2006

A Quantitative Analysis of Performance in a Multi-Protocol Ad Hoc 802.11b-based Wireless Local Network

Paul Christian Nielsen Jr.
Nova Southeastern University, drcnielsen@gmail.com

A Quantitative Analysis of Performance in a Multi-Protocol Ad Hoc 802.11b-based Wireless Local Network

by

Paul Christian Nielsen, Jr.

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Graduate School of Computer and Information Science
Nova Southeastern University

2006

We hereby certify that this dissertation, submitted by Paul Christian Nielsen Jr., conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

Marlyn Kemper Littman, Ph.D. Date
Chairperson of Dissertation Committee

Maxine S. Cohen, Ph.D. Date
Dissertation Committee Member

Sumitra Mukherjee, Ph.D. Date
Dissertation Committee Member

Approved:

Edward Lieblein, Ph.D. Date
Dean

Graduate School of Computer and Information Sciences
Nova Southeastern University

2006

An Abstract of a Dissertation Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

A Quantitative Analysis of Performance in a Multi-Protocol Ad Hoc 802.11b-based Wireless Local Network

by
Paul Christian Nielsen, Jr.

May 2006

The popularity of the Internet and the growing demand for ubiquitous connectivity accelerate the need for viable wireless local area network (WLAN) solutions. As a consequence, increasing number of manufacturers have adopted the Institute of Electrical and Electronic Engineers (IEEE) 802.11a/b/g set of WLAN standards and produced inexpensive wireless products to expand capabilities of existing LANs. IEEE 802.11b wireless products are widely accepted. Mobile ad hoc networks, a variant of the 802.11 standards, exist without the requirement for a wired infrastructure or host to provide routing, connectivity, and maintenance services. Because of the high variability of environments in which ad hoc networks operate, numerous routing protocols are proposed. Research indicates that these protocols are unsuited for efficient operation in multiple environments. In this investigation, the author examined the effect of multiple protocols on throughput and end-to-end delay in simulated ad hoc networks.

The author selected the ad hoc on-demand distance vector (AODV) and dynamic source routing (DSR) routing protocols for this research. The outcomes from the simulations conducted indicated increased end-to-end delay and reduced packet throughput as a result of the mixed populations of the AODV and DSR ad hoc routing protocols. The results also indicated that increasing node density and velocity improved packet throughput and reduced end-to-end delay.

Acknowledgements

I wish to express my deepest appreciation to my wife, Lita. Without her support and sacrifice this effort could not have been accomplished. I wish to thank my family whose constant encouragement helped me through the difficult patches of this journey. I offer my sincere thanks and appreciation to my dissertation chairperson, Dr. Marlyn Kemper Littman whose encouraging words and insightful guidance kept me from wandering too far off the path to completion. I wish also to thank Dr. Maxine Cohen and Dr. Sumitra Mukherjee for their contributions to my education as well as their guidance as members of my dissertation committee.

Table of Contents

Appendixes

List of Tables

Tables

List of Figures

Figures

Chapter 1

Introduction

The popularity of the Internet and ubiquitous computing contributes to the growing demand for wireless local area network (WLAN) solutions. The development of the Institute of Electrical and Electronic Engineers (IEEE) 802.11 suite of WLAN standards including 802.11a and 802.11b in the 1990s increased interest in mobile ad hoc networks or MANETs (Bruno, Conti, & Gregori, 2001). As a result, manufacturers including Cisco, Sony, Belkin, D-Link, and Microsoft currently produce inexpensive wireless products compliant with these standards to expand capabilities of existing wireline LANs by adding wireless connectivity.

IEEE 802.11b wireless products are widely accepted. Higher speed IEEE 802.11a and 802.11g standards-based devices are readily available on store shelves. Hardware providing mobility and access through wireless technology is pervasive in business and home environments. New products such as Personal Digital or Data Assistants (PDAs) and notebook computers feature embedded wireless capability. Yet, most networks still cannot support communications within a network without a wireline host (Sudame & Badrinath, 2001). Technical advances in mobile computing and ad hoc WLANs offer the potential for ubiquitous connectivity without the need for a host (Boukerche, 2004). Factors such as mobility, topography, and interference make achieving this goal challenging (Kim, Lee, & Helmy, 2004). The wires are severed, but the fundamental nature of the client-server LAN has not yet changed.

The primary characteristics of ad hoc networks are their temporary nature, self-organization, mobility, and capability in operating without an infrastructure-based host or server that provides basic communications services (Lee, Han, & Shin, 2002; Marina & Das, 2001; Mochocki & Madey, 2005). Instead, each client must contain the necessary software to provide basic discovery, routing, and connection maintenance routines for establishing and maintaining temporary communications with nearby similarly equipped clients (Buttyán & Hubaux, 2003; Lee et al., 2002). These functions are normally the domain of the routing protocol.

The concept of a mobile ad hoc network (MANET) was developed in the 1970s with mobile packet radio technology (Hubaux, Gross, Boudec, & Vetterli, 2001; Mochocki & Madey, 2005). In the last few years, novel routing protocols have been proposed specific to the mobile ad hoc environment (Papapetrou & Pavlidou, 2003; Park & Park, 2004; Valera, Seah, & Roa, 2003). Subsequent research has shown that the previously suggested protocols were not optimized for data-intensive, Quality of Service (QoS) or multi-hop communications as required in military and industrial environments (Lundberg, 2004). According to Kargl, Nagler, and Schlott (2002), at present there is no single routing protocol that will manage the needs of all conceivable mobile networking scenarios.

