SAND REPORT

SAND2002-3340
Unlimited Release
Printed October 2002

Performance Impacts of Lower-Layer Cryptographic Methods in Mobile Wireless Ad Hoc Networks

B. P. Van Leeuwen and M. D. Torgerson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2002-3340
Unlimited Release
Printed October 2002

Performance Impacts of Lower-Layer Cryptographic Methods in Mobile Wireless Ad Hoc Networks

Brian Van Leeuwen
Networked Systems Survivability and Assurance

Mark Torgerson
Cryptography and Information Systems Surety Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0785

Abstract

In high consequence systems, all layers of the protocol stack need security features. If network and data-link layer control messages are not secured, a network may be open to adversarial manipulation. The open nature of the wireless channel makes mobile wireless mobile ad hoc networks (MANETs) especially vulnerable to control plane manipulation. The objective of this research is to investigate MANET performance issues when cryptographic processing delays are applied at the data-link layer. The results of analysis are combined with modeling and simulation experiments to show that network performance in MANETs is highly sensitive to the cryptographic overhead.

This page intentionally left blank.

Contents

Figures

Tables

1. Introduction

Mobile Ad Hoc Networks (MANETs) are self-organizing networks that do not depend on a fixed communication infrastructure and are made up of mobile nodes that communicate via a wireless channel. The mobile nodes are typically resource constrained, have limited transmit capability, and in many cases have poor antenna placement resulting in limited communication range. A MANET overcomes these limitations by having its nodes assist each other by forwarding messages to their destinations. Since these nodes are mobile, the relay points are not stationary. The network must have mechanisms to maintain a current view of the dynamic network topology. Routing protocols are employed in MANETs to acquire enough topological information to allow application layer message delivery. Routing protocols depend on an exchange of routing information between network nodes.

MANETs exist to deliver application-layer data. If the integrity of the routing messages is compromised, then the network may suffer in its ability to transport data. Thus, methods must be employed to protect the integrity of the routing messages.

The research community has directed a great deal of attention toward the development of efficient routing protocols for MANETs [4, 7, 12, 15, 20, 37, 38, 39]. However, only recently have efforts been directed toward the development of secure routing methods [8, 10, 17, 18, 19, 33, 38, 43]. We do not discuss the security claims of any security method. This research examines security features from an efficiency point of view. We examine network performance issues that arise from increasing message sizes to account for authentication as well as those issues associated with applying processing delays of standard security protocols to routing messages. Our findings indicate that in many cases cryptographic delays destroy network functionality.

1.1 Lower-Layer Data Security in MANETs

MANETs communicate over an open wireless channel, which provides no physical protection to the data flows. Adversaries may acquire the data flows from places that are outside the physical control of the network and may do so with little chance of being detected. In many environments the protection of application-layer is sufficient for the needs of the network. However, there is a wealth of information to be found in the messages that originate at the lower layers of the OSI protocol stack. An adversary may passively monitor the network to determine its logical topology, determine critical nodes and their function. In a more active sense, an adversary may listen to network traffic, record and replay transmissions, and possibly inject false control information within the network. A MANET that does not secure its routing messages can be rendered non-functional when an adversary injects false topological information into the network [41]. In environments where a high-level of information security is required, all aspects of the network communications must be protected. In particular, data security features such as digital signatures and/or encryption must be applied to the lower-layer messages.

Wireless networks have special needs that require special solutions. Because of the open channel and resource constraints, security solutions that apply to wired networks do not necessarily have

utility in the wireless environment. Mobile nodes contend for a limited communication channel and may operate with extreme resource constraints. They have limited channel capacity, battery power, processing power, memory, etc. Because of these constraints, mobile devices tend to have operating systems and other software applications that are pared down to minimal size resulting in limited capabilities.

The resource constraints found in wireless networks are at odds with the fact that cryptographic primitives tend to be resource hungry. Authentication and/or encryption procedures consume time, computational resources, and have a bandwidth overhead. The addition of cryptographic processing delays and bandwidth overhead of appended signatures to application-layer traffic may not terribly inconvenience the users of the network. However, control messages are typically short and may be large in number. The functionality of the network depends on the fact that these control messages must be delivered and processed in near real-time. Significant delays and/or size overhead in these messages can negatively impact their temporal requirements. Further, cryptographic overheads incurred on the control messages may cause network congestion that causes unacceptable delays in the higher-layer messages. One must carefully analyze the needs of the network's lower-layer protocols to understand how cryptographic overhead and delays placed on control messages impact network throughput.

In general, the actions that can be taken in a MANET to prevent an adversary from compromising the network by protecting the data transmission fall into two categories [1]:

1. Prevent the transmission over the wireless medium from being detected through the use of physical-layer techniques.

2. Protect transmitted data with cryptographic techniques.

A wireless network may incorporate physical-layer techniques such as spread-spectrum modulation, power control, directional antennas, etc to make it more difficult to detect and manipulate data transmitted over a wireless channel. However, completely eliminating the possibility to detect the signal is impossible. Thus, cryptographic techniques play an important role in providing data security in a MANET.

1.2 Report Overview

In Section 2 we provide a description of lower-layer protocols used in MANETs. A description of the general aspects for the lower-layer protocols is presented and then a discussion of the particulars of the protocols that we chose for our simulations is provided. Section 3 provides a discussion of the performance criteria our analysis and simulations investigated. The section also includes a discussion on the parameters that are used in the simulation models and an explanation addressing their importance. Section 4 presents our analysis and simulation results. Sections 5 and 6 describe possible future work and conclusions, respectively. Finally, we include Appendix A that provides a short description of the modeling and simulation environment used in our research and details on the specific models used.

2. Overview of MANET Lower-Layer Protocols

This section gives a detailed description of important aspects the network layer, the zone routing protocol, the media access layer, and the IEEE 802.11 media access approach. Those familiar with these topics may wish to skip to Section 3.

Protocols to support MANET operation are located in the lower layers of the Open Systems Interconnect (OSI) model for telecommunication protocols. MANET methods are implemented in the network, data-link, and physical layers. The network layer includes the routing protocol. The data-link layer includes the Neighbor Discovery Protocol (NDP) and the Media Access Control (MAC) protocol. The physical layer addresses the transfer of data across the physical medium. Other than efficiency issues that impact the effective data rate, we do not address physical layer protocols. We focus on the network and data-link layer protocols, and refer to these two layers as the lower layers. By upper layers, we mean the layers that lie above the network layer in the OSI model.

Below we describe general aspects for the lower-layer protocols and then discuss the particulars of the protocols that we chose for our studies. There are several different MANET lower-layer protocols to choose from, each with its unique set of operational and computational aspects. The protocols selected provide a sample of ordinary MANET operation.

2.1 Network Layer

The network layer provides routing support for MANETs. Many different MANET routing protocols have been proposed [37]. These are divided into three categories based on the overall approach to establishing and maintaining a view of the network topology. The three categories are *(1) proactive, (2) reactive,* and (3) *hybrid.* Proactive routing protocols attempt to maintain routing tables that continuously reflect the current state of the network's view of its topology. Proactive approaches act independent of any application-layer traffic. The primary goal is to have an up-to-date route to each destination. This route is based on the node's current view of the network topology. In contrast, reactive routing responds to the demands of the upper-layer traffic. No routes are determined until an upper-layer request is made. Hybrid approaches to routing include both a proactive and a reactive region that together support the route determination procedure.

The advantage of a proactive approach is that when a multihop route is needed, a routing table is inquired and a route is immediately available. This routing approach has the benefit of minimizing packet latency. However, a proactive approach comes with a penalty of increased control-message traffic overhead. This traffic continuously uses a portion of the network capacity to keep the routing information current. Precious resources may be wasted in order to proactively maintain paths that are not needed.

The advantage of a reactive approach to MANET routing is that there is little or no wasted control-message traffic to discover routes that are never used. Control-messages are generated in direct response to an upper-layer route request. Implementations of routing protocols may take advantage of previous route requests by storing previously used routes, or they may take

advantage of route requests that have been forwarded through the node. However, if no route information is available when a route request is initiated, the time to discover a route can be quite high, because an inefficient global search is required to find the unknown node. Global searches can consume significant network resources.

Hybrid protocols generally lie between the proactive and reactive extremes in terms of control-message overhead and message latency. To maintain the proactive region a certain amount of control-message overhead must be generated. Unfortunately, there will be times where some of this proactive messaging will be wasted. However, if the proactive region is relatively small, there is a high probability that reactive messaging will be used as messages are forwarded through and beyond the proactive region. A node's proactive region may not contain the entire network, but it may be exploited to simplify the search for a route. This simplification reduces the overall message latency.

2.1.1 The Zone Routing Protocol

We selected the Zone Routing Protocol (ZRP) [15, 34, 35] for our study of MANET control traffic and implemented it for our simulations. This protocol was selected for our simulations because it is a novel hybrid routing protocol that uses an adjustable zone radius that can be tuned to optimize performance as a function of application-layer traffic and network mobility. ZRP is versatile and can be applied to a large range of applications. However, the methods use below can be adapted to work with any routing protocol.

ZRP divides each node's view of the network into two regions or zones: the local *intrazone* and extended *interzone*. To determine its intrazone, a node fixes a zone radius *r*. The node's intrazone is comprised of the nodes that lie within *r* hops of the node. Each node proactively maintains a logical view of its intrazone. This logical view is maintained via a routing table, which contains a route to each node in the intrazone. The network nodes that lie outside the intrazone comprise the node's interzone. Nodes do not proactively maintain route information about nodes in the interzone and must discover routes to nodes in the interzone when needed. However, a node may store a few previously discovered routes to nodes in its interzone. Further, a node uses its knowledge of the intrazone topology to improve the efficiency of the interzone route discovery process, by guiding route requests to the edge of its intrazone.

ZRP is placed at the network layer of the OSI protocol stack and is shown in Figure 2.1. It is comprised of two sub-protocols; the Intrazone Routing Protocol (IARP) and the Interzone Routing Protocol (IERP). The IARP provides proactive routing table generation and maintenance for the intrazone. It incorporates a data-link layer Neighbor Discovery Protocol (NDP) to discover a node's one-hop neighbors. The IERP provides reactive route discovery to nodes in the interzone.

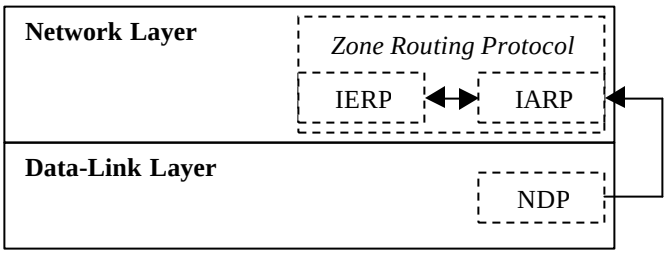

Figure 2.1: ZRP is included in the network layer and receives interprocess communication from the Neighbor Discovery Protocol (NDP) located in the data-link layer

Early white paper descriptions of ZRP [37] did not specify the exact nature of the NDP, IARP, or the IERP. The designers felt that any appropriate protocol would be sufficient. Implementers were encouraged to choose their favorite (and hopefully most efficient) component protocols. The designers also felt that any improvements in proactive or reactive protocol technology could be applied directly to ZRP. Implementers have gravitated toward certain methodologies for the component algorithms. For our simulations and analysis we examined a specific implementation of ZRP obtained from its original designers. This implementation is in line with the current state of the art and in line with standardization proposals [15]. From this point on, we describe the basic functionality of the specific implementation that we used for our studies. When we refer to ZRP we mean the version of ZRP that we obtained from the original designers.

Neighbor Discovery Protocol:

The IARP depends on the NDP to determine a node's one-hop neighbors. The NDP advertises a node's presence by periodically transmitting a Hello packet that contains the transmitting node's address. All neighboring nodes in the range of the transmitting node receive the beaconed Hello message. The NDP maintains a list of nodes from which it has received a Hello message, and initiates an interrupt that informs the IARP of any change in the one-hop neighbor list. Changes occur when the NDP receives a Hello from a node that is not in the neighbor list, or if it has not received a Hello message from a node in the list for a specified number of beacon intervals.

A node's transmitter power, antenna configuration, receiver sensitivity, and channel characteristics help in determining its set of one-hop neighbors. It should be noted that a node's one-hop neighbor region is mostly based on physical distance. In contrast, a node's zone radius is not a physical distance, but rather it is based on node relaying connectivity as shown in Figure 2.2.

There are overhead impacts associated with any proactive protocol based on the periodic beaconing of Hello messages. For ZRP the Hello messages are short, on the order of 34 bits. Even though these messages are relatively small, each node generates a fixed number of them per second regardless of node mobility and network topological changes. Periodic beaconing reduces the amount of bandwidth available for higher level messaging. The number of beacons that a node receives is directly proportional to the number of nodes in its one-hop neighborhood. The bandwidth impacts due to beaconing are greatest in dense networks.

Intrazone Routing Protocol:

Depending on how the network is initialized, the zone radius may be fixed and constant throughout the network, or each node may have the ability to monitor network traffic and determine the radius based on a predetermined set of rules. In order to explain the basics of the IARP we assume all nodes have the same zone radius.

The NDP provides the IARP with a list of one-hop neighbors. Each time the IARP receives a change in its list of neighbors, or on a periodic basis, it sends to all neighboring nodes its current one-hop neighbor list or link-state. Each node forwards any received link-state information. So, a node's link-state propagates throughout its intrazone. Using a minimum spanning tree algorithm and the collection of one-hop neighbor lists, each node computes a routing table. The routing table is recomputed whenever there is a change in the link-state information. Changes in the link-state information can occur when a node discovers a new, or loses an existing, one-hop neighbor or when the node is informed of a change in another node's link-state.

Interzone Routing Protocol:

IERP uses bordercasting to efficiently propagate its route discovery messages through the network. A node requiring a route to a node in its extended region will send a route request to the nodes at the peripheral of its IARP region. Each of these peripheral nodes will check their routing table to determine if they have a route to the destination. If no route is found, the peripheral node appends its address to the route request and forwards the request to its peripheral nodes.

When a node receives a route request for a node that lies within its intrazone, it will append its address and return a route reply along the list of nodes in the request. The source node stores and uses the discovered route as needed. If multiple routes are returned to the source node, the most efficient route is chosen and stored.

IERP bordercasting obtains efficiency over the standard "flood" approach by directing its searches to IARP region peripheral nodes. When interior nodes forward requests to peripheral nodes they record enough information to terminate redundant requests. Further, non-forwarding interior nodes promiscuously listen to requests so that they can also terminate future redundant requests. Details of these directed search mechanisms are described in detail in [35].

