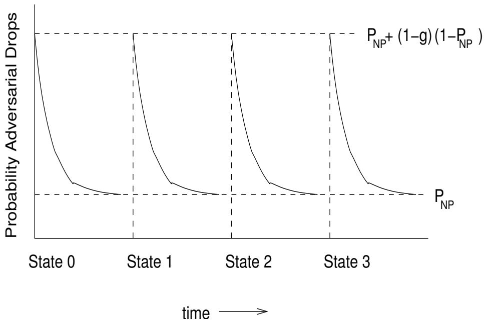

**Fig. 3**. The model predictions for the short time scale dynamics of the ODSBR protocol.

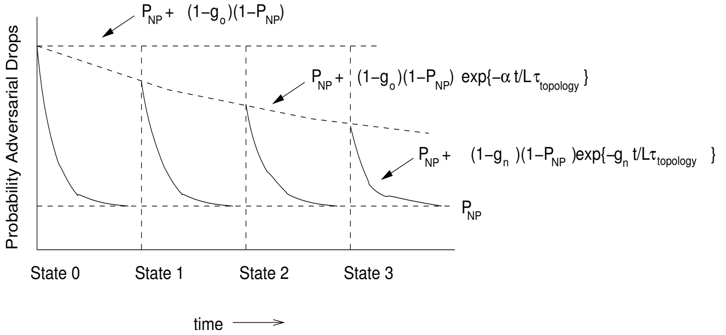

**Fig. 4**. The model predictions for the long time scale dynamics of the ODSBR protocol.

The exact form of $g(t_n)$ is not known. In general, it is a complex function of topology parameters, e.g., nodal density, $g_0$, and the extent of ODSBR's discovery of faulty links. By definition, $g(0) = g_0$ and $g(t \to \infty) = 1$. The simplest, non-trivial form that $g(t)$ may take is

$$g(t_n) \approx g_0 + (1 - g_0)\frac{l(t_n)}{L} \tag{3}$$

where $l(t_n)$ is the number of links tested by state $n$ and $L$ is the total number of links to be tested in the network. Assume that on each visit to a static topology state, an ODSBR node tests (or measures) $\alpha$ links at random. Some of these links may have been previously measured by the node. For simplicity, we model time as continuous and write

$$l(t + \delta t) \approx l(t) + (\delta t)\tau_{topology}^{-1} \times \{0 \times Pr\{link\ measured\} + \alpha \times Pr\{link\ not\ measured\}\} \tag{4}$$

where we eventually take the limit $\delta t \to 0$. The probability that a link has not already been measured is

$$Pr\{link\ not\ measured\} \approx (L - l(t))/L \tag{5}$$

Hence, we get

$$\frac{dl(t)}{dt} \approx \alpha(L - l(t))/(L\tau_{topology}) \tag{6}$$

This expression has the solution

$$l(t) = L\{1 - e^{-\alpha t/L\tau_{topology}}\} \tag{7}$$

Inserting this expression into the continuous time version of Eq.(3), we get

$$(1 - g(t)) = (1 - g_0)e^{-\alpha t/L\tau_{topology}} \tag{8}$$

The model suggests that the short and long time scale learning algorithms of the ODSBR protocol roughly converge exponentially. In reality, the long time scale learning algorithm converges slower than exponential (see [5]). The result for the long time scale convergence is a lower bound on the learning time due to the assumptions regarding $\alpha$. Table 1 lists the half life of each of the exponentially decaying terms.

## 6. SIMULATIONS STUDIES

In this section we analyze the effectiveness of our proposed ODSBR protocol enhancements in the context of our previously discussed protocol model and extensive simulation studies. In [4] a simulation model of the ODSBR protocol was developed based upon the NS2 simulation tool kit [14]. The attack models discussed in Section 4 were also implemented in the NS2 simulator. The stationary state, packet discard ratio was reported under various attack scenarios and different nodal speeds. Here we extend the work in [4] by analyzing ODSBR

**Table 1**. Half lives for the learning algorithms.

| Learning Algorithm | Half Life |
|---|---|
| Short Time Scale | $t_{1/2} = \frac{(2TW)}{gR} ln2$ |
| Long Time Scale | $t_{1/2} = \frac{Lr}{v} \frac{ln2}{\alpha}$ |

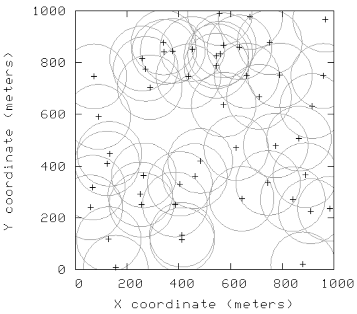

**Fig. 5**. An example topology for the MANET simulations.

protocol enhancements and the impact of the new MAC-level attack. This requires simulation modeling of the temporal behavior of the ODSBR protocol.