According to Hu, Perrig, and Johnson (2002) and Kawaguchia, Toyamaa, and Inagakia (2000), the demand for self-organizing wireless or ad hoc networks that can exist without a wireline infrastructure is evident in sectors that include business, education, and government. Boukerche (2004) and Buszko, Lee, and Helal (2001) point out that a functioning ad hoc network is a viable solution for supporting communications capabilities in environments such as battlefields and disaster sites where a wireline infrastructure deployment is not possible.

The difficulties uniquely associated with mobile wireless ad hoc networks continue to be active areas of research (Abolhasan, Lipman, & Chicharo, 2004; Viswanath, Obraczka, & Tsudik, 2004). The IEEE 802.11 Working Group for Wireless Local Area Networks is currently developing the new IEEE 802.11e standard to address QoS issues that exist in the IEEE 802 wireless LAN family (Mangold, Choi, May, Klein, Hiertz, & Stibor, 2002). Additionally, researchers such as Kargl et al. (2002) are designing a framework for developing common WLAN functions provided by most mobile ad hoc routing protocols. The literature includes a steady stream of novel approaches to solve the routing problems in ad hoc networks (Hu, Perrig, et al., 2002; Marina & Das, 2001; Roy & Garcia-Luna-Aceves, 2002). Nonetheless, advances in information technology (IT) relating to WLANs fall short of meeting the needs of an ad hoc network (Günes, Sorges, & Bouazizi, 2002; Viswanath & Obraczka, 2002). This is especially true when devices are highly mobile or when topology is variable (Kassabalidis et al., 2001).

According to Viswanath and Obraczka (2002) and Boleng, Navidi, and Camp (2002), there is no single routing protocol solution available for mobile ad hoc wireless networks (MANETs) that effectively supports MANET implementations in a wide range of environments. Nevertheless, current research efforts focus on creating single protocol solutions that require acceptance of tradeoffs in efficiency (Williams & Camp, 2002).

A recent example is the Sharp Hybrid Adaptive Routing Protocol (SHARP) proposed by Ramasubramanian, Haas, and Sirer (2003). Recognizing the inherent tradeoffs between on-demand or reactive and proactive protocols, these authors suggested a hybrid protocol that balances the benefits of reactive and proactive protocols. Reactive protocols determine paths only when a data packet is ready to transmit as opposed to proactive protocols. Proactive protocols actively maintain tables of path information whether data packets are ready to be transmitted or not. As a consequence, proactive protocols have lower delay and higher overhead compared to reactive protocols (Zhang & Jacob, 2003). The SHARP hybrid routing protocol adjusts dynamically to the need for route discovery and route table propagation on a per node basis (Ramasubramanian et al.). According to the authors, SHARP utilizes both a proactive and reactive protocol. While they claim that any reactive protocol may be used, the proactive portion is constituted by their SHARP protocol.

Researchers such as Haas and Pearlman (2001), Roy and Garcia-Luna-Aceves (2002), and Navid, Houda, and Bonnet (2000) also developed hybrid approaches that balance reactive and proactive approaches to ad hoc routing. In their work, however, only a single routing protocol is suggested that contained proactive and reactive elements. The use of two separate protocols in each node for ad hoc routing is a relatively new concept (Ramasubramanian et al., 2003).

There continues to be considerable research effort devoted to seeking single protocol solutions. However, increasingly researchers acknowledge that a single protocol solution is unlikely (Viswanath & Obraczka, 2002; Boleng et al., 2002). Without a single protocol standard in ad hoc routing, it is likely that the market will produce a number of products with competing technologies. This pattern has been repeated many times prior to the release of firm industry standards. Recent examples where a lack of suitable standards results in multiple competing technologies entering the marketplace include modem modulation techniques, high-speed serial connections, WLANs, and cellular telephone technology. In some cases, the products became obsolete when a standard was released. In other cases, multiple de-facto standards are developed.

Problem Statement

The problem investigated in this research was performance degradation resulting from multiple ad hoc wireless protocols operating in the same physical environment. The study of ad hoc wireless protocols in multi-protocol or heterogeneous environments is emerging as researchers conclude that single protocol solutions fail to meet the requirements of demanding ad hoc environments (Boleng, Navidi, & Camp, 2002; Samar, Pearlman, & Haas, 2004; Viswanath, Obraczka, & Tsudik, 2004).

A large body of work is devoted to the development of single routing protocol solutions to meet challenges of ad hoc networking (Papapetrou & Pavlidou, 2003). Researchers such as Lee et al. (2002), Kannan, Mellor, and Kouvatsos (2003); Lee, Hsu, Gerla, and Bagrodia (2000); and Marina and Das (2001) present comparisons of proposed new routing protocols against existing protocols. Studies by Bhargava (2003), Boukerche (2004), Das, Perkins, and Royer (2000); and Lu, Wang, Zhong, and Bhargava (2003); and Williams and Camp (2002) compare the efficiency of different protocols. According to Xu and Gerla (2002), most routing protocols in MANETs are designed for homogeneous environments that have performance problems related to scale. Recently, authors such as Calafate, Garcia, and Manzoni (2003) and Ge, Lamont, and Villasenor (2005) examined the impact of a heterogeneous environment on ad hoc routing protocols. Their research suggests that multiple protocols are required to manage dynamic heterogeneous environments. Other researchers such as Solis and Obraczka (2004) are developing a framework for multi-protocol interconnections in heterogeneous environments.