Figure 2.2 illustrates node S's two-hop intrazone. Peripheral nodes are the outermost nodes in the region and participate in the extended message routing. Nodes A, B, C, D, and E are S's one-hop neighbors. Nodes G, H, I, and K are its peripheral nodes. Nodes F and J are in the interzone of node S.

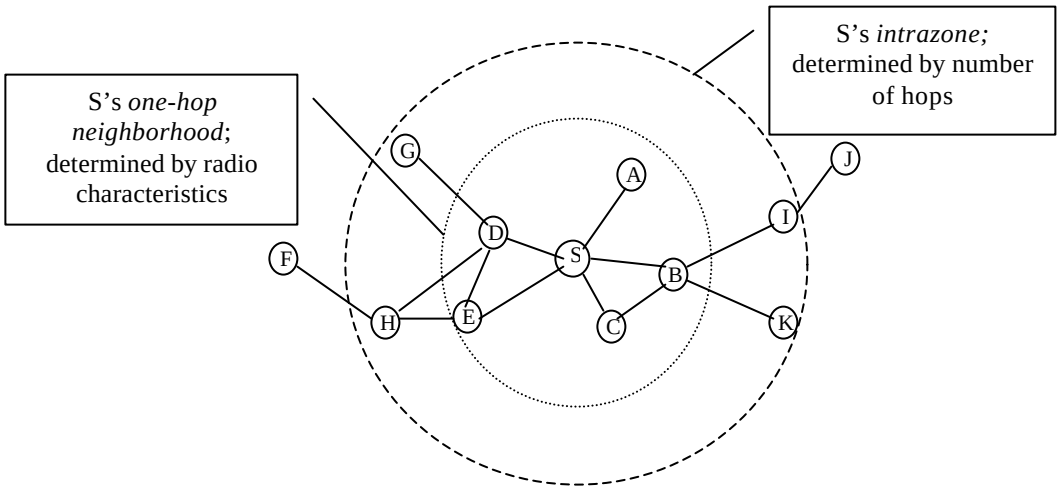

Figure 2.2: ZRP with an IARP two-hop intrazone.

2.2 Data-Link Layer

The Data-Link layer is the protocol layer that handles the interfacing of data to the physical link in a network. The Data-Link layer is layer two in the OSI model and contains two sub-layers that are described in the IEEE-802 LAN standards as follows:

- Logical Link Control (LLC)
- Media Access Control (MAC)

The LLC is concerned with managing traffic, both flow and error control, over the physical media.

The Data-Link layer initiates a communication sequence, divides output data into data frames, and handles the acknowledgements from the receiver that data arrived successfully. It also ensures that incoming data has been received correctly by analyzing bit patterns at special places in the frames.

The MAC sub-layer is responsible for controlling access to the wireless channel. In a large wireless network many nodes share the same channel, so there is a high probability that nodes will compete for channel resources. Part of the duties of the MAC is to prevent contentions if possible, and to resolve them when they do occur.

A number of approaches have been recommended for a wireless network to access the shared wireless channel [13]. One commonly used approach is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [3]. Under the CSMA/CA technique, a node that intends to transmit first listens to the channel to determine if any other node is transmitting. If the channel is busy, it will wait for a period of time. If the channel is free for a specific period of time, then the node transmits. Collisions occur only when two or more nodes select the same time to

transmit. If two nodes transmit at the same time and their messages collide, they will each back off for a period of time before attempting a retransmission.

One issue that arises in a wireless network (which does not arise in a wired network) is the *hidden terminal problem*. The hidden terminal problem describes a situation where node A would like to transmit a message to node B. Node C is in communication range of B but not in communication range of node A as shown in Figure 2.3. Node C will be unaware of node A's transmissions. Even if node C senses the channel before transmitting, that message will corrupt node A's transmission. This corruption is independent of node C's intended destination.

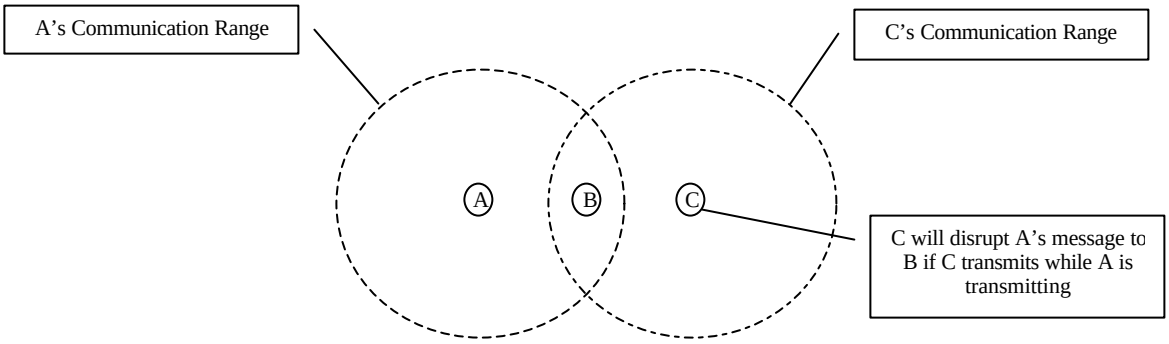

Figure 2.3: Wireless communication hidden terminal problem

2.2.1 IEEE 802.11 Media Access Control

Because of the hidden terminal problem, the CSMA/CA methodology alone cannot totally prevent collisions. Various implementations of CSMA/CA technology incorporate extensions that improve the collision avoidance property. One such method is the MAC defined by the IEEE 802.11 standard [2, 3]. This MAC has a number of features that mitigate many of the media access issues that arise in wireless LANs and is used because of its versatility and generally accepted robustness. Because of its popularity, it is the method of choice to benchmark approaches that deal with wireless media access.

The IEEE 802.11 standard uses an access mechanism called the Distributed Coordination Function (DCF) [3]. In addition to the typical CSMA/CA features, the DCF of the IEEE 802.11 MAC provides for packet fragmentation and reassembly, packet retransmissions, and acknowledgements. To reduce the possibility of collisions, the DCF also employs a virtual channel sense based on a request to send/clear to send (RTS/CTS) exchange. This approach adds stability to the channel at the cost of a small amount of overhead.

The IEEE 802.11 standard calls for inter-frame spacing times. The times are fixed in size relative to each other, but the actual value is based on the implementation hardware. However, the IEEE 802.11 standard suggests the times given below:

- Slot Time (Slot) – The slot time is the time that it takes for a node to determine if the channel has been accessed at the beginning of the previous slot. Typically a 50-microsecond spacing.

- Short Inter Frame Space (SIFS) – The SIFS is the time necessary for a node to process an incoming transmission and switch between transmit/receive as necessary. Typically a 28-microsecond spacing.

- Distributed Inter Frame Space (DIFS) – The DIFS is defined as a SIFS plus two slots. Typically a 128-microsecond spacing.

The following sequence of events are followed when node A wishes to send a packet to node B. Node A first senses the channel. If the channel is clear for a DIFS, node A transmits. If the channel is busy, node A waits until the channel is clear for a DIFS and then enters a random back-off mode. In the random back-off mode, node A sets a decrementing timer to a randomly chosen number of slots. If the channel becomes busy during the countdown, node A suspends counting until the channel is clear again. When the timer reaches zero, node A checks the channel to verify it is clear and then transmits. If the counter reaches zero, but the channel is busy, node A will reset the timer to an exponentially larger value. After a number of unsuccessful attempts the packet is dropped.

Now suppose that node A wishes to send a data packet to node B. Node A uses the transmission sequence described in the paragraph above to send a RTS to node B. If node B receives the RTS, it returns a CTS. Once node A receives the CTS from node B, it sends the data packet. When node B correctly receives the data packet, it sends an acknowledgement (ACK). When node A receives the ACK the exchange is complete. If at any time node A does not receive a CTS or ACK, it will attempt to resend the appropriate RTS or data packet. The data packet will be dropped by node A after a number of unsuccessful sending attempts. We refer to this process as the Link Layer Data Transaction (LLDT).

The RTS and the CTS both contain source and destination information. The RTS also includes the expected amount of time that the entire transaction will take. This time includes the CTS, DATA and ACK transmission times plus the appropriate SIFS message processing times. The CTS includes the timing information for the DATA and ACK transmission plus the appropriate SIFS message processing times. Nodes other than B that receive the RTS set their Network Allocation Vector (NAV) to the time set in the RTS. Nodes other than A that receive the CTS set their NAV to the time in the CTS. These nodes will not transmit for the time specified by the NAV. Figure 2.4 shows an exchange between nodes that use the LLDT virtual-channel sense.

Because of the overhead associated with the RTS/CTS exchange, small packets such as Hello messages are sent directly without the RTS/CTS exchange. Broadcast packets are also sent without the RTS/CTS exchange.

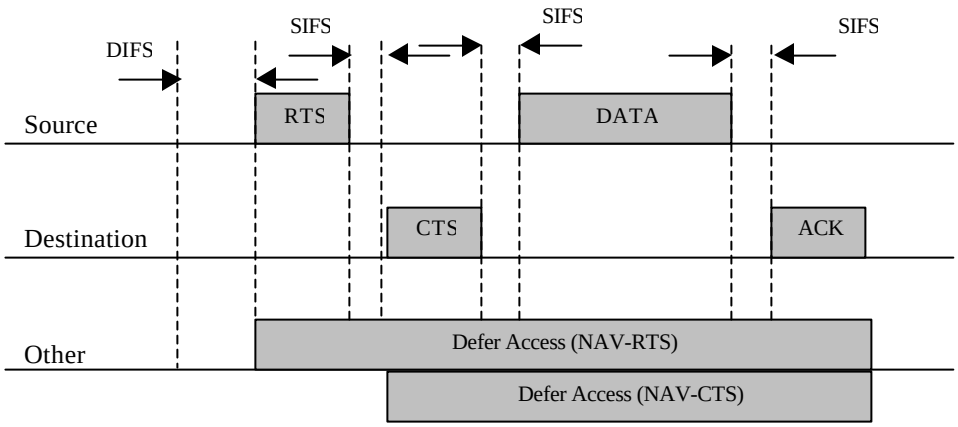

Figure 2.4: Media access control with virtual channel sense message exchange

The IEEE 802.11 MAC standard also provides a fragmentation and reassembly service for the data packets. Since the standard interfaces to higher layers that are based on Ethernet environments, data packet size can be up to the maximum Ethernet packet size; 1518 bytes long. In wireless communications long packets can decrease network performance since bit error rates are higher than in a wired network. If a bit error occurs in a packet transmission, the entire packet is lost. Thus, to maximize network performance, the selected packet size must be optimized based on packet header overhead and bit error rates.

3. Evaluation of Lower-Layer Protocols in MANETs

In the development of protocols for wireless MANETs, a number of criteria are used to measure the performance of the protocols [9, 30]. Here we do not compare protocols, we measure the impacts that cryptographic overheads have on network performance given the protocols described in the previous section. Below is a list of network aspects that we use to evaluate network performance and measure cryptographic overhead.

- Network Capacity –Network capacity is the total number of application-layer data bits that can be successfully transmitted by the network per unit of time. Network capacity is a function of bandwidth, number of nodes, number of channels, node density and other network topology aspects, etc. When viewed as a function of the number of nodes, network capacity is $O(N)$, where N is the total number of nodes in the network.

- End-to-End Capacity – In a multihop network, end-to-end capacity is the total number of application-layer data bits that can be successfully delivered to their intended destination by the network per unit of time. End-to-end capacity is a function of the network capacity and the number of hops required to deliver messages. In a model where intended destinations are chosen at random, the average number of hops to deliver a message is proportional to the network diameter, which is $O(\ddot{O}N)$. Thus, in this model end-to-end capacity is $O(N/\ddot{O}N)$.

- One-Hop Throughput – One-hop throughput is the average number of application-layer data bits that can be successfully transmitted from one node to another per unit of time. One-hop throughput is a function of bandwidth, node density, other network topology aspects, number of channels, etc. When viewed as a function of the number of nodes, network capacity is an *O(1)* constant.

- End-to-End Throughput – In a multihop network, end-to-end throughput is the average number of data bits that can be successfully delivered from a source to a destination per unit of time. End-to-end throughput is a function of one-hop throughput and the average number of hops required to deliver messages. In the random destination model with an average path length of *O(ÖN),* end-to-end throughput is *O(1/ÖN).*

- Route Acquisition Time – The time that it takes to discover a route from a source to a destination. This is mainly a function of the intrazone radius, average path length, and the size of the extended region route buffer. In our simulations we set the time that that it takes to examine the route table plus the time it takes to examine the route buffer to zero. The route acquisition time is measured only for destinations that lie outside the intrazone and not in the route buffer.

- Propagation Time – The time that it takes for a data message to traverse from source to destination. This depends on the path length, the time to complete a LLDT, and the amount of message queuing/delay at intermediate hops.

- End-to-End Data Delay – The time that it takes to deliver an application-layer packet. This is a function of route acquisition time and transmission time.

- Route Table Convergence Time – The time it takes for a node to acquire a stable view of its intrazone. This is a function of the intrazone radius, propagation time, Hello beacon interval, and node velocity.

3.1 Simulated Network Parameters

For our studies we have targeted a few characteristics that play a role in MANET operation. These aspects are variable parameters in our simulations.

Network Node Mobility:

Mobility impacts the performance of a MANET since the processes of node discovery, IARP update rate, route discovery, and route rediscovery are dependent on node movement. For maximum network performance, the Hello beacon rate and IARP update rate should be a function of the average node velocity. In general, a neighbor discovery beacon should be transmitted several times during the life of a link. This will ensure that the local neighborhood is fairly stable throughout several beacon periods. If the rate is too high, the network topology may not change during scores of beacon periods and thus waste precious bandwidth. On the other hand, if the beacon rate is too low as compared to node velocity, a node may never have a correct view of its one-hop neighborhood.

A typical method of determining the beacon rate is to set its value to some fraction of the transmit radius divided by the average node speed. However, this course method is optimal only in situations where the network nodes all move in random directions at approximately the same speed. In real networks a random direction, equal speed movement pattern is not realistic, nodes may have a wide range of speeds, and directions are not usually random.