The prominent metric we investigate is the *Probability of Adversarial Packet Discards*, which is the ratio of the packets dropped due to the adversarial behavior of the Byzantine nodes divided by the number of packets injected into the network by the application flows over the given time interval. The ODSBR protocol does not distinguish between dropping due to adversarial nodes and other causes of packet loss, e.g., buffer overflows, channel fading, etc. However, we have chosen to focus solely on adversarial drops as a metric and track the performance of the protocol with respect to its efficiency in minimizing this metric. Our simulation runs are divided up into ten equal time intervals and the adversarial packet discard probability is reported over these ten intervals. Our base case simulation model follows [4] and is comprised of 50 good nodes and 10 adversarial nodes located in a 1000 by 1000 meter grid. The nodes are randomly placed within the grid and move within the grid according to the modified Way Point model [20]. Each result represents the average over 30 independent simulation runs. The underlying radio model is that of an 802.11 wireless network with a data rate of 2 Mbps and a radio range of 250 meters as determined by a Two Ground Wave propagation model [18]. Ten CBR traffic sources are chosen at random over the good nodes. These traffic sources are connected to 10 traffic sinks, also chosen at random over the good nodes excluding the source node. An aggregate load of 0.1Mbps was offered to the network by having each flow send 256 byte packets at approximately 4.9 packets per second. Figure 5 shows an example of a random placement of 50 nodes within a 1000 by 1000 meter grid. Each node is identified at the center of a circle of radius 125 meters, which is half their radio range[3]. Hence, if the circles around two nodes overlap, then they have radio connectivity. Else, they do not. The figure indicates the density of nodes studied within this paper.

In the next sub-section we analyze the time dependent behavior of the original ODSBR protocol through simulation. These results are discussed within the context of the short-time dynamics model developed in the previous section. We then analyze in three following sub-sections the three protocol enhancements addressed within this paper, i.e., a) increasing the packet flow rates through smaller packet sizes, b) implementing a network layer retransmission protocol, and c) implementing a nodal weighting component to the Component Weighting Table. These are analyzed in the context of our simulation models.

### 6.1. Short Time Scale Dynamics

Figure 6 shows the temporal behavior of the adversarial packet discard probability for simulation runs of 30 seconds. Each data point represents the discard probability averaged over 3 second intervals. The two plots show the short term learning of the ODSBR protocol in the presence of 10 randomly placed adversarial nodes running the Black Hole Attack. The Left Hand Side (LHS) plot runs the ODSBR protocol as defined in [3] and [4], while the Right Hand Side (RHS) plot shows the performance of the modified ODSBR protocol which incorporates a form of nodal weighting introduced in this paper and discussed in the next section. We see that there is only a slight difference between the dynamics of the Link versus the combined Link and Nodal Weighting. The four separate curves within each plot represent simulation runs with different nodal speeds, i.e., 0, 1, 5 and 10 meters per second. Due to packet buffering and finite route discovery times, there is a

[3] Plotting their full radio range on the plot is too confusing to the eye, and does not give a good sense of the nodal densities.

startup period in the beginning of the simulation runs. Once initial routes are established and packets begin to flow into the network, the Byzantine nodes running their Black Hole Attack begin to drop data packets. This results in the relative large peak in the probability of adversarial packet drops in the 3 to 12 second range. As the route monitoring function begins to fault isolate the adversarial links, initiating path reroutes, the probability of adversarial drops begins to decrease in roughly an exponential fashion as predicted by our model of the short time scale dynamics. The probability of adversarial packet drops will not approach zero due to the the finite probability that no good path exists between a pair of nodes. This is illustrated in the simulation results by observing the zero speed curves. These appear to converge to a small but finite lower value.