The challenges in designing practical, efficient, and flexible ad hoc routing protocols are formidable (Sinha, Krishnamurthy, & Dao, 2000). After a decade of concerted effort, researchers continue developing routing protocols in an attempt to meet the challenging demands of ad hoc routing environments (Papapetrou & Pavlidou, 2003). Heterogeneous environments contribute to the complexity of identifying protocol solutions for functioning in large and highly mobile ad hoc networks (Xu & Gerla, 2002).

Based on the research of Calafate et al. (2003) and Solis and Obraczka (2004), this author determined that the current lack of standards in routing protocols for wireless IEEE 802.11b-based ad hoc networks promotes deployment of multiple ad hoc routing protocols in the environment. Consequently, situations arise where multiple ad hoc routing protocols may be deployed in the same geographic area. Based on the work of Abolhasan, Lipman, and Chicharo (2004); Boukerche (2004), and Tseng, Ni, Chen, and Sheu (2002), this author determined that multiple routing protocols operating in the same geographic area will degrade the efficiency and effectiveness of the protocols in a manner similar to the hidden terminal problem (Sheu & Chen, 2002).

Ad hoc nodes are subject to the hidden terminal problem (Haas, Deng, Liang, Papadimitratos, & Sajama, 2002; Kuri & Kasara, 2001; Prakash, 2001). The hidden terminal problem exists when a node attempts to communicate with another node after failing to detect the transmission of a third node. The third node is outside the detection range of the sending node, but within the range of the intended receiver (Haas et al., 2002). This interference can create unidirectional links as well as other transmission failures significantly degrading the efficiency of the network (Calafate et al., 2003; Tseng et al., 2002). The hidden terminal problem is also the result of differences in the power or transmission range of nodes in heterogeneous environments (Calafate et al.).

A search of the literature in print and in the digital libraries of the Institute of Electrical and Electronic Engineers (IEEE) and the Association for Computing Machinery (ACM) indicated that research involving heterogeneous ad hoc environments is a relatively recent development with few quantitative studies involving multiple independent ad hoc routing protocols operating within the same environment. Xu and Gerla (2002) noted that work prior to their study focused solely on homogeneous environments of a single routing protocol. A body of literature covering the hidden terminal problem may offer insights into the disruptive effects of a multi-protocol environment.

Statement of Goal

The goal of this research was to examine the effect on efficiency of multiple routing protocols coexisting in the same wireless ad hoc network. Stated in the affirmative, this researcher proposed the hypothesis that there is a difference in packet delivery ratio and latency in environments containing multiple ad hoc routing protocols and those containing a single ad hoc routing protocol. The primary goal stated as a null hypothesis was: There is no difference in efficiency as measured by packet delivery ratio and latency between environments containing a single ad hoc routing protocol and one containing two ad hoc routing protocols.

The initial study in this investigation simulated two protocols, specifically, dynamic source routing (DSR) and ad hoc on-demand distance vector (AODV) functioning within the same bounded experimental area. Independent variables were the number and mobility of each node containing the protocols under study. Dependent variables were (a) the ratio of number of packets sent to the number of packets received (delivery ratio) and (b) the end-to-end delivery delay (latency).

An extensive review of current in-print and online literature covering the past 10 years from the IEEE and the ACM indicated research into heterogeneous multi-protocol ad hoc environments was an emerging area of study. Research also indicated that software agent technology applied in solving other problems present in the wireless and ad hoc domains was also applicable to heterogeneous environments (Kawaguchi & Inagaki, 2000; Marwaha, Tham, & Srinivasan, 2002; Spohn & Garcia-Luna-Aceves, 2001).

Agent technology is also applied to monitoring environmental variables and making dynamic changes to routing protocol behavior. For example, Viswanath and Obraczka (2002) utilized an intelligent software agent to proactively modify the flooding mechanism used in a simulated velocity triggered and dynamically switched routing environment. The agent used the environmental variable node velocity to dynamically select between different flooding mechanisms.

Agents are useful in the formation of ad hoc networks (Kawaguchi & Inagaki 2000). Chacón, Bell, and McCormick (2000) documented how agents improved efficiency and robustness in networks. Dunne (2001) showed that mobile agents could discover network resources in peer-to-peer networks. Peer-to-peer networks share the key characteristics of self-organization, decentralization, route discovery, and route maintenance with ad hoc networks (Hu, Das, & Pucha, 2003). Günes et al. (2002) developed an ant-colony based routing protocol using ant-like agents first proposed by Di Caro and Dorigo (1998).

Relevance and Significance

A hallmark of ad hoc networks, self-organization can create significant impacts in the education, government, and commercial sectors. Hubaux et al. (2001) state that by their very nature ad hoc networks can bring about a paradigm shift in the way networks operate. According to Hubaux et al., ad hoc networks can lead to fundamental changes in the relationships between information technology (IT) and societal organizations by changing the nature of networking and self-organizing structures. Despite the potential benefits, ad hoc networks present a number of unique challenges that remain unsolved. According to Günes et al. (2002), route determination was the main problem due to node mobility. Presently, no single ad hoc routing protocol provides flexible solutions in the variable environments in which ad hoc networks operate (Denko, 2003; Kargl et al., 2002; Günes et al.).

Despite efforts of investigators such as Choudhury, Paul, and Bandyopadhyay (2004); Haas and Pearlman (2001), Papapetrou and Haas (2003), and Prakash (2001), challenges in the field of MANETs remain largely unsolved. Message routing and network path discovery are major challenges that are actively discussed in the literature in the field of mobile ad hoc networks (Al-Shurman, Yoo, & Park, 2004; Calafate, Garcia, & Manzoni, 2003). Discovery techniques such as flooding or sending messages to all nodes (Lundberg, 2004; Obraczka et al., 2001) resulted in other problems such as broadcast storms (Li & Cuthbert, 2004; Tseng et al., 2002). With battery operated mobile devices, energy efficiency was a major issue (Abolhasan & Wysocki, 2003; Wieselthier, Nguyen, & Ephremides, 2001).