Ideally to obtain the optimum beacon rate, each node would have to be able to measure its own speed as well as its speed relative to the other nodes and then adjust its own beacon rate for those conditions. To add the capability to incorporate node speed into the beacon rate, a node would have to not only have the ability to determine its own speed, the MAC layer protocols would have to be extended to dynamically adjust the beacon rate. At this time no hardware or software enhancements of this type are available.

Our focus is on ground-based ad hoc networks. As such, simulated node velocity is limited to that which is reasonable for ground-based vehicles. Simulated nodes are given a random direction on a two-dimensional rectangular grid with boundary. When nodes intersect the boundary they are redirected with equal angle.

In our simulations and comparable research, the beacon rate is fixed to a value that is high enough to accommodate all reasonable node velocities. For most of our simulations we set this value to be 0.33 seconds. Incidentally, this value is approximately $0.036(r/s)$, where r is the transmit radius (chosen here as 300 meters) and s is a maximum closing speed of 120 km/hr.

In practice, it is sufficient for the IARP update period to be three times that of the beacon period [36].

Mobility also affects a node's ability to acquire and maintain reactive routes. If node speed is high enough, queued routes quickly become invalid. This causes frequent applications of the route discovery procedure. The network can waste considerable bandwidth to constantly update older routes. In the extreme, even the route discovery process can be impeded by mobility. Routes may become invalid before they can be used. In this case, "flooding" of application layer messages may be the only recourse for message delivery.

Application-Layer Traffic Load:

In our studies we have chosen a random destination model, where the destination of application-layer messages is chosen at random. An increase of packets delivered into the proactive region does not increase control message overhead. However, if the application-layer traffic is low, then the effort to maintain a large proactive region is wasteful.

In ZRP, the number of control messages generated to support route discovery into the reactive region is a function the application-layer traffic. If a series of application-layer packets is destined to a single node, then a single route discovery is initiated to establish the route for this series. On the other hand, if each packet in the series were destined to a different location, then each packet would require a route discovery. Clearly, the amount of control traffic overhead is a function of the number of destinations. One must keep in mind that mobility will cause previously discovered routes to become invalid. These invalid routes must be rediscovered, thus generating additional control traffic. If control and application-layer traffic must share the same

capacity limited channel, then excessive control traffic will have a direct impact on the amount of application-layer traffic that can be transmitted throughout the network.

Three different rates of application-layer packet generation were selected for our simulations. The application-layer generates the traffic for a session. In one session, a node randomly chooses a destination, and then sends a fixed number of 1000 bit packets to that destination. The node then waits for a fixed amount of time before beginning another session. The three different generation rates are described in Table 3.1.

Table 3.1: Application-layer traffic generation

Application-Layer Traffic Destination Variation	Number of Packets per Session	Elapsed Time per Session	Time Between Sessions
High	2	0.125 sec	0.125 sec
Medium	10	0.625 sec	0.625 sec
Low	20	1.250 sec	1.250 sec

Additional routing protocol parameters that impact the successful delivery of application-layer packets to the destination are: *Update Packet List Period* and *Discard Old Packet Time.* These two times determine how long a packet will remain queued while the routing protocol attempts to discover a route. The *Update Packet List Period* defines the frequency that the protocol reviews the send queue. Any packet that has been in the send queue for a time longer than the *Discard Old Packet Time* will be removed from the queue and is considered as a dropped packet. Since we are evaluating lower protocol performance, we have not incorporated a higher layer reliable delivery mechanism. We count dropped packets, but make no attempt to retransmit them.

Cryptographic Overhead:

Typically, cryptography is applied to upper-layer messages. However, without confidentiality and authentication techniques applied to the lower-layer messages, the lower-layer is open for adversarial manipulation. For instance, an adversary may spontaneously generate an RTS indicating that a maximal length packet will be transmitted and then not transmit the data packet. Any node hearing this message will set its NAV to its maximum value and wait while nothing happens. Or the adversary may send out a false CTS and ACK. If the intended target node is not in range or otherwise busy, it will not receive the data packet and the packet will be lost and the source node will believe otherwise. In either case, the network will suffer considerable performance degradations [41].

For a MANET to be secure, security features should protect both the upper and lower-layer data. Every network message must have security features applied to it. This includes messages such as Hello, RTS, etc. The use of cryptography imposes computational costs on the nodes that implement cryptographic algorithms. These costs may include the time to encrypt, decrypt, sign, and verify messages. These operations increase packet latency and, in most cases, reduce data

throughput. Digital signatures appended to messages leads to increased packet size and reduces overall throughput.

To properly utilize cryptographic primitives, a network must have a viable security policy. This security policy must include a provision for key management and auditing procedures. Key distribution, exchange, archival, and recovery all impact the performance of the network. Invasive network audits can destroy network capacity. In addition, secure network initialization and configuration may be time and labor intensive. The performance impacts of a properly implemented security policy are beyond the scope of this paper.

Nodes must securely hold cryptographic keys as well as the code used to employ the cryptographic primitives. This does not impact the network steady-state performance, but does increase node complexity and hardware requirements.

A well-accepted set of speed benchmarks for cryptographic protocols provided by Wai Dai can be found at [11]. With regards to the implementation platform Wai Dai says, "All were coded in C++ or ported to C++ from C implementations, compiled with Microsoft Visual C++ 6.0 SP4 (optimize for speed, blend code generation), and ran on a Celeron 850MHz processor using Windows 2000. Two assembly routines were used for multiple-precision addition and subtraction."

The timing information in Table 3.2 has been taken directly from [11]. We have presented a sample of common signature algorithms. This list is by no means all-inclusive, but it does give a flavor for expected run times. The first three algorithms in the table are symmetric key signature methods and have processing times on par with most symmetric key encryption methods. Note that the symmetric methods are roughly three orders of magnitude faster than the public key methods.

Table 3.2: Execution time for various signature algorithms

Signature Algorithm	**Execution Time**
MD5	100 megabits/sec (5.12microseconds/512bit block)
SHA-1	48 megabits/sec (10.7 microseconds/512bit block)
SHA-256	24 megabits/sec (42.8 microseconds/1024bit block)
RSA-512 Signature	1.92 ms
RSA-512 Verification	0.13 ms
RSA-1024 Signature	10.29 ms
RSA-1024 Verification	0.30 ms
RSA-2048 Signature	64.13 ms
RSA-2048 Verification	0.89 ms
DSA-512 Signature	1.77 ms
DSA-1024 Signature	5.5 ms

If these authentication protocols were implemented on a less powerful wireless device, the signing and verification times would increase dramatically. Depending on the implementation, factors of 10-1000 times slower are possible.

Very large public key signatures may be attractive for use at the application layer, but are inappropriate for use at the data-link layer. For example, RSA-2048 is 256-bytes in length. It takes on the order of 0.6 seconds to sign on a fairly strong machine. The same signature on an eight-bit processor will take tens of seconds to accomplish. The whole IEEE 802.11 LLDT would take a few minutes to accomplish no matter the transceiver data rate. Using the IEEE 802.11 MAC approach, the channel would sit idle for a considerable amount of time while participating nodes signed and verified the messages of the LLDT. Combine this with the wireless channel contention issues, end-to-end throughput is reduced to the order of a few bits per second no matter how high the transceiver data rate. These extremely large public key signatures do not make sense for securing MANET lower-layer messages. We limit the signature size to 128 bytes, which corresponds to RSA-1024. We show that this may still be too large to be practical.

We have taken a black box approach to the signature and encryption algorithms. For various experiments we have taken a range of processing times suggested as appropriate by Table 3.2. No distinction between encryption and authentication is made; the processing times of both are lumped into a single delay. A range of signature sizes is also used. Because of a possible wide variance in hardware/software implementations, we have made no attempt to correlate signature size with processing time.

In our simulations, we have accounted for cryptographic overheads on each and every packet. All one-hop messages must be signed and verified, so the appropriate delays are accumulated every time a node generates, receives to forward, forwards, or receives a packet. In addition, we assume that a node's processor is capable of processing a single cryptographic primitive at a time, thus we have included queuing delays.

3.2 Modeling and Simulation Parameters Summary

An OPNET Modeler simulation was developed to measure the performance of ZRP and cryptographic impacts. Detailed models of the components of ZRP are used in the simulations and were provided by Cornell University [15]. Process models to represent the routing protocol, node movement, the application layer traffic generation, the media access control, the physical radio link, the wireless channel, and the overhead impacts of the applied cryptography were developed to support the simulations studies.

The simulations were performed on an 800MHz Windows PC. Simulations were limited to 49 node networks. High fidelity models of all relevant protocols simulated with OPNET Modeler are computationally intensive. One has the choice to either remove protocols, reduce the fidelity of the protocols to abstract out the essential details, or else keep the number of nodes small. We chose the latter in an attempt to be as faithful to the protocols as possible.

In our simulations we used a transmit radius of 300 meters, which corresponds to the IEEE 802.11 MAC protocol requirement of a less than one microsecond air propagation time [32]. The

fixed simulation values used in the studies presented in this report are defined in Table 3.3. Details of the OPNET process models used in the simulation are presented in Appendix A.

Table 3.3: Simulation parameters

Parameter	**Value**
Number of Nodes	49 nodes
Network Coverage Area	1,600 x 1,600 meters
Node Transmission Radius	300 meters
Node Beacon Period	0.33 second
Node IARP Update Period	1.0 second
Node Update Packet List Period	1.0 second
Node Discard Old Packets Time	1.0 second
Node Transceiver Data Rate	10 Mbps

4. Analysis and Simulations

In this section we present the results and conclusions of various analysis and simulation experiments that were conducted. The pertinent details of the experiments are given below. For the sake of clarity, we give an outline of the ordering of the subsections.

1. Link-Layer Data Transaction
2. End-to-End Data Throughput
3. Optimal Zone Radius
4. End-to-End Data Delay
5. Routing Table Convergence

4.1 Link-Layer Data Transaction

To prevent adversarial manipulations, the media access messages need to have security features placed on them. However, questions about performance arise. Each message in the media access transaction must be signed by the sender and verified by the intended receiver. Since any node able to receive an RTS or CTS is an intended receiver, all nodes in the area must verify a RTS and CTS exchange. The act of signing or verifying a RTS, CTS, DATA, and ACK must be done during the LLDT. However, the source node is able to sign the RTS and verify the ACK off line, so the LLDT must include six authentication procedures.

The 802.11 MAC standard calls for RTS, CTS, ACK and DATA header to be of size 40, 39, 39, and 47 bytes respectively. DATA sizes vary, up to a maximum of 1518 byte packets. Application of the digital signatures above will add from between 16 and 128 bytes to the packet size. With a transceiver data rate of r megabits per second, the time in milliseconds to transmit a K byte packet is:

$$T = 8*K/(1000*r)$$
$$= K/(125*r)$$

Table 4.1 provides examples of the time, in milliseconds, to transmit packets of the given number of bytes at the given data rates. The sample packet sizes given are: the smallest packet, the smallest packet with the smallest signature, the largest data size, the largest data size with header and the largest signature. Transmission times are linear in the number of bits transmitted and in the transmission data rate, so simple interpolation allows computation of other timing values.

Table 4.1: Time to transmit packets at the given data rates

Data Rate	**Transmit time for 39 bytes**	**Transmit time for a 55 byte packet**	**Transmit time for 1518 bytes**	**Transmit time for a 1693 byte packet**
1 megabit/sec	0.31ms	0.44ms	12.44ms	13.54ms
10 megabit/sec	0.031ms	0.044ms	1.244ms	1.354ms

To accurately measure the performance impacts of security we need to have complete network information including the actual communication protocols, the signature algorithm, transmission data rate, hardware and software capabilities, etc. In order to make general statements we make a few simplifying assumptions. We assume that the time to cryptographically process a received message is equal to the time for the sender to cryptographically prepare the message for transmission. We assume that this time is independent of the size of the message and denote it in milliseconds as AUTH. We assume that DIFS=0.128 milliseconds and SIFS=0.028 milliseconds.

Encryption and decryption times are roughly equal for most symmetric key encryption algorithms. Similarly, the statement that "signature and verification times are roughly equal" is valid for many signature algorithms. The most notable exception is RSA. The times that are given in Table 3.2 assume a public exponent of 17. More conservative implementations use a larger public exponent. This would close the gap between times to sign and verify. Further, since there is the same number of signatures as verifications in an LLDT, and they must be done sequentially, then one can think of AUTH as the average of the two times.

Let *R, C, D, H, A,* and *S* be the number of bytes in the RTS, CTS, DATA, HEADER, ACK, and Signature respectively. The total time, in milliseconds, that the channel is occupied for an LLDT is:

$$T = 6*AUTH+DIFS+3*SIFS+(R+C+D+H+A+4*S)/(125*r)$$

$$= 6*AUTH+0.212+(165+D+4*S)/(125*r)$$

One method to reduce the time T, and thus increase the available throughput of the network, would be to increase the data rate *r*. There is a point at which 6*AUTH +0.212 becomes the dominant factor in T and further increases in *r* will not significantly affect the total transaction

time. Similarly, reducing AUTH will not significantly reduce T when 0.212+(165+*D*+4**S*)/(125**r*) becomes the dominant factor. For that matter, the inter frame spacing times of 0.212 ms give a lower bound on the total transaction time.

Another issue is that of idle channel time. Nodes not directly participating in an LLDT set their NAV, which must include AUTH times. These nodes will sit idle during the transaction. If AUTH is large in comparison to the other factors, the channel will sit idle while LLDT participants are completing their cryptographic processing. This time may well be used for other smaller transmissions. However, interleaving communications in this fashion would require a significant rewrite of the IEEE 802.11 MAC approach.

The ratio of the time to transmit DATA and the time to complete a LLDT is the percentage of the transmission data rate that is available to a node. We denote this percentage as P, and it is given by:

$$P=D/(r*(125*AUTH+26.5)+165+D+4*S)$$

Because the processing power can vary from device to device even the values of AUTH and *S* can be treated as independent variables. In this analysis we set the data rate to either one or ten megabits/second, we set the signature size to 16 and 128 bytes, and examine 64 and 1518 byte DATA packets.

Figure 4.1 and

Figure 4.2 show P with AUTH ranging from 0.001 to 100 milliseconds (log scale). Any AUTH value that is in the one-second range will force multihop messages to take multiple seconds to reach its destination.