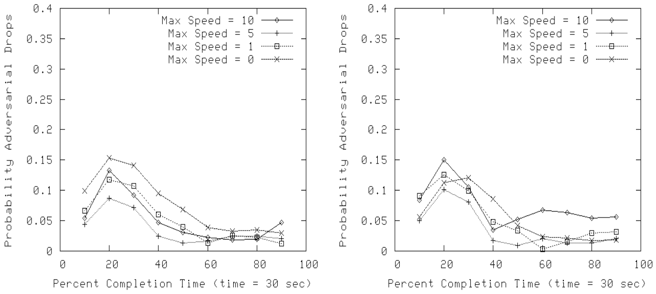

**Fig. 6**. The convergence of the search algorithms against the BH attack.

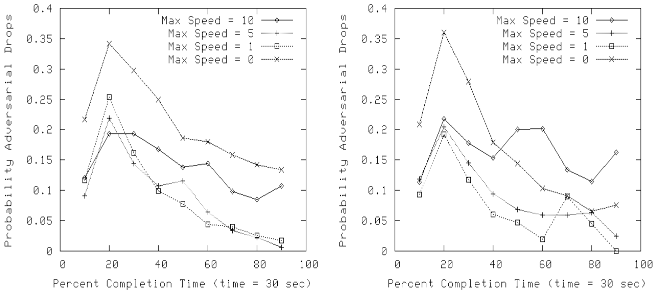

**Fig. 7**. The convergence of the search algorithms against the BHF attack.

Figure 7 shows the comparable results for adversarial nodes implementing both the Black Hole and the Flood Rush (BHF) attack. The impact of this attack is slightly worse than the BH attack, at least in the initial loss probabilities. We speculate that this may be due to ODSBR first routing across bad paths determined by the flood rushing adversaries, but then switching to a shorter path once it receives a better *route_resp* packet. Figure 8 shows comparable results for adversaries which are implementing Black Hole with Flood Rush and forming a fully interconnected set of Worm Holes (BHFSW). This is an extremely powerful attack scenario. From the simulations, we see that the ODSBR protocol is reducing the probability of adversarial dropping, however the results are not as good as for the results of the two previous attack models due to the topological complexity that the Super Worm Holes introduce. Notice that the Nodal Weighting scheme shows improved performance in the RHS plot.

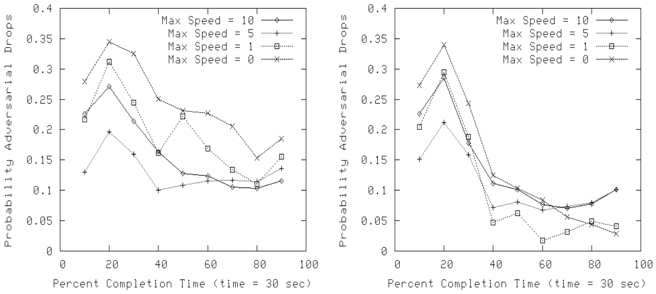

**Fig. 8**. The convergence of the search algorithms against the BHFSW attack.

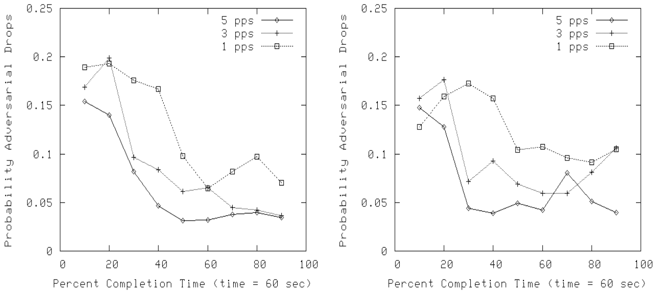

**Fig. 9**. The impact of packet flow rates on convergence times for two different mobilities, i.e., 1 mps for the LHS plot and 5 mps for the RHS plot.

Our analytic model predicts a short term half life of $(2TW/Rg)ln2 \approx (2x0.1x100/5x0.4)ln2 = 7$ seconds. The expected time within the static topology state is roughly $r/v \approx 250/(5mps) = 50$ seconds. So these simulation runs represent the short term learning behavior of the ODSBR protocol.