Numerous routing and communications protocols were proposed for implementing wireless ad hoc networks (Kawaguchi & Inagaki, 2000; Günes et al., 2002). According to Kassabalidis et al. (2001), problems with network throughput and delay, diversity of equipment type, and reliability and scaling negatively affected development of new protocols to accommodate the complexities in ad hoc networks. Kassabalidis et al. concluded that traditional static and dynamic routing protocols are unable to manage networks that are large in scale, feature rapidly changing topology, or have unstable linkages.

Chacón et al. (2000) proposed the use of autonomous software agents for ad hoc networks to solve routing problems. As a consequence of its small footprint and mobility, agent technology is especially useful in creating solutions in ad hoc routing and distributed computing applications (Illmann, Krueger, Kargl, & Weber, 2001). Agents have the capability of some autonomous action. Choudhury et al. (2004) presented a strong case for utilizing mobile agents to perform complex network management functions and encouraged continued development of adaptive agents.

Kassabalidis et al. (2001) identified advanced swarm-based mobile agents as a possible solution to the difficult problem of route discovery in highly dynamic heterogeneous wireless ad hoc networks. Based on the earlier groundbreaking research of Di Caro and Dorigo (1998), Kassabalidis et al. (2001) promoted the use of an antnet-like agent set that uses stigmergy, or indirect communications. In a stigmergic system, agents communicate indirectly with one another by placing information in predefined areas of the environment such as a cache or table that may be used by other agents as necessary. This indirect approach is similar to the use of pheromone trails created by foraging ants and followed by other ants who do not directly communicate with the original trailblazer.

According to Haas and Pearlman (2001), mobile agents represent a technological means to provide an adaptive solution for mediating a multi-protocol environmental context. White, Pagurek, and Duego (2002) indicate that multiple ad hoc protocols may be deployed to accommodate changing topography, node mobility, and node failure. This situation arises in field environments. The static routing protocols mentioned previously do not adapt well to situations where nodes are highly mobile (Haas & Pearlman). In addition, most of the proposed routing protocols do a poor job of managing unidirectional links (Li & Rus, 2003; Prakash, 2001). Constantly changing connectivity along with location changes create high processing overhead and communications chatter that interfere with protocol efficiency. Multiple protocols can address problems created by scalability and node mobility and provide the adaptive responses necessary to facilitate optimal routing in dynamic environments (Günes et al., 2002; Puliafito & Tomarchio, 2000). Along with resolving technical challenges such as self-organization, node diversity management, security, delay control, and quality of service (QoS), there are other benefits to developing stable adaptive ad hoc networks (Poon & Li, 2003).

Utilizing computing technology requires heavy investments in hardware and software. Typically, these investments involve expenditures for a traditional infrastructure. The proliferation of wireless devices adhering to the Institute for Electrical and Electronics Engineers (IEEE) 802.11b specification and other IEEE 802.11 WLAN extensions as well as new standards under development by organizations such as the IEEE create the opportunity to add standards-based mobility and transient ad hoc WLANs to the IT collection of productivity tools, thereby reducing the need for extensive infrastructure investments.

The ability to form robust and reliable wireless ad hoc networks removes some of the necessity of maintaining expensive wireline infrastructure and provides a potential operating environment that is less subject to critical failure (Buszko et al., 2001). The ability to develop functional networks without the expense of extending in-place services has cost savings implications for corporations. Current wireline networks are unsuited for short-term additions of workstations. Intermittent use areas such as conference rooms often require dedicated infrastructures that waste resources. Integration of wireless ad hoc networking into wireline environments can substantially reduce the need for this dedicated equipment.

Protocol performance evaluation and simulation are important tools in developing new protocols (Abolhasan & Wysocki, 2003; Boukerche, 2004). Most performance comparisons have focused on contrasting single protocols in homogeneous environments (Buttyan & Hubaux, 2003; Haas & Pearlman, 2001; Shen & Jaikaeo, 2003). With the emergence of heterogeneous networks and new ad hoc routing protocols such as Hierarchical Optimized Link State Routing (HOLSR) additional simulations of comparative performance are necessary (Ge, Lamont, & Villasenor, 2005). In addition, evaluation of existing protocol performance in multi-protocol environments provides essential data for mobile ad hoc protocol developers.

Barriers and Issues

The author examined a relatively new branch of computing that is still evolving. Currently, there are few options providing an adaptive response to highly variable wireless ad hoc network environments (Kargl et al., 2002; Boleng et al., 2002). The proliferation of wireless-enabled Personal Digital Assistants (PDAs) and notebook computers greatly complicates the issue by creating a heterogeneous mix of processing, capacity, power, range, and display capabilities (Ge et al., 2005; Xu & Gerla, 2002).

Work to address the difficult issues of routing and messaging in ad hoc networks is ongoing (Mochocki & Madey, 2005; Papadimitratos & Haas, 2003). However, the variability of environmental factors such as topography, unidirectional links, mobility, and power variances make single protocol solutions improbable (Xu & Gerla, 2002). Several efforts are underway exploring the relatively new research area involving heterogeneous mobile ad hoc environments (Ge, Lamont, & Villasenor, 2005; Mochocki & Madey). Mochocki and Madey explored the difficulty in simulating heterogeneous networks that combine MANETs and sensor networks. Ge, Lamont, and Villasenor approached the issues of routing and messaging in heterogeneous networks by enhancing the performance of the Optimized Link State Routing (OLSR) protocol. Wedde et al.