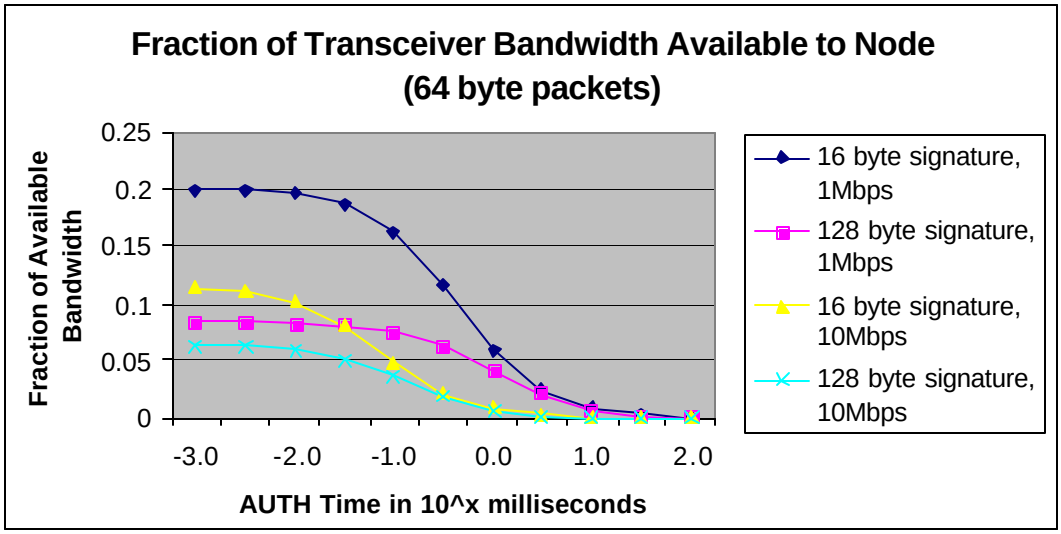

Figure 4.1: Fraction of transceiver bandwidth available to node transmitting 64 byte packets

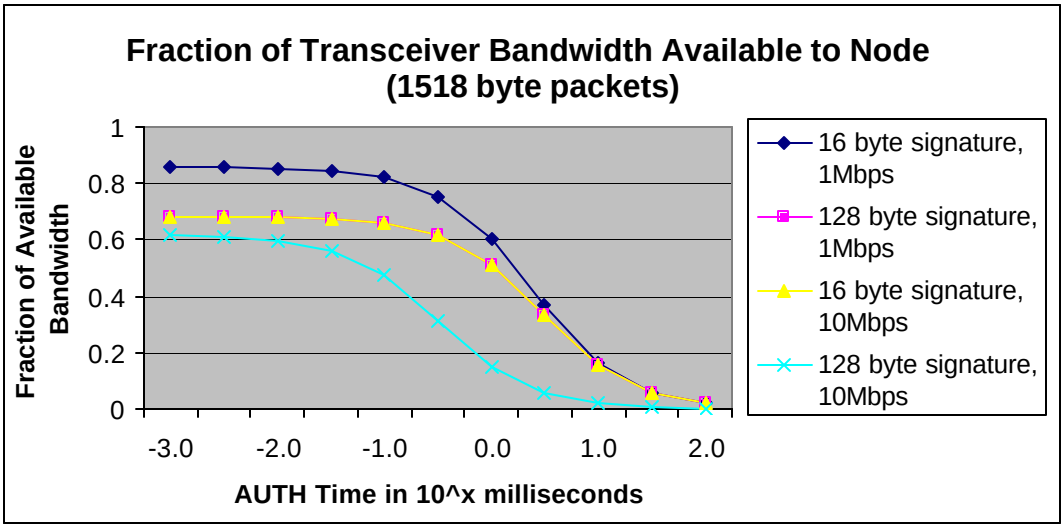

Figure 4.2: Fraction of transceiver bandwidth available to node transmitting 1518 byte packets

The results of this subsection will be combined with information from subsequent sections to give an overall picture of the impacts of security delays on network throughput.

4.2 End-to-End Data Throughput

The end-to-end throughput in a multihop network is a function of the network diameter. Roughly speaking, it is the one-hop throughput divided by average number of hops that a message must traverse. Assuming a roughly uniform nodal density and random message destination, the average path length is on the order of the square root of the number of nodes in the network.

In an actual wireless network there are generally two radii of importance. The transmission radius is the distance for which a node's transmission has a high probability of being accurately received. The interference radius is the distance for which a node's transmission can interfere with another transmission. The second is usually much greater than the first. In [30] it is argued that under the assumption that the interference radius is slightly more than twice the transmission radius that there will be a factor of on the order of *twelve* decreases in the one-hop throughput. The IEEE 802.11 MAC standard describes this ratio of transmission to interference radius.

The factor of twelve decrease in the one-hop throughput described in [30] assumes that the communicating nodes are set up in a regular lattice and that no node has more than four neighbors. Average number of nodes in the one-hop neighborhood also plays a role in the one-hop throughput, since all nodes must share the channel with their closest neighbors. Given a large network with a fixed number of nodes and a uniform density, the number of one-hop neighbors increases quadratically with the transmit radius. On the other hand, the number of hops needed to traverse the network decreases linearly with the transmit radius.

Position and nodal movement also play roles in the capacity of a network. In a simulated network where nodes have random initial position and move in random directions the nodal distribution is never uniform. At times, portions of the network will be sparsely populated while other areas will be densely populated. The sparse areas may contain nodes that have no one-hop neighbors and hence cannot communicate with any other node. For these nodes, their one-hop throughput is zero. The random placement of low-density areas tends to increase the average path length as paths must navigate around the holes rather than pass directly through them. In the dense areas there is much greater contention for the channel. Thus the end-to-end throughput for any particular node is less than what one would predict using a uniform distribution model. It is difficult to precisely predict the impact of all of these issues on the end-to-end throughput.

If the network density is low enough, network partitions are common in a random movement model. Even if the network is reasonably dense, network partitions are common enough to cause simulation problems. As nodes randomly choose packet destinations in a different connected component, the dropped packet rate raises accordingly. This dramatically affects the overall end-to-end throughput that any node is able to realize.

Combining the one twelfth factor from [30] and the average path length the end-to-end throughput may be estimated by dividing the transceiver data rate by at least 12*root(N). In addition, application-layer messages must share the remaining channel with lower-layer control messaging.

4.3 Optimal Zone Radius

We simulated ZRP over various zone radiuses, which were incremented from one to five. Technically, ZRP with a zone radius of zero is equivalent to a true “flood” search algorithm. However, the ZRP modeling blocks to which we had access do not allow for a zero radius. One of the reasons for this is that a zone radius of zero is fundamentally different than a non-zero radius and would require on-the-fly disabling the NDP protocols, link-state update protocols, etc. A zone radius of one is practically equivalent to a “flood” search. However, there is a slight savings over a flood on the occasion that an upper-layer message is destined for a node in the one-hop neighborhood, thus no flood is initiated. Also during a route discovery, when a node sees the destination node in its neighbor list, it returns a path rather than continuing with a path search. With a zone radius of one, the one-hop neighbors are the peripheral bordercast nodes. This results in no bordercast savings during route discovery.

In our simulations with a 49-node network, a zone radius of five is close to a complete proactive link-state routing. If the nodes are placed regularly in our square grid, then the five-hop intrazone of a randomly placed node will likely contain all network nodes. In the few cases where a node has a non-empty interzone, there is an overwhelming probability that the node has a peripheral node with a path to the destination.

As the zone radius increases, more proactive routing traffic is generated. Each node forwards its link-state information to every node in its intrazone. As the radius grows the number of nodes in the interzone grows with the square of the zone radius. Thus, control traffic should increase on

O(radius^2*)*. Figure 4.3 illustrates the IARP proactive region overhead traffic as the zone radius increases.

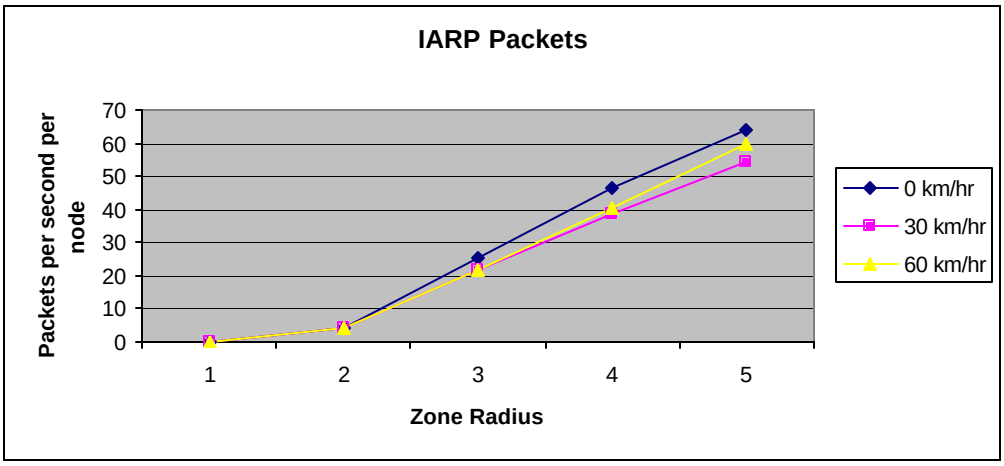

Figure 4.3: IARP control packets increase as the zone radius is increased

One can observe two factors about Figure 4.3. The first is that the IARP traffic seems to be independent of the node velocity. This is expected because the control messages are generated on a periodic rate that is independent of velocity. The second is that the expected quadratic nature of the curves appears to not hold. This fact can be attributed to the small size of the network in relationship to the larger zone radii. The large intrazones simply run out of new nodes to include. However, the values given provide a lower bound on the number of overhead messages that the network must support.

As the zone radius increases the amount of IERP traffic is reduced. Large radii require less frequent initiation of the route discovery protocol since more proactive routes have been determined by IARP. In addition large zone radii reduce the amount of traffic generated during a route discovery because of use of peripheral nodes and bordercasting efficiencies.

Figure 4.4 illustrates the IERP reactive region overhead traffic as the zone radius increases. For this figure the application-layer traffic generation was set to the medium level as defined in Section 3.1.

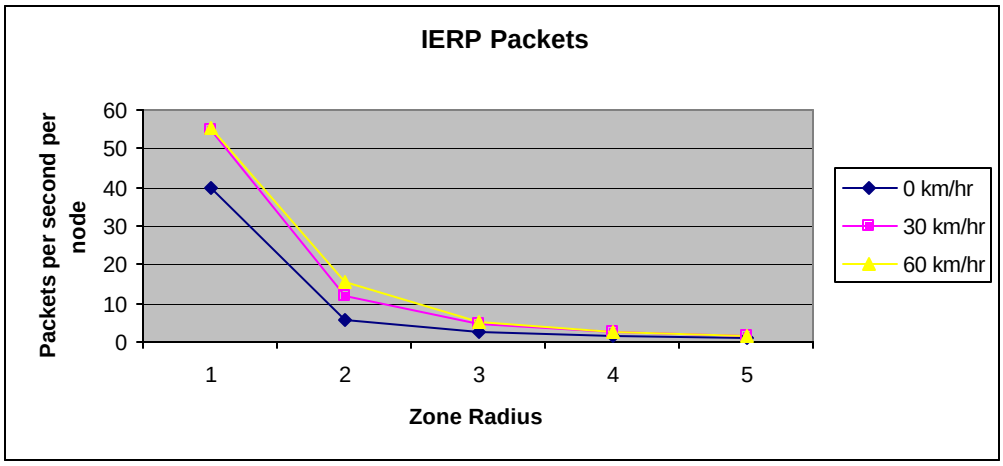

Figure 4.4: IERP control packets decrease as the zone radius is increased

As one would expect, the IERP traffic does depend on node velocity. When network nodes have zero velocity, routes are discovered and stored in a routing table and that is valid for all time. In contrast, if the transmitting, receiving, and/or intermediate hop nodes have velocity then discovered routes eventually become invalid and must be replaced.

IARP and IERP control traffic components are summed and plotted as a function of the zone radius in Figure 4.5.

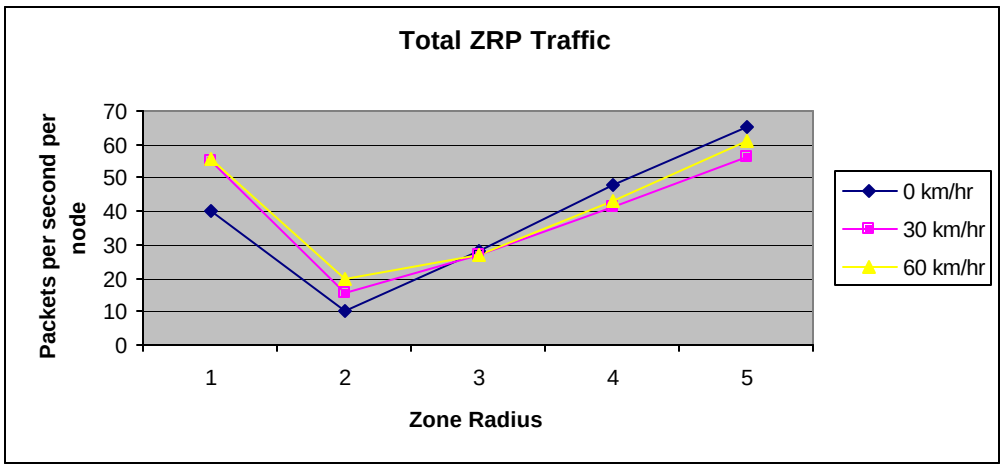

Figure 4.5: Total ZRP control packets as a function of the zone radius

Due to the increasing and decreasing nature of the IARP and IERP traffic respectively, the total control traffic curves are convex. The minimum represents the optimal zone radius for the given velocity.

With ZRP, every network has an optimal zone radius. For some networks this optimal zone radius may be two for a purely reactive approach or for some other networks the radius may be

infinite, thus a purely proactive approach. For most networks, a zone radius somewhere in between is best.

In our simulations, the optimal zone radius is two for all velocities. A zone radius of two takes advantage of the bordercasting techniques and is more efficient than a zone radius of one, which is essentially a flood search. It should also be noted that increasing node speed yields greater numbers of control messages. Rediscovery of routes occurs more frequently with higher node speed.

Cryptographic Overhead:

In the current implementation of ZRP, the link-state updates are unicast to each node in sender's one-hop neighborhood and are accompanied by the RTS/CTS exchange. All IERP messages are sent to specific nodes and are also accompanied by an RTS/CTS exchange. With *n* and T as the number of control messages a node sends per second and the time to complete an LLDT respectively, then *n**T is the percentage of a second that the channel is busy due to a single node's necessary control messaging.