### 6.2. Increased Packet Flow Rate

We now investigate potential protocol enhancements to the ODSBR protocol. We first look into the possibility of reducing network packet size to achieve increased packet flow rates to improve protocol convergence. This of course comes with the negative impact of increased packet protocol overhead. Figure 9 shows the simulation results for different packet flow rates as a function of nodal mobility. The plot on the LHS of the figure gives the results for three different packet flow rates, i.e., 1 packet per second (pps), 3 pps and 5 pps for a nodal maximum speed of 1 meter per second (mps). The plot on the RHS of the figure shows the comparable results for a maximum nodal speed of 5 mps. We see that the half life for the short time scale convergence increases as the packet flow rates decrease, as we expect. The increase is not directly proportional to the packet flow rate because the route discovery times have not proportionally changed. Here, the route lifetime for the 1 mps nodal speed cases (the LHS of the figure) is roughly 250 seconds, while for the 5 mps nodal speed cases (the RHS of the figure) is roughly 50 seconds. The Route Monitoring lifetime of the ODSBR protocol is expected to be roughly 50 seconds for the slowest packet flow rate of 1 pps. Thus, we expect that the convergence of the learning protocols to be much better in the plot on the LHS than on the RHS. The convergence of the 1 pps flow rate with a maximum nodal speed of 5 mps is rather poor, as we would expect. The convergence in all cases in Figure 9 improves with increased packet flow rates.

### 6.3. Network Layer Retransmission

Another way to effectively increase the packet flow rate is to implement a network layer retransmission protocol. This was suggested in [17] as a means to improve the overall robustness of the network layer against general Byzantine attacks. We have implemented a simple retransmission protocol at the network layer in our NS2 simulation. The ODSBR protocol runs an end-to-end acknowledgment protocol with a timeout mechanism, which complements the acknowledgment protocol in the event of packet loss. The default timeout is 0.5 seconds times the number of hops remaining to the destination node. We implemented the retransmission protocol at the source node by keeping track of the number of times a given packet was transmitted to the destination. When the source node either times out or receives a lost packet indication, the source retransmits the packet in the event that it has not been transmitted in excess of $n_{retries}$ times. Thus, the source node effectively increases the packet flow rate by retransmitting each packet up to $n_{retries}$ number of times in the presence of an adversary.

Figure 10 shows the results of the network layer retransmissions on the convergence of the ODSBR protocol. The plot on the LHS presents the raw results from our simulation runs. The three curves on this plot represent the results for an $n_{retries}$ value of 0, 2 and 4, respectively. As the number of retries increases the maximum peak in the ratio of adversarial discards increases, and in fact exceeds unity. The reason for this is that what is being plotted is the ratio of the number of adversarial drops in the network divided by the of packets sent by the application. Due to the possibility that each application packet is retransmitted, this value can exceed unity. To better visualize the results in the RHS plot, we have scaled the individual curves according to the peak values (the curve for an $n_{retries} = 0$ was left unchanged). As $n_{retries}$ increases, the convergence half lives decrease, although not as dramatically as one might expect. The application flow rate in these examples is roughly 5 pps. At that rate, the threshold detection and fault isolation time is roughly 7 seconds. The default timeout for the ODSBR protocol is 0.5 seconds times the mean path length. Hence, the route is established and carries quite a few packets prior to the first packet retransmission occurring. Hence, the retransmission protocol is effective in increasing the packet flow rate only over a later portion of the path lifetime. Nonetheless, we do see a decrease in the half life of the convergence times as $n_{retries}$ increases.

In addition to decreasing the convergence time of the route monitoring times for the ODSBR protocol, the existence of a network layer retransmission protocol may further buffer (or completely mask) the effects of other Byzantine attack models. In [1], a number of novel Byzantine attacks against TCP flows were discussed. We plan to investigate these aspects of a network layer retransmission protocol in the presence of other attack models and traffic types in future work.

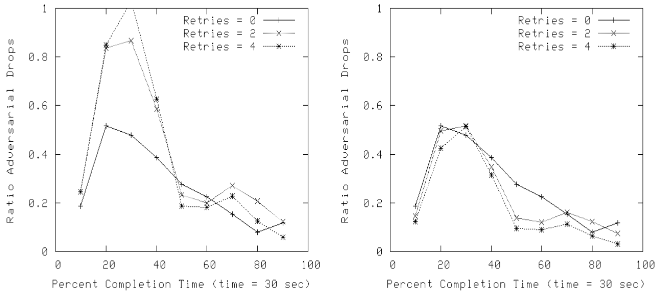

**Fig. 10**. The impact of packet retransmission on convergence times for 30 second run with and without renormalization.