(2005) contributed improvements to energy management in MANETs through a new protocol based on the behavior of bees. Despite these efforts, heterogeneous multiple protocol environments continue to provide a more versatile alternative than single protocol environments.

Major barriers to this research involved access to detailed data on the design and function of the proposed routing protocols and development of a suitable test environment. Typically, data on the design and function of the current routing protocols were available within the literature at a high level. However, while protocols are in the development stage, details are subject to change (Denko, 2003; Lee et al., 2002; Park & Park, 2004; Prakash, 2001). In addition, difficulties in gaining permissions from developers of auxiliary programs used in this research delayed starting the simulations (Jiang & Camp, 2002; Williams & Camp, 2002). Consequently, the author was restricted in terms of the number of routing protocols that were available for integration into the simulations by this author.

Access to an appropriate test environment was also a major barrier. Equipment and software costs approached several thousand dollars to create a physical network of several dozen nodes. Creation of an environment with a sufficient number of physical nodes to provide relevant statistical measures was highly impractical (Kargl et al., 2002). Therefore, the author relied on simulations. A review of the literature indicated that most proposals for routing protocols and other ad hoc network support protocols relied on simulations to provide quantitative data (Boleng et al., 2002; Kargl et al.). Simulations in this inquiry were necessary because of the large set of variables that existed in the ad hoc wireless domain.

The simulation environment was complex. As a consequence, extensive training was necessary. In order to develop the appropriate skills necessary to effectively use the selected simulator and the associated tools and programming languages the author participated in several training sessions.

Computation requirements of simulated environments tax the capabilities of personal computers (PCs) since most simulators employ UNIX, Linux, or mainframe operating systems. The author used both Linux and Windows environments limited to single Intel-based processor machines operating at moderate speeds. As a consequence, the author selected network simulator version 2 (*ns-2*). Versions of *ns-2* were available that operated in both Linux and Windows environments simulating UNIX. Widespread support contributes to the validity of the *ns-2* simulation environment. *ns-2* operations are supported by documentation provided by an active user community.

Research questions to be investigated

The primary questions that were investigated included the following:

- What was the effect on packet delivery ratios and end-to-end delay of message packets in multi-protocol environments?
- What was the relationship between population numbers of each protocol and the degradation of packet delivery ratios and end-to-end delay?

Additional secondary questions considered included the following:

- Which of the two selected ad hoc protocols were most disruptive when inserted into a relatively homogeneous environment?

- Were projected gains in efficiency using agents offset by increases in bandwidth required to accommodate additional overhead traffic?
- Were swarm-based agent approaches more efficient than dedicated purpose agents?

Secondary questions were addressed through an extensive review of the literature in pursuit of answers to the primary questions. Additional research involving on these questions is suggested as part of future research suggestions. Questions without specific answers generated as part of this inquiry are part of the future work statement at the conclusion of this dissertation.

Limitations and Delimitations

The most significant limitation in this research was the lack of funding. Funding limited the equipment and software available to conduct the simulations required by the inquiry. As a consequence, the author selected the Microsoft Windows based UNIX simulator *cygwin* and the *ns-2* network simulator. Both software packages were available free of charge to the research and educational community.

While the *ns-2* simulation package is widely accepted by the research community, the simulator program is still evolving. Two significant new releases during this inquiry that resolved programming problems and added new functionality became available. The author assumed that the version specified for this investigation produced correct data that could be duplicated using the same version on another system. As a consequence, the author verified results on Linux and Windows platforms. *ns-2* also represented a significant limitation to this study. This simulator was originally designed to simulate homogeneous environments consisting of a single routing protocol. As a consequence, the author created a simulated heterogeneous environment through the development of creative control scripts.

Another limitation was time. The field of wireless ad hoc routing is rapidly evolving. New routing protocols are introduced at a rapid rate in addition to advances in agent technology. The author was challenged in keeping current on these changes and producing the research proposed. As a consequence, the author conducted this research in the most expeditious manner possible.

A delimitation of this research was the participation of the user support group for *ns-2* and the wireless ad hoc research community at large. These individuals were required to resolve unforeseen problems with the simulation environment and the protocols contained therein. This community of users provided indirect and intermittent support. The accuracy of the information they provided was assumed to be correct.

Definition of terms

This section contains a list of terms that were used throughout this investigation.

Active message – A message that is contained within an agent wrapper. This allows messages to self-propagate or otherwise exhibit autonomous behaviors not possible without the added agent intelligence (Li & Rus, 2000). Active messages can also remain in an intermediate node until a destination node is available.

Agent – An encapsulated computer system or program situated in a software application or networked environment and capable of flexible, autonomous action based on specific environmental events in order to meet its design objectives (Jennings, 2001).

ALOHANET – Named for the Palo Alto, California, Research Center Aloha Network. ALOHANET was the first wireless packet network used to connect computers on the Hawaiian Islands. Developed by Norman Abramson, ALOHANET was also the first network to connect to the mainland Advanced Research Projects Agency Network (ARPANET) (Microsoft, 2002).

ALOHA Protocol – A fixed wireless communications random access protocol based on a star network topology that used a collision detection and time setback mechanism. This technique became widely used in the Ethernet protocol (Naor & Levy, 2001).