In Section 4.2 it is shown that in a network with a somewhat minimal and regular density, no node may utilize more than one twelfth of the available transceiver data rate. The factor of one twelfth is not absolute and may decrease further, if the node density increases. A node that uses its entire portion of the available bandwidth to allow for control messaging does not have the ability to send or forward application layer data. Table 4.2 combines some of the results from Section 4.1 with the results in Figure 4.5 to obtain a view of the remaining throughput when overhead is accounted for. For simplicity we assume that each control packet is 64 bytes in length. Note: that the IERP route request packets may be larger.

Table 4.2: Final throughput available after cryptographic processing time, channel contention, and path length are considered (considers a 10 MHz channel)

Cryptographic Processing Time	**Maximum Number of LLDT Messages per Second** (*M*)	**Maximum Number of Messages with Channel Contention** (*N=M/12*)	**Available Number of Messages for Application after Accounting for Control Messages** (*P=N-20*)	**Available Number of Messages for Application after Accounting for Path Length** (*Q=P/5*)	**Remaining Throughput** (*R=512Q*)
0.0us	2214	184.5	164.5	32.9	16.8K
10us	1974	164.5	144.5	28.9	14.8K
100us	956	79.6	59.6	11.9	6.1K
1ms	155	12.9	none	none	Non-functional
10ms	17	1.4	none	none	Non-functional

Figure 4.5 was created with application-layer data being transmitted in the few kilobit per second range. This is a small percentage of the available transceiver data rate and should not be considered a difficult application-layer load. For application-layer messages, a node may use only the percentage of the one-twelfth channel that is left over after the control messaging. To obtain an estimate of the end-to-end throughput one may divide the one-hop throughput by the average path length to account for messaging forwarding requirements. In Table 4.2 we set the path length to five.

With the minimal number of about 20 control messages per second and the channel contention factor of 12, the LLDT cannot take more than 1/(20*12)=4.2 milliseconds or else the node will expend more than its share of the channel just to process control data. Since there are six applications of cryptographic delay in an LLDT, the cryptographic delay must be less than 4.2/6=0.7ms. The 0.7ms is an upper bound on the cryptographic delay, and does not account for the need for inter frame spacing nor does it account for the transmission time.

It apparent from Figure 4.5 a non-optimal selection of the zone radius leads to significant increase in the amount of control traffic. For instance, a zone radius of four requires more than 40 control packets per second. In this case, each LLDT would need to take less than 2.1 milliseconds, which implies an upper bound on the cryptographic delay of 0.35 ms.

Accounting for channel contention, routing messages, path length, but without security delays, Table 4.2 shows that the end-to-end throughput is only a small fraction of the transceiver data rate. Even when the cryptographic delays are in the few microsecond range, the end-to-end throughput drops 12 percent. However, when the cryptographic delays are on the order of a millisecond, the network essentially becomes none-functional. These results show that public key cryptographic algorithms are not appropriate for securing lower-layer messages.

Application-layer Loads:

The amount of IERP traffic is an increasing function of the amount of application-layer traffic. Thus, one would expect that the optimal zone radius is also a function of the amount of application-layer traffic, however the optimal radius is an integer and may not vary greatly with small changes in the traffic load. Figure 4.6 compares control traffic with 30-km/hr node speed with three different application-layer loads.

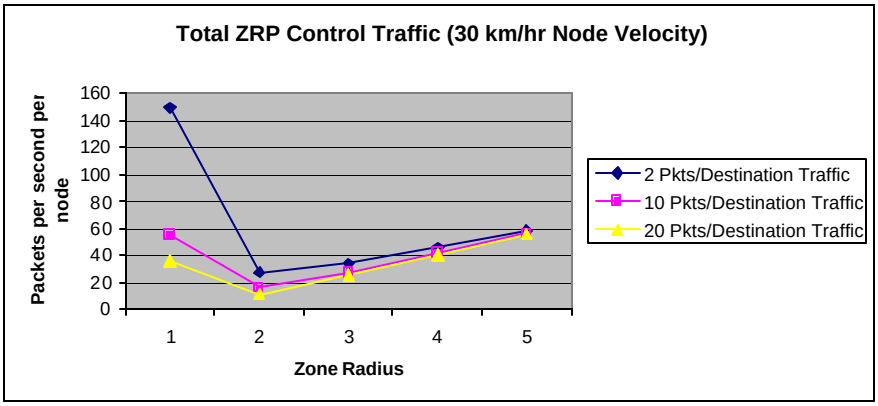

Figure 4.6: Optimal zone radius dependence on application-layer traffic

For a zone radius of one, ZRP is essentially an on-demand routing protocol. When the application-layer data is directed to a large number of different destinations many route requests must be initiated, resulting in a significant increase in control traffic. On the other hand, when the zone radius is large more destinations lay within the intrazone so fewer route requests are needed. When a route request is initiated it benefits from bordercasting and generates less route-discovery control traffic. The generated route discovery traffic is also a smaller portion of the overall control traffic. Thus, when the zone radius is large, the amount of control traffic does not vary greatly with variance in the application-layer data destinations.

Initial Node Placement:

Initial placement of network nodes does affect the amount of lower-layer traffic. We examine two different placement patterns. Figure 4.7 shows an initial placement of nodes in a regular lattice and where nodes are randomly placed in the simulation grid.

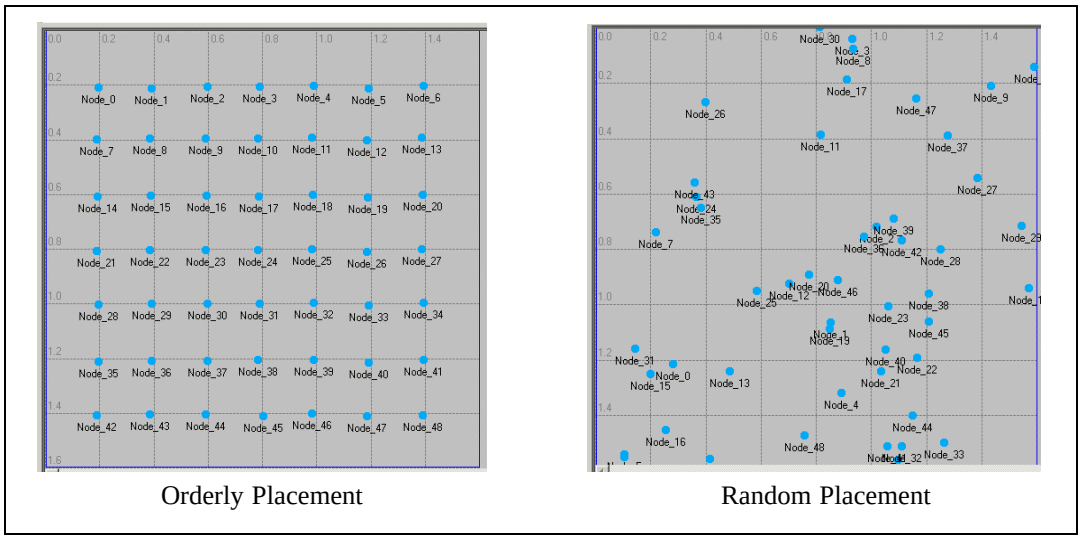

Figure 4.7: Simulations are done with both orderly node placement and random node placement

If a network is to maintain full connectivity, then each node must be connected to at least one other node that has further connectivity into the network. If a node loses contact with all other nodes or a group of nodes loses contact with another group of nodes, the network has *partitioned* and some messages will not reach their destination. Partitioned subnetworks cannot communicate with each other. With the average nodal density used in these simulations network partitions are common as seen in Figure 4.7.

The amount of IARP and IERP traffic is a function of the largest connected component in the network. For routing purposes, the nodes in any particular partitioned subnetwork will view the world as consisting of that subnetwork. The total number of lower-layer messages generated by the network is the sum of the number of lower-layer messages generated by each of the connected components. Increasing the zone radius past the diameter of any particular component will not result in an increase in the amount of IARP traffic within that component. Increasing the network zone radius results in an increase in the overall IARP traffic as long as there is a connected component with a diameter large enough to use the larger radius. As the radius increases there are fewer components that are able to take advantage of that change, and thus fewer nodes contribute to an increase in the IARP traffic.

IERP traffic is generated in direct response to application-layer traffic generation. Each route request generates a number of IERP messages equal to some fraction of the number of nodes in the connected component in which the initiating node lies. Again, the total is dominated by the largest connected component in the network.

In Figure 4.8 we show the number of control messages per second with the two different initial positions as shown in Figure 4.7. Nodes are given zero velocity because a network with random movement and an orderly initial position degenerates into a network with random placement.

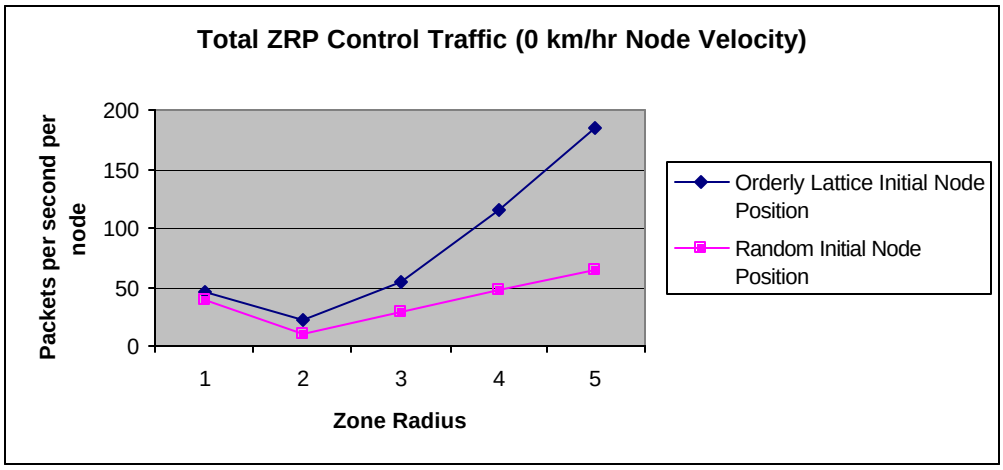

Figure 4.8: Total ZRP control traffic with both an orderly lattice initial node position and a random initial node position

There are no network partitions in the orderly placement scenario. This allows for the full effects of the control traffic's quadratic dependence on the zone radius to be seen. This quadratic dependence on the zone radius results in a significant number of control messages for large zone radii. Thus the selection of the zone radius is more critical when the network is fully connected than when it is partitioned.

4.4 End-to-End Data Delay

An important measure of any routing protocol is the total time to propagate an application-layer packet from a source node to destination node. This delay is defined as the amount of time from packet creation to the time it is received at the destination application-layer. In our simulations, delays from route discovery, data transmission, cryptographic processing and queuing are accumulated. However, we do not include the delays associated with media access control. If a node's destination is listed in its routing table, the query to the routing protocol is immediately answered and no time is assumed to expire for this step and no delay is accumulated for this action. On the other hand, if no route is listed in the table, a route request is initiated. In this case, the application-layer packet is queued and a route-request packet is generated and transmitted.

Figure 4.9 shows two nodes communicating over a multihop path. In this figure, Node 2 can communicate directly with Node 9, but cannot communicate directly with Node 16. Thus, for Node 2 to communicate with Node 44 a number of nodes must participate in relaying the message to Node 44. Figure 4.10 plots the time for Node 2 to send a packet to Node 44 with zone radii of two and six respectively.

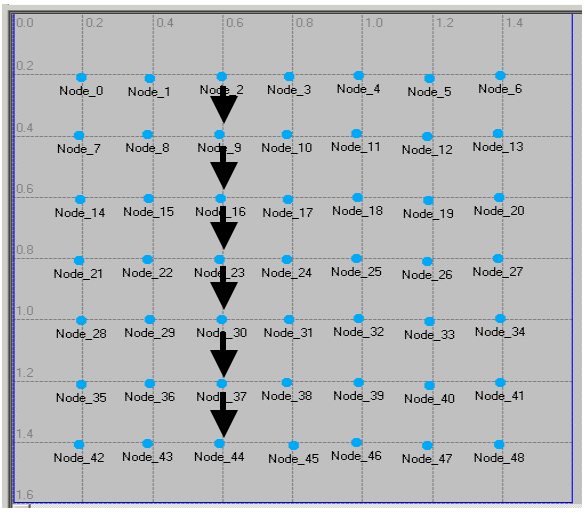

Figure 4.9: Node 2 transmitting data packets to Node 44

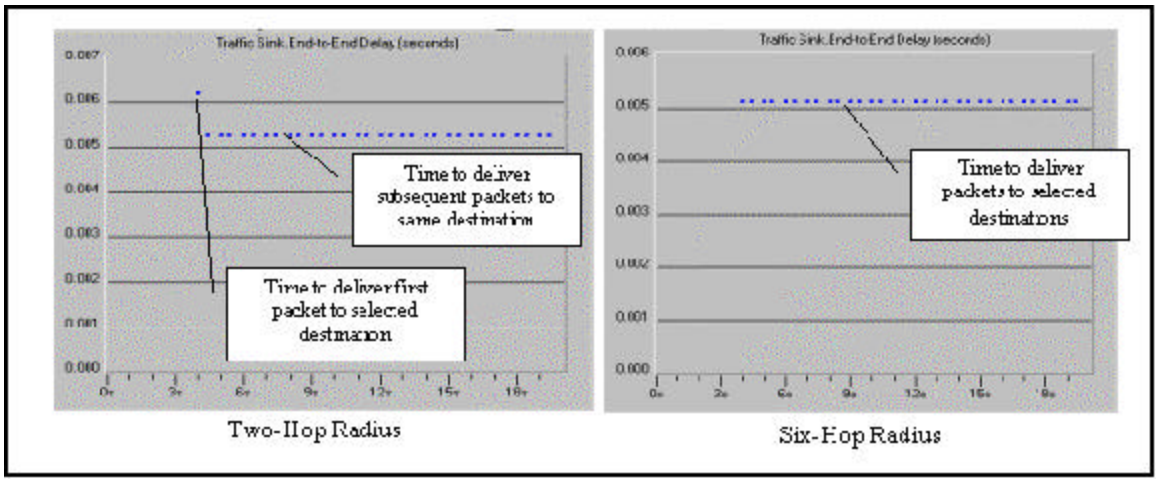

Figure 4.10: Time to deliver a number of application-layer packets from Node 2 to Node 44 with a two-hop radius and six-hop radius

With a zone radius of two, the first packet sent to node 44 takes longer than the subsequent packets. This is due to the route discovery process. After the route has been discovered the end-to-end delay is essentially the same for both of the zone radii considered. These results are based on a packet size of 1024 bytes and a transceiver data rate of 10Mbps. Only node 2 is generating application layer messages.