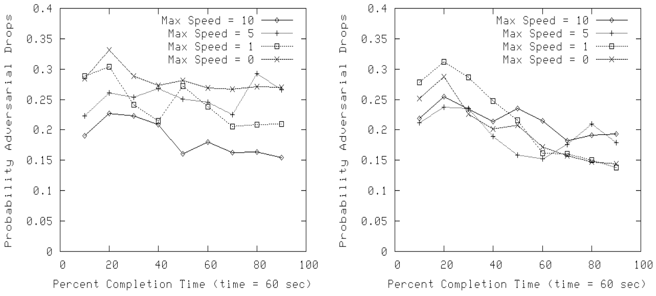

**Fig. 11**. The convergence of the search algorithms against the MAC-level attack model.

## 6.4. Nodal Weighting

In this section we discuss the impact of the addition of Nodal Weighting to the ODSBR component reputation database. We expect that this enhancement will improve the long time scale convergence of the protocol. On this time scale, learning is the result of building up knowledge of the behavior of the entire network. We were motivated to propose a Nodal Weighting component to the ODSBR reputation database while investigating the impact of the new, MAC-level attacker discussed in Section 4. This attack pulls in many routes because it makes many distantly separated nodes think they are directly connected and therefore tricks nodes into thinking there is a much shorter path (through the MAC-level tunnel) available to them. Hence, we discuss the impact of Nodal Weighting primarily in the context of this new attack model.

Figure 12 shows static test cases we first consider in order to test the performance of the learning algorithms under controlled conditions against the MAC-level attack model. Three test cases are considered. For each case, the nodes in solid black are participants in the ODSBR routing protocol and the nodes in dashed black represent the two colluding MAC-level attack nodes. The dashed black line between the two MAC-level nodes represent the tunnel set up between the two colluding nodes. The source node is on the far left of each figure and the destination node is on the far right. The nodes are assumed to have a radio range of 250 meters. The distances between the nodes are indicated in the figure. Case A has only a single virtual path through the colluding pair. Case B has $3^2$ virtual paths due to the fact that 3 good nodes are found within range of each colluding node. Case C has $5^2$ virtual paths.

We ran NS2 simulations for each topology case. The MAC-Level colluding nodes were implemented within the ∼*ns2/mac* section of the NS2 code. We implemented the MAC-Level attackers by having them corrupt tunneled data packets, which are eventually discarded by a receiving node's LLC layer. We measured the time to which the ODSBR protocol first set up a good route (around the MAC-level adversaries). Theses results are reported in Table 2. The results represent the average over 30 independent simulation runs. Also, the source runs a CBR application generating roughly 4.9 packets per second to the destination node.

We investigate two learning algorithms, i.e.,

*Ln,m,p* - where $n$ is the proportional weight given to the path hop count, $m$ is the proportional weight given to the links based upon the fault count, and $p$ is the proportional weight given to the nodes in the path based upon the number of faults summed across all of their links. The $L$ indicates that the respective weighting is increased linearly proportional to the component's fault count. The fault count is the number of times a link has been implicated by the ODSBR fault isolation function. The path weight is given by the sum of the hop, link and nodal weight for each component in the path. So $L1, 1, 0$ implies that all paths have an initial weight equal to the number of hops in the path and that the current weight of the link is equal to one times the number of faults.

*Wn,m,p* - where the $W$ indicates that the link weights are proportional to $2^{(number\ of\ faults)}$. Thus, $W1, 1, 0$ represents the original ODSBR weighting mechanism developed in [3] and reported on in [4]. The notation $W1, 1, 1$ represents a new learning algorithm we refer to as *Nodal Weighting*. In nodal weighting, an additional nodal weight is given to the path weight. For each node in the path, a nodal weight is developed by summing the link weights of all the links connecting to the node and then multiplying that sum by the proportional weighting factor, $p$.