AODV (Ad Hoc On-demand Distance Vector Routing) – An improved version of the Direct Sequenced Distance Vector (DSDV) protocol. AODV reduces overhead and bandwidth consumption by reactively maintaining distance vector tables rather than doing them proactively as in DSDV. Table updates are done through a flooding process (Ye, Krishnamurthy, & Tripathi, 2003)

ARA (Ant colony based Routing Algorithm) - One of the new class of routing protocols based on swarm intelligence. Like other swarm intelligence protocols, this algorithm uses ant-like agents to establish paths to destinations. ARA is based on the concept of stigmergy or indirect communications (Günes et al., 2002).

ARPANET (Advanced Research Projects Agency NETwork) – The progenitor of the modern Internet. ARPANET was originally a government funded experiment in establishing data networks between computers in military and educational environments (Patel et al., 2003)

AX.25 – A modified version of the X.25 protocol adopted by amateur radio operators to send text messages between radio stations. The protocol operates at the Transport Layer or Layer 4 of the Open System Interconnection (OSI) model (Beech, Nielsen, & Taylor, 1998). AX.25 provides error correction, and employs datagrams for transmission (Newton, 2006).

BANT (Backward ANT) – An ant-like software agent. Backward ants generally travel from a destination discovered by a forward ant (FANT) back to the source. The backward ant re-enforces the pheromone trail and contains the path information to the destination that was obtained by the forward ant (Günes et al., 2002).

Bluetooth – Defined by the IEEE 802.15 specification as a short range low-power wireless networking system. Bluetooth interlinks voice and data devices in wireless Personal Area Networks (WPANs) and operates in the unlicensed Industrial, Scientific, and Medical (ISM) band at 2.4 to 2.4835 Gigahertz (GHz) (Hać, 2003).

Broadband – Defined by Newton (2006) as a transmission facility that has a capacity greater than a voice grade line of 3000 Hertz (Hz). Other industry definitions vary. In the telecommunications and data industries, broadband refers to data rates of 1.5 Mbps (megabits per second) or greater.

Broadcasting – The process of sending a packet to all directly reachable neighboring nodes (Kargl et al., 2002). Broadcasting is commonly used to obtain routing and other information from within a network.

BSS (**Basic Service Set)** – Defined under the 802.11 specifications. BSS is the basic building block of a WLAN and consists of member stations that are in communications with each other. The BSS may also form a Distribution System (DS) used to interconnect BSSs (Hać, 2003).

CEDAR (Core Extraction Distributed Ad hoc Routing) – A hierarchical ad hoc routing protocol. CEDAR introduces the concept of core nodes that dominate a set of non-core nodes. Periodic updates are used to keep a link state table along with lists of node identification (ID) numbers current. Route searching is done reactively and returns a shortest length core path to the destination (Haas et al., 2001).

CHAMP (Caching and Multi-path Routing Protocol) – A reactive ad hoc protocol. CHAMP adds packet caching to reduce the general problem of non-deliverable packets (Valera et al., 2003).

Complementary Code Keying (CCK) – Radio frequency (RF) modulation technique used by IEEE 802.11b and supported by IEEE 802.11g (Sheu & Chen, 2002).

CSMA/CA (**Carrier Sense Multiple Access with Collision Avoidance)** – A technique for monitoring the physical interface prior to initiating a transmission to ensure the channel is not in use. If a carrier is detected, a delay is enforced before the process is repeated (Microsoft, 2002). The physical interface may be a wireline interface or a wireless interface as with 802.11.

DCF (**Distributed Coordination Function)** – Implements the 802.11 basic access methods. DCF uses CSMA/CA to provide asynchronous connectivity (Hać, 2003).

DDR (Distributed Dynamic Routing Algorithm) – Introduced in 2000 by Nikaein, Labiod, and Bonnet (2000). DDR is a hierarchical hybrid protocol that uses a tree and forest technique to define non-overlapping zones without the use of the Global Positioning System (GPS) or other positioning methods. The function of the protocol is similar to the Zone Routing Protocol (ZRP) and the Zone-based Hierarchical Link State (ZHLS) routing protocol (Nikaein et al.).

DSR (Dynamic Source Routing) – A reactive ad hoc protocol that uses an on-demand discovery technique to acquire the complete path to the destination. The newly acquired path then becomes the addressing portion of the message header so that the message can reach the receiver. DSR can generate large numbers of overhead packets during the discovery process (Prakash, 2001).

DSSS (**Direct Sequenced Spread Spectrum)** – One of two frequency distribution techniques that enables operation in the ISM radio frequencies. Originally derived from the direct satellite broadcast industry, DSSS distributes the radio signal around a central frequency using a distribution algorithm (Hać, 2003).

FANT (**forward ant)** – A biologically inspired software agent that seeks a path from a source to a specified destination. The FANT usually produces an artificial pheromone trail along the path taken. As noted, FANTs are often associated with backward ants (BANTs) (Günes et al., 2002).

Federal Communications Commission (FCC) – A federal organization created under the 1934 Communications Act to regulate and manage the radio and telecommunications industries (Newton, 2006).

FHSS (Frequency Hopping Spread Spectrum) – One of two frequency distribution techniques that enables operation in the ISM spectral frequencies. The FHSS technique transmits on a designated frequency for a short period of time before changing to another frequency within the frequency band. The pattern of changes is determined at the time communications is established between two or more nodes (Hać, 2003).

Flooding – The process of sending a packet of information to all nodes or a selected subset of nodes within a wireless network (Kargl et al., 2002).

FSR (Fisheye State Routing) – A proactive or table-based protocol that retains less specific information on routes as they extend farther away from each node FSR conserves local storage and bandwidth (Hać, 2003).

GFSK (Gaussian Frequency Shift Keying) – A modulation scheme which first filters data through a Gaussian filter before modulating with a simple frequency modulation (Hać, 2003).