Below is a set of curves that show average end-to-end delay for various zone radii and traffic loads. Each figure includes the average end-to-end delay with and without the overhead of cryptographic features. The cryptographic processing delay is one millisecond. The simulations for these plots have all nodes generating traffic. Nodes receive packets from multiple destinations, they forward theses message as well as generate their own. Packets inevitably will be queued since the node can transmit a single message at a time.

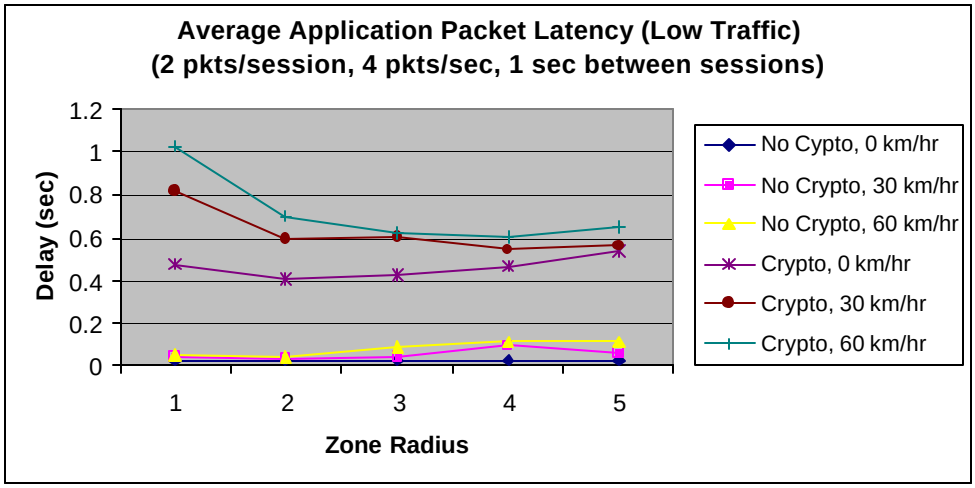

Figure 4.11: Impacts of link-layer cryptography on low application-layer traffic

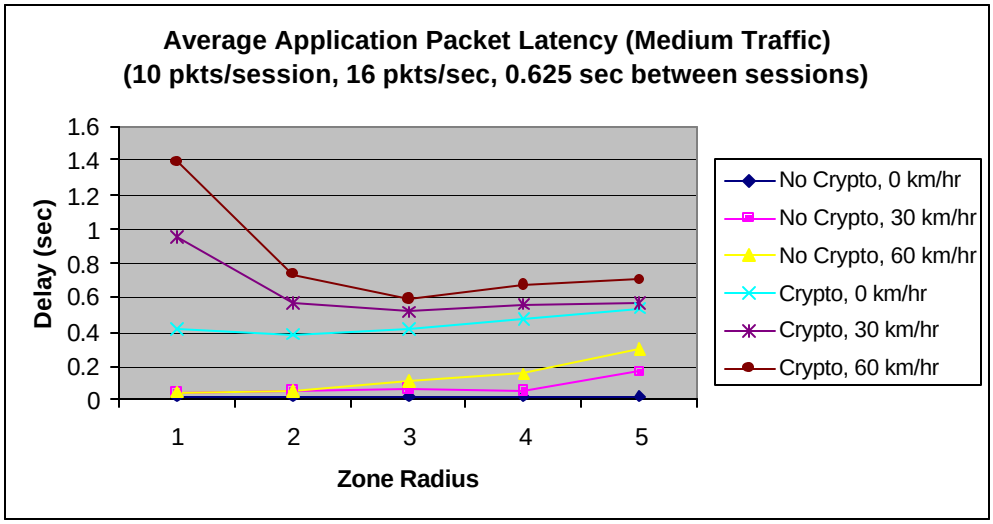

Figure 4.12: Impacts of link-layer cryptography on medium application-layer traffic

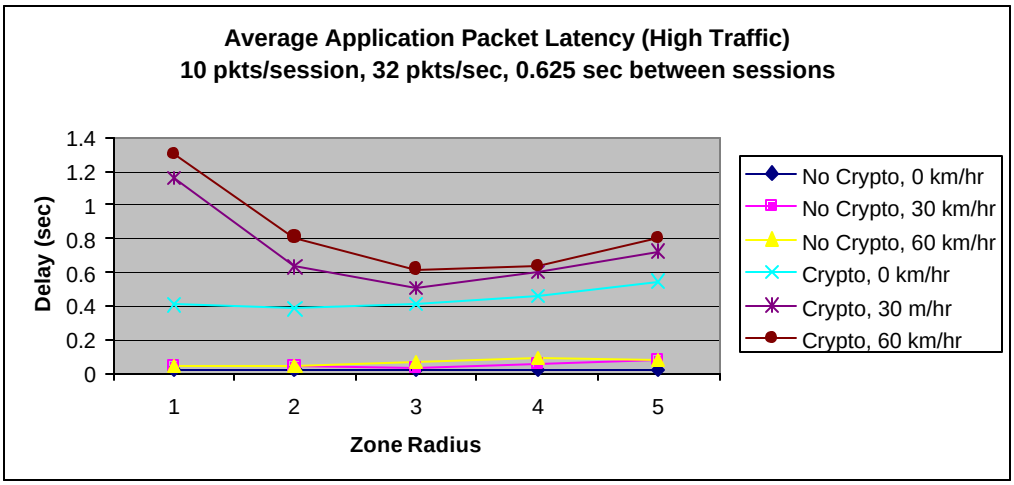

Figure 4.13: Impacts of link-layer cryptography on high application-layer traffic

It appears that the average end-to-end delay follows fairly closely the shape of the curves in Figure 4.5. One would expect there to be a correlation in the amount of application-layer delay and the amount of lower-layer control traffic. Application-layer messages must compete with the control traffic for channel resources. However, the delays due to route discovery in the small zone radii are washed out because there are 10 packets per session. The initial packet suffers significant delay and subsequent packets to that destination take advantage of the discovered route and do not suffer that delay.

4.5 Routing Table Convergence

ZRP depends on IARP to proactively build a routing table based on the link-state of neighboring nodes. At network start-up, each node builds from scratch a routing table. With each new link-state update that it receives a node refines its view of its intrazone. If at some point the routing table reflects the actual topology of the intrazone, the table is considered to have converged. The time that it takes for the routing table to converge is dependent on the timer setting of the IARP link-state update period and the NDP beacon period as well as node speed. In reality, node movement and the periodic update of topological information means that there will be times that the routing table is not 100 percent correct. As speed increases, with the beacon rates constant, one would expect that there is a point where the routing table will consistently have an incorrect view of the zone. On the other hand, if the speed is zero, once the table converges it will always be correct.

Conceivably there are various metrics that one could apply to measure the quality of the routing table as it is being built. At any given moment, one may compare every possible path in the not quite complete route-table with every possible path in the true intrazone. This is somewhat time consuming. We selected a much simpler method of measuring the quality of the route table as it evolves. As the routing table is built, it grows in size and at some point the size remains roughly constant. At this point the routing table contains all the pertinent information about the zone and rarely changes except to compensate for link-state changes. For our simulations, our measure of routing table convergence is a ratio of the size of the table to the size of the steady state table.

Figure 4.14 illustrates the convergence of the IARP routing table as time progresses. The zone radius is six, the initial node placement is a regular lattice, and various cryptographic delays are applied. Since the convergence time is fairly small, node movement is minimal during that time, so only the results of zero node speed are presented.

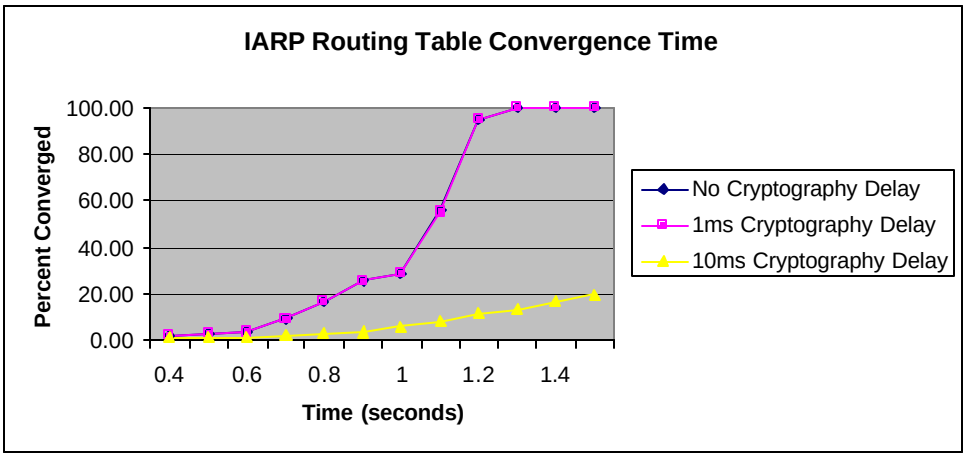

Figure 4.14: Time for protocol to complete proactive region routing table
(Note: *No* and *1ms* Delay curves are virtually identical)

The IARP update period will dramatically affect the rate at which the routing table converges. Each node generates its own link-state messages at a rate of one per IARP update period starting

at a random time during the first period from the startup of the network. By the end of that first period all nodes have sent their initial link state information. The nodes that sent their information early send an incomplete view of their intrazone, while those who send later send a more complete view of their zone. In our simulations, the update period is set to one second. It is at this point that there is a significant increase in the quality of the routing table for all but the case where the cryptographic overhead is highest. When the delay associated with cryptography is a sizeable fraction of the IARP update period its impact is felt during route table generation.

Inspection of a routing table that has attained 30 percent of its steady state size reveals that routes to most nodes in the intrazone are contained in the table, however some of the most efficient paths are missing. Even when the table is in a state of growth and not 100 percent converged, the network is still functional.

5. Future Work

There are significant performance impacts on MANETs that employ link-layer security features. For simulation purposes our research required the selection and use of a specific MANET routing protocol and media access protocol. During our work, several questions arose as to how to modify the protocols we studied to make them more efficient so that they would not be so sensitive to cryptographic delays. Our results indicate that further protocol efficiency improvements are necessary before security features can be easily added to MANET lower-layer protocol. In this section we present a list of further research areas.

- Cornell's ZRP uses a *reliable* broadcast that sends broadcast messages to each neighbor in transmission range individually. The simulation determines the neighbors by a physical distance calculation. Link-state updates that are distributed with a true broadcast will reduce the amount of IARP message traffic thus improving routing protocol performance. However, with a true broadcast approach, there will be an increased probability of dropped packets because of packet collisions due to the hidden terminal problem. Collisions may occur frequently since there may be a large number of link-state updates. Further, key management techniques become an issue in the broadcast mode. A node signs a message and broadcasts it, and each receiving node must verify the message. This verification can only take place if the verifying nodes have the keys necessary to do so. Public key protocols are ideal for such a scenario, but we have shown that they are too computationally intensive to be of use for securing lower-layer messages. Network-wide symmetric keys are an issue if a node is compromised. Neighborhood-wide symmetric keys are a possible option. A different MAC approach that reduces the probability and/or impact of collisions during the broadcast of lower-layer messages is another area of research.

- Our implementation of ZRP uses a periodic IARP beacon and a periodic link-state update that is independent of node velocity. In each period, IARP transmits the current link-state even if it has not changed. In turn, the neighbor nodes forward the link-state to nodes within the intrazone. Unless a node's link state changes, sending out its complete link-state information is a waste of resources. Hello beacons may be modified to contain a flag indicating that there has been no change in the link-state. Once the network has

converged, it may be more efficient to propagate information describing the change in link-state rather than propagate the entire table. If the network is relatively static, it may be possible to have the link-state change information tag along with upper-layer messages rather than be sent out individually. If the network is more dynamic, then it may be better to revert to the passing of link-state information.

- The selected simulation scenarios used in our studies all suffered some degree of network partitioning when the nodes moved in random directions. One can easily consider applications where the node density would be less than the scenario used in this report and thus would suffer from a greater number of partitions. Each subnetwork requires fewer control messages per unit of time, but the nodes in the subnetwork have a smaller set of possible message destinations. In a high consequence network, it may be that certain nodes are designated to put themselves into a position that will keep the network from partitioning. More reasonable movement patterns need to be studied and incorporated into the simulations. However, it may be quite resource expensive to guarantee a network will not partition. Thus, methods to cache messages for later delivery are necessary for efficient MANET operation.

- Because node density plays a significant role in the efficiency of the network, methods of adaptive power control should be folded into the routing algorithm. Nodes that have many neighbors can reduce their transmission radius and maintain communication with the network. This may increase the number of hops necessary to send a message, but will reduce the amount of interference with neighboring nodes. This results in more nodes being able to communicate at any given time. However, nodes equipped with adaptive power control will have a higher incidence of asymmetric links. Nodes with unequal power transmissions will also subvert, to some extent, the CSMA/CA media access technologies.

6. Conclusions

This report began with an introduction and brief explanation of the network and data-link layers used in a typical wireless MANET. Protocols implemented at these layers generate control traffic to support their operation. This control traffic is critical to network operation and must be secured. Cryptographic features can be used to secure the control messages, however, such use comes at a cost in performance. This research identifies specific costs in performance when cryptographic features are used to secure lower-layer messages.

To identify specific performance costs we selected widely known routing and data-link layer protocols. ZRP was selected for the routing protocol and IEEE 802.11 standard was selected for the data-link layer protocol.

An analysis is presented of the message exchange that takes place when the IEEE 802.11 MAC accesses the wireless medium. Cryptographic features were applied to each of the messages involved in the data-link layer exchange. The time that it takes to complete a data-link layer RTS/CTS/DATA/ACK exchange is highly sensitive to the time required for cryptographic

processing. In the extreme, available channel bandwidth remains idle while nodes process cryptographic algorithms.

To quantify the performance impact that lower-layer cryptographic features have on overall network performance, we considered the data throughput available when MANET overheads are accounted for. We show throughput results when various cryptographic processing delays are applied. We also consider channel contention, multihop impacts, and control overhead. We observe a drastic reduction in end-to-end throughput when these overheads are considered.