**Table 2**. Learning Times for Static Test Cases.

| Case | Weights | Time<br>(seconds) | Search<br>Combinations |
|---:|---:|---:|---:|
| A | L1,1,0 | 75 | 1 |
| A | W1,1,0 | 30 | 1 |
| A | W1,1,1 | 15 | 1 |
| | | | |
| B | L1,1,0 | 680 | 9 |
| B | W1,1,0 | 240 | 9 |
| B | W1,1,1 | 48 | 9 |
| | | | |
| C | L1,1,0 | 1900 | 25 |
| C | W1,1,0 | 720 | 25 |
| C | W1,1,1 | 72 | 25 |

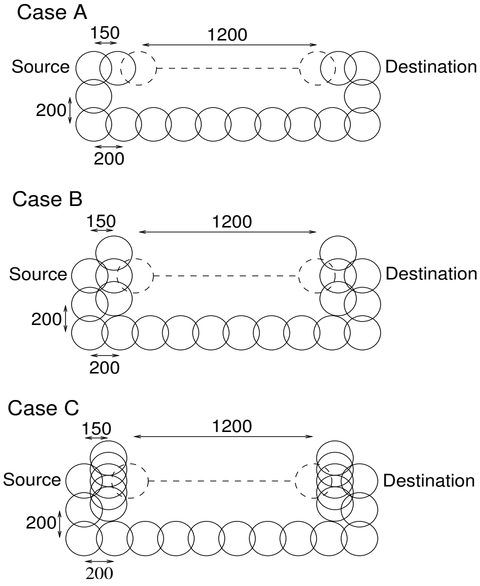

**Fig. 12**. The topology for the test cases.

The convergence results shown in Table 2 for Case A show good route discovery times of 75, 30 and 15 seconds respectively for the $L1,1,0$, $W1,1,0$ and $W1,1,1$ algorithms. Notice that the path length for the bad path is 3 while the path length for the good path is 13 hops. Thus the algorithm must weight the bad path by at least an additional 10 points in order for the source node to consider choosing the good path. For the $L1,1,0$ algorithm, the source node must discover and fault isolate the faulty link eleven times in order to increase the weight of the bad path by more than ten points. Given an application flow rate of roughly 5 packets per second, a threshold of 0.1 and a sliding window of 100 packets, then the time to discover and isolate a link is going to be longer than four seconds each round. This, in addition to route discovery times, transmission times, etc. result in a route time of 75 seconds. For $W1,1,0$ weighting, due to the exponential weighting based upon the number of faults, the source only needs to fault isolate the bad link four times in order for the source to increase the bad route length by more than 10 points. This cuts the good path discovery time down by more than half, compared to the $L1,1,0$ learning algorithm. The $W1,1,1$ algorithm cuts this in half again because of the triple counting of the link faults, once for the nodal weighting of the node on the left side of the link, once for the link weighting and once for the nodal weighting for the node on the right hand side of the link.

For the Cases B and C, the combinations are multiplied due to the number of virtual paths to search. For Case B, where there are nine virtual paths to search, the discovery times have increased by roughly nine times over the results in Case A. For Case C, where there are 25 virtual paths to search, the discovery times have increased by roughly 25 times over the results in Case A. The effectiveness of this attack increases in proportion to the square of the number of nodes within the vicinity of the MAC-level nodes, i.e., the learning times of the algorithms scale like $z^2$ where $z$ is the average number of nodes within radio range of the adversaries. Further, the different learning algorithms, i.e., $Ln,n,0$ and $Wn,n,0$, affect convergence times by modifying the constant of proportionality but not by fundamentally changing the $z^2$ scaling. However, the Nodal Weighting algorithm, $W1,1,1$, in addition decreases the potential number of search options. The learning time for the Nodal Weighting case increases in proportion to the number of nodes $z$ within range of the adversary. This is verified by the simulation results.

A potential problem with Nodal Weighting schemes is that they allow for a relatively large number of colluding adversaries to effectively ‘gang up’ on a good node within the vicinity of colluding adversaries. Because a weight is associated with each node comprising a bad link, the nodal weight of the good node will increase in proportion to the number of associated bad nodes. We believe that the algorithms will still perform correctly in these cases, however the convergence properties may not be as favorable.