GHz (Gigahertz) – Metric for one billion cycles per second (Newton, 2006).

Global Positioning System (GPS) – A system of satellites operated by the U.S. Department of Defense. A GPS provides location information through triangulation (Buszko et al., 2001).

GSR (Global State Routing) – Classified as a table-driven routing protocol GSR differs from other table-based protocols such as distributed Bellman-Ford (DBF) and link state (LS) protocols by exchanging link state information when obtaining route information. That can lead to rather large tables in larger networks. However, the GSR protocol does reduce some overhead packet traffic associated with route maintenance compared to DBF and LS protocols (Pei, Gerla, & Chen, 2000).

HARP (Hybrid Ad Hoc Routing Protocol) – A hierarchical routing protocol similar in function to the zone routing protocol (ZRP) and the zone hierarchical link state (ZHLS) protocol (Nikaein et al., 2001). The protocol restricts stored information by relying on the distributed dynamic routing protocol (DDR) to maintain topology information.

HomeRF – A short range network designed to interconnect voice and data devices in a home setting. Speed and capability are similar to Bluetooth with the addition of a good quality of service (QoS) mechanism (Wheat, Hiser, Tucker, Neely, & McCullough, 2001).

Hop - A transmission between two devices such as routers or nodes. Transmissions in ad hoc networks are measured in hops (Draves, Padhye, & Zill, 2004).

Hertz (Hz) – A designator for the measurement of frequency in cycles per second. One Hz is one cycle per second (Newton, 2006).

HSR (Hierarchical State Routing) – A hierarchical routing protocol that utilizes clustering. Networks are partitioned into clusters, each with a cluster head. The cluster head serves as a coordinator and gateway to other cluster heads (Pei & Gerla, 2001).

IEEE (Institute of Electrical and Electronic Engineers) – Largest international professional organization. IEEE has a wide number of Working Committees involved in the development of standards and technologies in a number of industries including telecommunications and computer networks.

IEEE 802.11 - Specifies network topologies (infrastructure and ad hoc), medium access control (MAC) layer protocols, and physical layer interfaces (IEEE, 1999).

IEEE 802.11a – One of several WLAN specifications defined under the IEEE 802.11 standards. 802.11a operates in the unlicensed National Information Infrastructure (UNII) band at 5.725 to 5.850 GHz and has a relatively short range (Wheat et al., 2001). 802.11a supports next-generation fixed wireless access (FWA) with optimal speeds of 100 Mbps (Littman, 2002).

IEEE 802.11b –Operates in the unlicensed ISM band at 2.4 to 2.4835 GHz and provides an optimum throughput up to 11 Mbps. (Hać, 2003).

IEEE 802.11g –Operates in the unlicensed ISM band at 2.4 to 2.4835 GHz. 802.11g provides a packet throughput that is similar to 802.11a. However, in contrast to 802.11b, 802.11g is backwards compatible with 802.11b (Vines, 2002).

Infrared (IR) – A band of electromagnetic wavelengths between .75 micrometers (µm) and 100 µm just below red in the visible spectrum (Newton, 2006). IR is limited to line-of-sight transmissions.

IrDA (**Infrared Data Association)** – A non-profit trade association that develops and promotes wireless infrared connectivity (Vines, 2002).

ISM (Industrial, Scientific and Medical) – Band of unlicensed frequencies set aside by the FCC for experimental and public use. ISM bands cover three frequency ranges, specifically, 902 to 928 MHz, 2.400 to 2.4835 GHz, and 5.725 to 5.85 GHz and are described in Part 15.247 of the FCC regulations (Newton, 2006).

Kilo (k) – One thousand in metric terms. For example, 10 kHz is ten thousand cycles per second (Newton, 2006).

Landmark – A router or switch that is a specific number of hops away from other routers designated as landmarks. The distance in hops is defined by the routing protocol used. Landmarks are useful in reducing the size of routing tables in large hierarchical networks (Haas & Pearlman, 2001).

LANMAR (Landmark Ad Hoc Routing) – A landmark scheme developed for wireline networks. Logical subnets are defined in terms of pre-selected landmark. Routes to landmarks are proactively maintained. Messages are forwarded to landmarks for redistribution (Pei, Gerla, & Hong, 2000). A drawback to this protocol is that all nodes in a subnet are expected to move as a group (Haas et al., 2002).

MAC (Medium Access Control) – A media specific protocol described in the IEEE 802 LAN standards (Hać, 2003).The MAC layer is a sub-layer of the Open Systems Interconnect (OSI) Layer 2 or data link layer (DLL) as defined by the International Standards Organization (ISO) (Shurman, Yoo, & Park, 2004).

MANET (Mobile Ad hoc NETwork) – A common acronym for mobile ad hoc networks. While this term is often used interchangeably with WLANs, a MANET is a subset of a WLAN which specifically uses wireless ad hoc features.

MANSI (Multicast for Ad hoc Network with Swarm Intelligence) – An on-demand protocol based on the ant metaphor. Designed specifically for the multicast environment, MANSI uses a core-based approach with agents developing paths and tracking link costs (Shen & Jaikaeo, 2003).

MARP (Multi-Agent Routing Protocol) – A recent addition to the list of ad hoc routing protocols. According to the developers of this protocol, agents are used to create a framework for managing multi-hop communications over networks with changing topologies (Choudhury, Paul, & Bandyopadhyay, 2004).

Megahertz (MHz) – Metric for one million hertz. For example, 1 MHz is one million hertz or cycles per second (Newton, 2006).