Since application-layer traffic and node mobility have a large impact on routing protocol performance, we simulated protocol operation under various parameter settings to determine optimal zone radii to minimize the amount of control traffic. Changes in application-layer traffic and mobility cause this optimal zone radius to vary slightly. Varying these parameters in a larger network will lead to a greater variation in the optimal zone radius. Our results show that small deviations from the optimal zone radius leads to a large increase in control traffic.

End-to-end delay of application-layer messages is found to increase when cryptographic features are used in MANETs. Messages directed to the interzone will suffer increased end-to-end delay since control messages are delayed at each hop while their cryptographic features are processed. In this case the initial message suffers discovery delays and subsequent messages to the same destination take advantage of the discovered route and do not suffer the discovery delay. Messages directed to proactive route regions also suffer from delays since the channel is occupied longer and thus additional time is required to access the channel.

Application-layer end-to-end delay is also impacted by the time it takes to generate a useful routing table. Cryptographic processing overheads slow down the generation and maintenance of each node's routing table. Our simulation studies indicate that until the processing delays are on the order of the link-state update period, the delays do not significantly impact table convergence. However, when the delays become large enough, routing table generation is slowed down and network operation will depend on potentially stale routing tables, resulting in inefficient operation.

Some method of security must be applied at the data-link layer to protect lower-layer control messages to enhance system security. Our study shows that applying cryptographic features to the control messages in many cases introduce unacceptable network performance. Our findings indicate MANET operation depends on a number of highly sensitive attributes. Non-optimal network design will inevitably lead to drastic decreases in network performance. Blind application of cryptographic features to secure lower-layer MANET protocols could easily destroy network functionality.

7. References

[1] O. Berg, T. Berg, S. Haavik, J. Hjelmstad, and R. Skaug. *Spread Spectrum in Mobile Communications.* IEEE, 1998.

[2] N. Borisov, I. Goldberg, D. Wagner. *Intercepting Mobile Communications: The Insecurity of 802.11,* Proceedings of MOBICOM2001, 2001.

[3] Breezecom, *IEEE 802.11 Technical Tutorial.* http://www.breezecom.com.

[4] J. Broch, D. Johnson and D. Maltz. *Dynamic Source Routing (DSR).* Internet Draft, draft-ietf-manet-dsr-03.txt, October 22, 1999.

[5] J. Broch, D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva. *A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols*. Proceedings of the Fourth Annual ACM/IEEE Conference on Mobile Computing and Networking, Mobicom, 1998.

[6] W. Chen, N. Jain, and S. Singh. *ANMP: Ad Hoc Network Management Protocol,* IEEE Journal on Selected Areas in Communications, August, 1999.

[7] M. S. Corson and V. Park. *Temporally Ordered Routing Algorithm (TORA).* Internet Draft, draft-ietf-manet-tora-spec-02.txt, October 22, 1999.

[8] M. S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, and A. Qayyum. *An Internet MANET Encapsulation Protocol (IMEP) Specificaion*. Internet Draft, draft-ietf-manet-imep-spec-02.txt, August 21, 1999.

[9] M. S. Corson and J. Macker, *Mobile Ad Hoc Networking: Routing Protocol Performance Issues and Evaluation Considerations*. Internet Draft, January 1999.

[10] B. Dahill, B. N. Levine, E. Royer, C. Shields. *A Secure Routing Protocol for Ad Hoc Networks.* Technical Report UM-CS-2001-037, Electrical Engineering and Computer Science, University of Michigan, August 2001.

[11] W. Dai. *Crypto++ 4.0 Benchmarks.* http://www.eskimo.com/~weidai/.

[12] J. J. Garcia-Luna-Aceves, M. Spohn, and D. Beyer. *Source Tree Adaptive Routing (STAR) Protocol.* Internet Draft, draft-ietf-manet-star-00.txt, October 22, 1999.

[13] Z. J. Haas, J. Deng, B. Liang, P. Papadimitratos, S. Sajama. *Wireless Ad Hoc Networks.* Cornell University Report. December, 2001.

[14] Z. J. Haas and M. R. Pearlman. *Providing Ad Hoc Connectivity With the Reconfgurable Wireless Networks.* In Charles Perkins, editor, *Ad Hoc Networks.* Addison Wesley Longman, 2000.

[15] Z. J. Haas and M. R. Pearlman. *The Zone Routing Protocol (ZRP) for Ad Hoc Networks.* draft-ietf-manet-zone-zrp-03.txt, March 2000.

[16] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. *Adaptive Protocols for Information Dissemination in Wireless Sensor Networks.* Fifth ACM/IEEE MOBICOM Conference, Seattle, WA, August 1999.

[17] Y. Hu, D. B. Johnson, A. Perrig. *Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks.* Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking, (MobiCom 2002).

[18] Y. Hu, D. B. Johnson, A. Perrig. *SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks.* Fourth IEEE Workshop on Mobile Computing Systems and Applications \mbox\rm (WMCSA '02), 2002.

[19] S. Jacobs and M. S. Corson. *MANET Authentication Architecture, Internet Draft.* Internet Draft, draft-jacobs-imep-auth-arch-01.txt, March 1999.

[20] P. Jacquet, P. Muhlethaler, and A. Qayyum. *Optimized Link State Routing (OLSR) Protocol.* Internet Draft, draft-ietf-manet-olsr-01.txt, February 7, 2000.

[21] P. Jacquet and L. Viennot. *Overhead In Mobile Ad Hoc Network Protocols.* INRIA Report, June 2000.

[22] S. Jiang, N. Vaidya, and W. Zhao. *Preventing Traffic Analysis in Packet Radio Networks.* DARPA Research.

[23] S. Jiang, N. Vaidya, and W. Zhao. *Power-Aware Traffic Cover Mode to Prevent Traffic Analysis in Wireless Ad Hoc Networks.* IEEE INFOCOM 2001.

[24] V. Karpijoki. *Signaling And Routing Security In Mobile And Ad Hoc Networks.* Report. Helsinki University of Technology, May 2000.

[25] S. Kent and R. Atkinson. *IP Authentication Header.* RFC 2402, November 1998.

[26] S. Kent and R. Atkinson. *IP Encapsulating Security Protocol (ESP).* RFC 2406, November 1998.

[27] S. Kent and R. Atkinson. *Security Architecture for the Internet Protocol.* RFC 2401, November 1998.

[28] V. Lakshminarayan. *NETWARS Standards Architecture and Implementation Issues*. NETWARS Standards Working Group. 1998.

[29] S. M. Lawandowski, D. J. Van Hook, G. C. O'Leary, J. W. Haines, and L. M. Rossey, *SARA: Survivable Autonomic Response Architecture.* DARPA Report.

[30] J. Li, C. Blake, D. De Couto, H. Lee, R. Morris. *Capacity of Ad Hoc Wireless Networks,* The 7th annual International Conference on Mobile Computing and Networking, Rome, Italy, July 2001.

[31] J. Lundberg. *Routing Security in Ad Hoc Networks.* Tik-110.501 Seminar on Network Security, Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, 2000.

[32] OPNET Technologies. *OPNET Users Manual.* doc@opnet.com.

[33] P. Papadimitratos and Z. J. Haas. *Secure Routing for Mobile Ad Hoc Networks.* In SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), January 2002.

[34] M. R. Pearlman and Z. J. Haas. *Determining the Optimal Configuration for the Zone Routing Protocol*. IEEE Journal on Selected Areas in Communications, August 1999.

[35] M. R. Pearlman and Z. J. Haas. *The Performance of Query Control Schemes for the Zone Routing Protocol*. AFRL Report.

[36] M. R. Pearlman. Email Conversation. September 26, 2001.

[37] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesley, 2001.

[38] C. E. Perkins, E. M. Royer, and S. R. Das. *Ad Hoc On-Demand Distance Vector (AODV) Routing.* Internet Draft, draft-ietf-manet-aodv-04.txt, October 22, 1999.

[39] C. E. Perkins and P. Bhagwat. *Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing For Mobile Computers*. In Proc. SIGCOMM'94, pages 234–244, August 1994.

[40] C. E. Perkins and E. M. Royer. *Ad Hoc On-Demand Distance Vector Routing.* In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pages 90-100, February 1999. New Orleans, LA.

[41] M. Torgerson and B. Van Leeuwen. *Routing Data Authentication in Wireless Ad Hoc Networks.* Sandia National Laboratories, SAND Report 2001-3119, October 2001.

[42] B. Van Leeuwen, J. Espinoza, and P. Sholander. *Effective Protocols for Mobile Communications and Networking.* Sandia National Laboratories, SAND Report SAND98-2753, December 1998.

[43] M. G. Zapata. *Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing.* Internet Draft, draft-guerrero-manet-saodv-00.txt, August 12, 2001.

[44] L. Zhou and Z. Haas. *Securing Ad Hoc Networks.* DARPA Report.

8. Appendix A (Simulation Details)

The simulation software used for this research was the OPNET Modeler and Radio simulation package Version 7.0. We also selected Cornell's Zone Routing Protocol (ZRP) [15] for our simulation studies, because it is a hybrid approach that includes both a proactive and reactive routing region. In addition, an OPNET model for the ZRP implementation was available from Pearlman and Haas [15] of Cornell University. In this section, we review some of the details and specifics of the models we created for our simulations.

8.1 Optimum Network Performance (OPNET):

Optimum Network Performance (OPNET) [32] is a comprehensive engineering system capable of simulating large communication networks with detailed protocol modeling and performance analysis capability. OPNET features include: graphical specifications of models; a dynamic, event-scheduled simulation kernel; integrated tools for data analysis; and hierarchical, object-based modeling. OPNET's hierarchical modeling structure accommodates difficult problems such as distributed protocol development and performance evaluation. With OPNET, network simulation modeling can be performed at a high-level using OPNET's extensive library of vendor devices, or at a very low level, where custom protocols using a graphical state transition diagram editor and C/C++ code can be developed. The functionality of each programmable block is defined by a process model that combines a finite state machine with the flexibility of the C programming language, and an extensive library of predefined models. Ad hoc routing protocols are still under development and so OPNET does not have predefined models for these protocols. Fortunately, OPNET is flexible enough to allow the user to develop and incorporate custom models into the simulation tool.

A network application model is used in OPNET to generate typical network traffic patterns. The simulator has standard client-server node models as well as provides the user with opportunity to create custom or user-defined application level traffic. Client-server and server-server interactions can be described as a series of phases, consisting of either data transfer or processing.

OPNET analyzes system behavior and performance with a discrete-event simulation engine. Discrete-event simulation is an approach that supports realistic modeling of complex systems that can be represented as a progression of related events. This approach models system behavior based on objects and distinct events such as the arrival of packets at various points in a network. Each object has associated attributes that control its behavior in the simulation.

The OPNET simulation tool is generally accepted as the simulation tool of choice in the DoD. Many future DoD communication systems are, or will soon be, network based. The Joint Staff J6 is in the process of providing a common set of network analysis tools to the defense community through its Networks and Warfare Simulations (NETWARS) effort [28]. The NETWARS effort, which uses OPNET as the simulation engine, has involved developing a number of peripheral tools that can be used for efficiently building OPNET network models, running simulations, and analyzing various OPNET results.

8.2 Modeling and Simulating the Zone Routing Protocol:

The simulation is implemented by creating individual mobile nodes that incorporate the ZRP routing protocol. These mobile nodes are distributed throughout a two dimensional area to represent a network. Each mobile node is identical and creates application-layer traffic as described in Section 3. Mobile nodes are given an initial position, a speed, and a random direction. The mobile nodes, shown in Figure 8.1, are further subdivided into node objects. The various node objects that make up a MANET node are:

- Movement object
- Traffic generation object
- Routing object
- Security overhead object
- Media access control (MAC) object
- Transceiver object
- Channel object

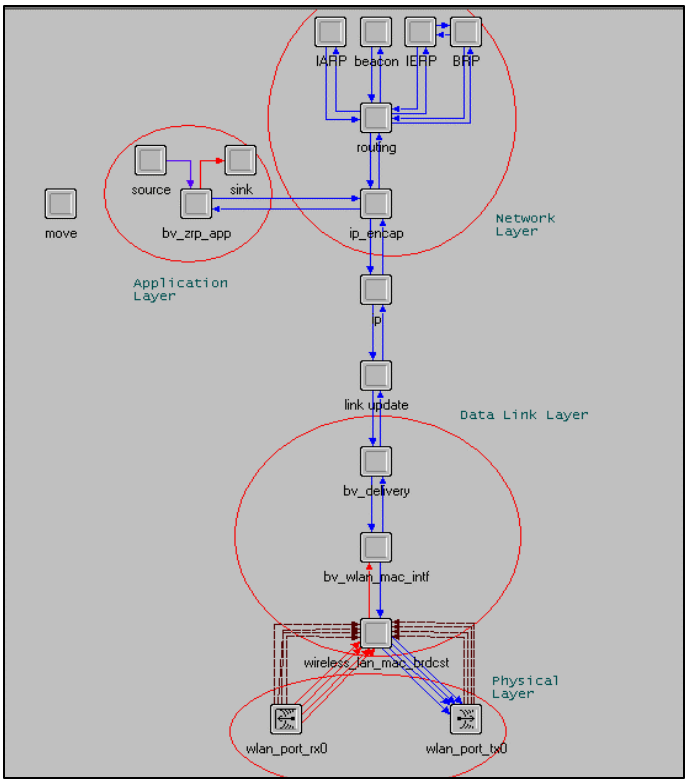

Figure 8.1: Model of MANET node

Movement object:

Nodes are permitted to move in a two-dimensional field. The node's initial position is defined by its (x,y) coordinate positing in the simulation area. A global speed is entered into the simulation

variable block. During the simulator's initialization phase a direction based on a random number is applied to the node. As the simulation runs, each node moves with the entered speed in the identified direction. Node positions are updated at each event occurrence.