We now investigate the performance of the Nodal Weighting algorithm within a MANET. We want to ensure that introducing Nodal Weighting within our MANET model does not adversely affect the performance of the ODSBR learning capabilities. For these simulations, the long time scale half life is predicted to be $(L\tau_{topology}/\alpha)ln2$ where $\alpha \approx (1-g_0)/g_0$, $L$ is the number of potential links in the MANET and $\tau_{topology}$ is roughly $r/v$. Assuming $g_0 \approx 0.8$, $r = 250m$, $v = 5m/s$ and $L = (60)^2/2$, we get roughly 7 hours for the long time scale learning half life. This is an extremely long time is due to a) the number of links is proportional to $N^2$ and b) the random nature of the measurement strategy. Nodal weighting has the potential for effectively reducing $L$ from $N^2/2$ to $N$. This reduces the long time scale learning half life by a factor of 30; reducing the 7 hours to 14 minutes. This is a rather dramatic improvement. It is only possible to reduce this long time scale learning further by incorporating some form of shared learning between the good nodes in the network. One possible scheme is suggested in [2]. Other schemes are possible. This is a challenging area of study due to the security implications of shared trust in Byzantine environments.

Figures 6, 7 and 8 present the convergence results for the ODSBR short time scale Nodal Weighting learning algorithm in the context of our initial MANET model. The plots on the LHS show the results of the W1,1,0 learning algorithm, while the plots on the RHS of the figure show the corresponding results for the W1,1,1 learning algorithm. There appears to be little difference between the convergence of the ODSBR short time scale behavior between the non-Nodal and the Nodal Weighting algorithms. Clearly the introduction of the W1,1,1 learning algorithm does not adversely affect the ODSBR protocol performance. Nodal Weighting does show improvement for all mobilities in the presence of BHFSW attackers. We expect that the Nodal Weighting algorithm shows improvement in the BHFSW scenarios due to the relatively large number of search paths required of the learning algorithms.

We next investigate performance in the presence of the MAC-Level attackers in a MANET. For this set of simulation studies we had to modify the topology from our previous square topology to a rectangular topology. This was necessary in order to separate the MAC-level colluding nodes far enough such that they did not interfere with each other's transmissions. The MAC-level adversaries are placed at the locations given by coordinates (225,250) and (1775,250) in the rectangular grid with dimensions of 2000 by 500 meters. In order to keep the colluding MAC-level nodes from picking up identical CTS transmissions, we must keep their separation larger than 700 meters, which this topology does. For comparison of results, we kept the nodal density unchanged by choosing a topology with the same area as our previous simulation studies.

Figure 11 shows the simulation results on the above discussed topology and MAC-level attacks. The time duration for these simulation runs is 60 seconds. The plot on the LHS shows the results for mobilities of 0, 1, 5, and 10 mps with ODSBR nodes running the non-Nodal Weighting algorithm, W,1,1,0. While the plot on the RHS shows the results for mobilities of 0, 1, 5, and 10 mps with ODSBR nodes running the Nodal Weighting algorithm, W,1,1,1. The results for the non-Nodal Weighting algorithm show that the algorithm is having a hard time finding good routes through the network. However, the Nodal Weighting results, on the RHS of the figure, demonstrate a trend toward convergence. The demonstrated improvement in the Nodal Weighting algorithm, in our static topology cases above, represents the long time scale dynamics of the algorithms. While these MANET simulation results, which represent only 60 seconds of run time, demonstrate the short time scale dynamics of the ODSBR protocol. Even here, we are seeing some improvement in the short time scale dynamics of the protocol with Nodal Weighting.

## 7. CONCLUSIONS

We investigated the benefits of several ODSBR protocol enhancements through analytic and simulation modeling. The specific enhancements analyzed are a) shorter packets for increase flow rates, b) network layer retransmissions, and c) the introduction of Nodal Weighting to the ODSBR reputation database.. Further, we analyzed a new attack model, termed the MAC-Level attacker. This attack model forces the routing protocol to search through a large number of adversarial paths in order to find existing good paths. As such this attack model dramatically demonstrates the differences in Link and Nodal Weighting schemes within the ODSBR protocol.