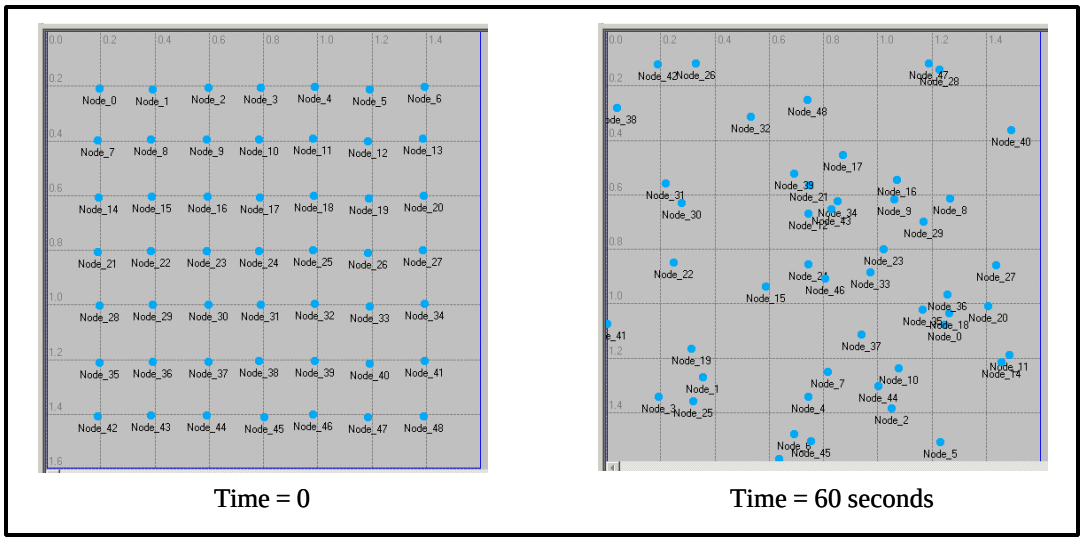

Figure 8.2: Movement object moves nodes in random direction during simulation

Traffic generation object:

A simple application layer traffic generation object is used to create traffic from each of the nodes. Source nodes generate packets destined for randomly chosen destination nodes. Any node in the network is an eligible choice for destination. The application-layer traffic is delivered on best effort basis and no retries are attempted if a packet cannot be successfully delivered. The application-layer generates a *session* of traffic to the destination node. Sessions start at a random time and have a predefined number of packets be directed to the destination node. It is at the beginning of each session that the random destination is selected. This session approach to packet destination choice is an attempt to mimic true network traffic, where larger streams of data destined to a node are packetized and shipped out in succession. The application-layer has a pause between sessions prior to the selection of a new destination for the next session. The traffic generation process object is shown in Figure 8.3.

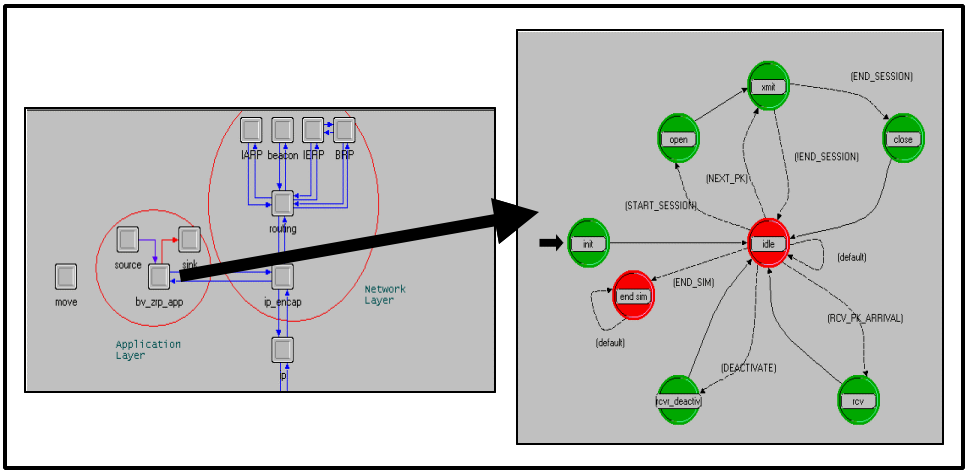

Figure 8.3: Traffic generation process object

Routing object:

ZRP is implemented in the routing object. This object accepts packets from the application-layer and determines if a route exists in the routing table or the message must be queued and a route discovery process initiated. If the node's routing table has a path to the destination node, the packet is appended with the next hop and the packet is submitted to the security overhead object and then to the MAC object for ultimate delivery to the destination.

The routing object is comprised of three sub-routing objects that are described in Section 2.1.1. The routing sub-objects are neighbor discovery protocol (NDP), intrazone routing protocol (IARP), and the interzone routing protocol (IERP).

Security overhead object:

The security overhead object accounts for overhead delays associated with cryptographic security features. The security overhead object accounts for delays and packet length increases that a security approach imposes on network packets. These overhead delays must be accounted for at each node upon both reception and transmission. We assume that a node's processor can process a single cryptographic feature at a time, thus a method for accounting for queuing delays is also included in this object. The security object block is shown in Figure 8.4.

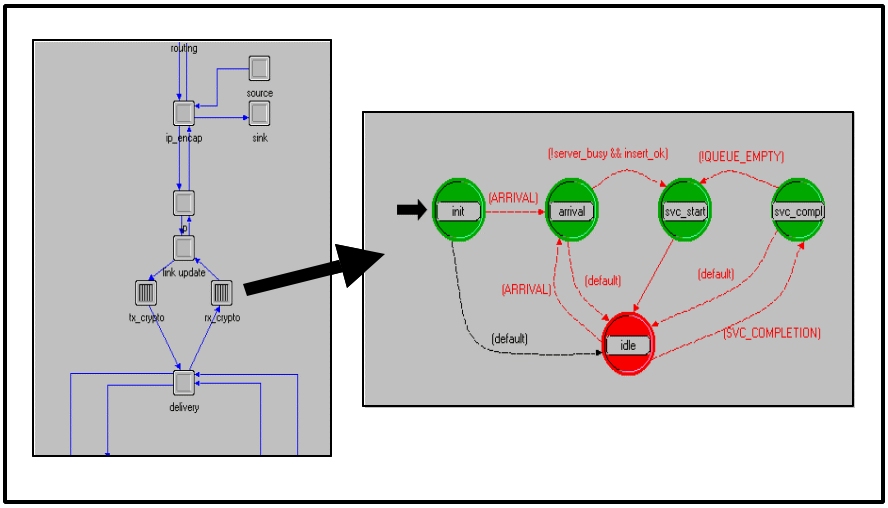

Figure 8.4: Simulation process blocks that incorporate link-layer cryptographic overhead

Media access control (MAC) object:

The MAC object includes the neighbor discovery protocol and the typical MAC functions. The neighbor discovery protocol periodically broadcasts a short packet that includes the broadcasting node's address. The broadcasts are heard by neighboring nodes within communication range.

Two MAC models are considered in this research; an ideal model and a high-fidelity model of the IEEE 802.11 protocol that is based on CSMA/CA. The ideal MAC object provides ideal scheduling of packet transmissions to avoid collisions and ensures packet delivery by assigning the packet to the destination node with the necessary delays. The ideal approach is used to focus the analysis on the routing protocol. The 802.11 MAC is explained below.

Transceiver object:

Simulations are executed with either an ideal transceiver object or a more realistic transceiver object associated with OPNET's 802.11 wireless LAN model. The ideal transceiver object accounts for signal strength and receiver sensitivity based on a predefined distance. Transmissions are received free from errors provided the receiving node lies within the predefined distance; otherwise the transmissions are lost. Channel contention issues for the ideal transceiver are limited to contention of packets from the transmitting node. The transceiver object queues packets that require transmission until the node completes transmitting previously queued packets. Broadcast messages are unicast to each node within range.

OPNET's 802.11 wireless LAN model supports a high fidelity representation of the transceiver and channel access. This model incorporates all LAN nodes that share the wireless channel, thus network wide channel contention issues can be addressed. In theory the model will improve the fidelity of simulations; unfortunately OPNET's model does not perform as advertised and simulation results with this model are not included.

Channel object:

OPNET simulates the communications between two nodes with a transceiver pipeline process. The transceiver pipeline mimics various aspects of the channel behavior such as transmission delay, link closure, channel match, transmitter and receiver antenna gain, propagation delay, received power, background noise, interference noise, receiver required signal-to-noise (SNR) ratio, bit error rate, error allocation, and error correction function. However, as the simulation includes more of these channel aspects the simulation run-time increases. In our ideal MAC and channel simulation, the various channel impacts other than propagation delay are not included. Most of the channel characteristics are not necessary for an evaluation of the security overhead impacts of on lower-layer performance. Figure 8.5 illustrates the channel aspects available to the OPNET simulator.

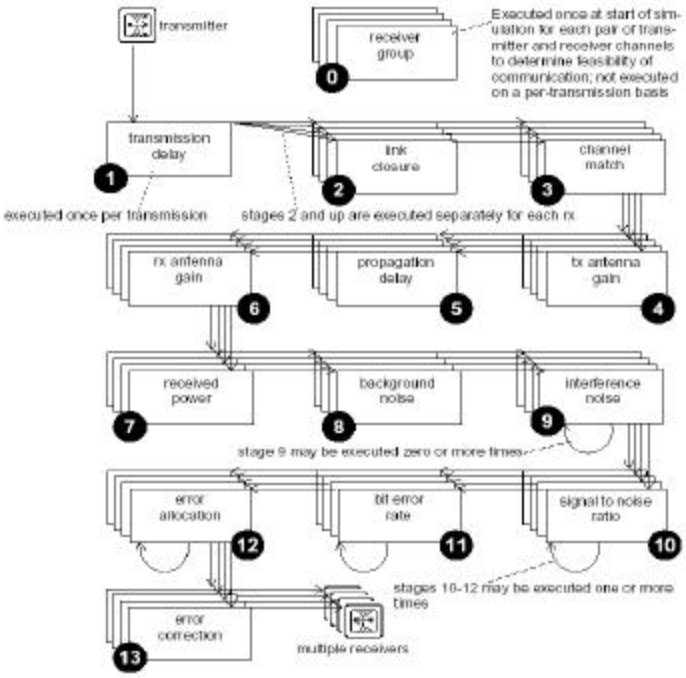

Figure 8.5: Elements that are modeled in the OPNET Radio Channel [32]

8.3 Modeling and Simulating the IEEE 802.11 MAC

Simulations were considered with OPNET's high-fidelity model of the IEEE 802.11 MAC sub-layer. These simulations include wireless channel aspects such as channel access, contention issues, and the affects of the IEEE 802.11 standard virtual channel access techniques. Channel impairments due to environmental noise can be included in simulations with the high-fidelity model. Unfortunately, at the completion of this research several bugs remained in OPNET's IEEE 802.11 wireless MAC model and our confidence in the results did not meet our standards for this report.

The IEEE 802.11 standard requires that all nodes in a cell, called a Basic Service Set (BSS), be within a limited range of other nodes in the BSS. The range is defined such that all nodes in the BSS must be able to receive either the RTS or the CTS of two nodes that wish to communicate and be within range of an access point. The access point is a gateway to some larger network.

As written, the IEEE 802.11 standard is not intended for multihop networks. However, the protocol can easily be applied to such a network. The standard requires that a node with its NAV set must ignore all communications until its NAV counts to zero. If a node sends a packet to another node that has its NAV set, the packet will be ignored and the sending node will retransmit after the appropriate amount of time.

It is possible to make the transition from a single cell network using the IEEE 802.11 MAC to a multihop network using IEEE 802.11 technology. Furthermore, academic research done in the area of multihop wireless networks uses the 802.11 MAC approach. As such, one would think that OPNET's extensively detailed 802.11 model would be ideal for simulating a multihop network. However, this is not the case. OPNET has chosen to follow the IEEE 802.11 admonition that all nodes *should* hear either the RTS or the CTS and has essentially ignored the admonition that all nodes *should* ignore communications when their NAV is set.

Study of OPNET's IEEE 802.11 MAC code indicates the model requires *all* network nodes be part of the same BSS. That is, all nodes *must* receive either the RTS or CTS. When a node has its NAV set and it is the target of another transmission, the model's fragmentation and reassembly module fails to resolve the fact that the packet was not really received. The simulation fails when the module attempts to remove from the reassembly buffer a packet that that has not been stored in the buffer. In our simulations, any nodes that attempted to communicate with another node that had its NAV set caused the simulation to come to an abrupt end.

Because the hidden terminal problem, is inevitable in multihop wireless networks, it is clear that nodes will attempt to communicate with nodes that have their NAV set. Node movement or changing of BSS also means that nodes will attempt to communicate with nodes that have their NAV set. The small failure to properly account for unreceived packets makes OPNET's IEEE 802.11 MAC model unusable for anything but a simulation of single cell implementations of the IEEE 802.11 MAC. Unfortunately, the way the OPNET code is written makes fixing the problem non-trivial.

8.4 Simulation Statistics Collection

This research is primarily interested in collecting statistics related to routing protocol control data and the impacts of security on the network control data. The OPNET simulator only provides a partial solution to our data collection interests. OPNET is used to collect individual packet latency results, and incorporate an inherent statistical collection method for each of their models. However, collecting statistics in our custom models and over several simulation runs is quite cumbersome to use the OPNET statistic collection method. In this research, simulation data is collected with custom variables and is written to a data file upon completion of the simulation. Compilation of the data is performed with Excel.

DISTRIBUTION:

	MS	
1	0839	R. L. Craft, 16000
1	0775	M. J. Eaton, 5852
1	1138	L. J. Ellis, 6502
1	1002	P. Garcia, 15202
1	0785	D. P. Duggan, 6516
1	9201	M. M. Goldsby, 8114
1	1125	J. J. Harrington, 15252
1	0785	R. L. Hutchinson, 6516
1	9201	M. M. Johnson, 8114
1	9101	R. D. Kyker, 8232
1	0812	J. H. Maestas, 9334
1	0785	T. S. McDonald, 6514
1	1125	A. K. Miller, 15252
1	0529	M. B. Murphy, 2346
1	1004	F. J. Oppel III, 15221
1	0455	R. D. Pollock, 6501
1	1170	R. D. Skocypec, 15310
1	0784	M. J. Skroch, 6512
1	0785	J. E. Stamp, 6516
1	0801	M. R. Sjulin, 9330
1	0806	T. D. Tarman, 9336
1	0455	R. S. Tamashiro, 6517
5	0785	M. D. Torgerson, 6514
1	0784	R. E. Trellue, 6501
5	0785	B. P. Van Leeuwen, 6516
1	0103	J. P. VanDevender, 12100
1	0451	S. G. Varnado, 6500
1	0806	E. L. Witzke, 9336
1	0785	W. F. Young, 6516
1	0188	LDRD Program Office, 1030 (Attn: Donna Chavez)
2	0899	Technical Library, 9616
1	0612	Review & Approval Desk for DOE/OSTI, 9612
1	9018	Central Technical Files, 8945-1