We analyzed our proposed protocol enhancements through an analytic model of the protocol dynamics and through extensive simulation studies. These simulations were performed for a range of network parameters including varying nodal speeds, different Byzantine attack models and different network topologies. These studies demonstrated improved convergence times due to the proposed protocol enhancements. However, the impact of a network layer retransmission protocol was not as great as we expected due to the relatively large value of the retransmission time outs with respect to the fault isolation times of the ODSBR protocol. The incorporation of the Nodal Weighting scheme showed dramatic improvements in searching through complex topologies introduced by the MAC-Level adversarial attack model and hence improved the convergence time performance of the ODSBR protocol.

In future work, we plan on additional studies of the Network Layer Retransmission protocol in carrying TCP traffic and in protecting against other attack models. However, the most notable and challenging future work item is the investigation of methods for good nodes to share information learned regarding the faults in the network with other nodes. Several possibilities are under study and present an extremely interesting area for research.

## 8. ACKNOWLEDGMENTS

We would like to thank Cristina Nita-Rotaru of Purdue University for her advise and comments on aspects of this paper and the ODSBR protocol.

### References

1. Aad, I., Hubaux, J.-P. and E.W. Knightly, *Denial of Service Resilience in Ad Hoc Networks*, Mobicom'04, Philadelphia, September 2004.
2. Avramopoulos, I., Kobayashi, H., Wang, R. and A. Krishnamurthy, *Highly Secure and Efficient Routing*, INFOCOM'04, March 2004.
3. Awerbuch, B., R., Holmer, D., Nita-Rotaru, C. and H. Rubens, *An on-demand secure routing protocol resilient to byzantine failures*, ACM Workshop on Wireless Security (WiSe), September 2002.
4. Awerbuch, B., Curtmola, R., Holmer, D., Nita-Rotaru, C. and H. Rubens, *On the Survivability of Routing Protocols in Ad Hoc Wireless Networks*, SecureCom'05, September 2005.
5. Awerbuch, B., Cole, R.G., Curtmola, R., Holmer, D., Nita-Rotaru, C. and H. Rubens, *Dynamics of learning Algorithms for the On-Demand Secure Byzantine Routing Protocol*, JHU Applied Physics Laboratory Technical Report No. VIC-05-088, November 2005.
6. Boneh, D., Gentry, C., Shacham, H. and B. Lynn, *Aggregate and verifiable encrypted signatures from bilinear maps*, in Proceedings of Eurocrypt'03, 2003.
7. Buttyan, L. and J.P.Hubaux, *Enforcing Service Availability in Mobile Ad-Hoc WANs*, In Proceeding of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc), Boston, MA, USA, August 2000.
8. Feller, W., *An Introduction to Probability Theory and Its Applications*, John Wiley and Sons, New York, NY, 1968.
9. Hu, Y.-C., Perrig, A. and D.B.Johnson, *Ariadne: A secure on-demand routing protocol for ad hoc networks*, In Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002), September 2002.
10. Kent, S., Lynn, C. and K. Seo, *Secure Border Gateway Protocol (S-BGP)*, IEEE JSAC Issue of Network Security, Vol. 18, No. 4, April 2000.
11. Lamport, L., Shostak, R. and M. Pease, *The Byzantine Generals Problem*, ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982.
12. Marti, S., Giuli, T.J., Lai, K. and M. Baker, *Mitigating routing misbehavior in mobile ad hoc networks*, In Mobile Computing and Networking (MobiCom 2000), September 2000.
13. Murphy, S., Badger, M. and B. Wellington, *OSPF with Digital Signatures*, IETF RFC 2154, June 1997.
14. The Network Simulator - ns2, *http://www.isi.edu/nsnam/ns/*, 2005.
15. Papadimitratos, P. and Z. Haas, *Secure routing for mobile ad hoc networks*, In SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, 2002.
16. Papadimitratos, P. and Z. Haas, *Secure data transmission in mobile ad hoc networks*, In ACM Workshop on Wireless Security (WiSe), 2003.
17. Perlman, R., *Network Layer Protocols with Byzantine Robustness*, MIT Thesis, August 1988.
18. Rappaport, T. *Wireless Communications Principles and Practice*, 2nd Ed., Pearson Education Press (Singapore), 2002.
19. White, R., *Securing BGP Through Secure Origin BGP*, The Internet Protocol Journal, June 2003.
20. Yoon, J., Liu, M. and B.D. Noble, *Random waypoint considered harmful*, INFOCOM'03, April 2003.