Monarch Initiative – A joint development project between Rice University and Carnegie Mellon University to create wireless and mobility simulation extensions for the network simulator software (*ns-2*) program. The extensions have since been incorporated into the *ns-2* releases (The CMU Monarch Project, 1999).

MPR (Multi-point Relay) – A node that is within two hops of neighboring nodes. MPR is implemented by the optimized link state routing (OLSR) protocol to reduce table size (Haas et al., 2001; Williams & Camp, 2002).

Multicast – A message sent to multiple recipients from a single source. Multicasting technology can be used by routing protocols to obtain route information by sending route requests to all nodes within receiving range. A multicast routing protocol such as MANSI can also broadcast messages to multiple recipients (Newton, 2006).

***ns-2* (network simulator-2)** – A discrete event network simulator that simulates ad hoc networks (Hu et al., 2002). The simulator can manage complex wireless scenarios and ad hoc environments.

OFDM (Orthogonal Frequency-Division Multiplexing) – An encoding technique used in WLANs and cellular telephony applications. OFDM divides signals into a number of smaller channels to enable efficient bandwidth use (Littman, 2002).

OLSR (Optimized Link State Routing) - A table-driven link state routing protocol used in ad hoc networks. OLSR employs multipoint relays (MPRs), a subset of neighbors within two hops of the sender, to send packets rather than broadcasting packets through the entire neighbor set (Haas et al., 2001).

On-demand protocols – Also referred to as reactive protocols. On-demand protocols generally obtain specific path or route information only when a message is ready to be sent (Hu et al., 2002).

OSI (Open System Interconnection) Reference Model – A seven layer network architectural model developed by the International Standards Organization (ISO). The

OSI Reference Model defines the Application, Presentation, Session, Transport, Network, Data Link and Physical Layers and describes layer functions (Goleniewski, 2002).

PCF (Point Coordination Function) – An optional basic access method that may be implemented in the distributed coordination function of the 802.11 MAC protocol. PCF provides synchronous contention-free connectivity by polling individual nodes (Hać, 2003).

PDA (Personal Digital or Data Assistant) – A class of small handheld computer-based devices. PDAs support infrared or spread spectrum connections (Wheat et al., 2001).

Piconet – A small network formed by Bluetooth devices that extends to no more than 30 feet. A piconet consists of at least one master and one or more slave nodes that share a common frequency hopping pattern (Wheat et al., 2001).

Proactive routing protocols – Also called table-based protocols. These protocols actively maintain path or route information in the form of tables. Data in tables are acquired through frequent broadcasts that solicit path information (Hu et al., 2002).

Reactive Protocols – Also called on-demand protocols. Reactive protocols generally obtain specific path or route information only when a message is ready to be sent (Hu et al., 2002).

RREP (Route REPly) – A message packet sent in response to the Route Request (RREQ) packet. The RREP contains information on the route or path to the destination requested through the RREQ (Tian, Hahner, Becker, Stepanov, & Rothermel, 2002). Both RREP and RREQ are control packets and part of the overhead in ad hoc networks.

RREQ (Route REQuest) - A packet request sent to surrounding nodes to elicit routing information. RREQ is broadcast to a wide audience using a process known as flooding (Tian et al., 2002).

Scatternet – A collection of up to 10 piconets. Each piconet must have at least one master node. A node may be a master in one piconet and a slave in another piconet (Wheat et al., 2001).

SHARP (Sharp Hybrid Adaptive Routing Protocol) – A hybrid protocol. Based on the use of a reactive protocol such as AODV, DSR, or Temporally Ordered Routing Algorithm (TORA), the SHARP protocol operates as a proactive manager (Ramasubramanian et al., 2003).

Slotted ALOHA Protocol – An improvement over the original ALOHA protocol. Slotted ALOHA uses fixed-time slots for transmission in an approach that is similar to Time Division Multiplexing (TDM). As a result, the need for contention control necessary in the original ALOHA protocol is eliminated (Naor & Levy, 2001).

SS (Spread Spectrum) – A modulation technique that distributes a radio signal over a range of frequencies. Originally developed for secure military communications, this technique is difficult to jam and prevents a single user from dominating the band (Hać, 2003).

SWAP (Shared Wireless Application Protocol) – A specification developed by the Home Radio Frequency (HomeRF) Working Group. SWAP allows various electronic devices to share voice and data in an in-home environment (Vines, 2001).

TORA (Temporally Ordered Routing Algorithm) – A distributed on-demand protocol that provides a multi-path route to the designated destination. TORA avoids the creation of request–response loops known to generate excessive traffic, thereby improving efficiency. TORA provides either reactive or proactive WLAN route maintenance (Hać, 2003).

UHF (Ultra High Frequency) - A part of the radio frequencies that ranges from 300 MHz through 3 GHz (Newton, 2006).

Unicast – Sending a message to a single recipient (Norton, 2006).

UNII (Unlicensed National Information Infrastructure) – A radio frequency band in the 5.725 to 5.850 GHz range designated by the FCC and used by IEEE 802.11a WLANs (Hać, 2003).

WHIRL (Wireless Hierarchical Routing Protocol with Group Mobility) – A hierarchical protocol introduced by Pei, Gerla, Hong, and Chaing (1999). WHIRL divides an ad hoc network into logical subnets and assigns cluster heads called home agents (HAs) for management of very large dynamic networks (Pei et al.).

WPAN (Wireless Personal Area Network) – Defined by the IEEE. A WPAN is a very small network extending no more than 30 feet (Golmie, Chevrollier, & Rebala, 2003).

WRP (Wireless Routing Protocol) – An early proactive distance vector protocol for ad hoc networks. WRP maintains extensive tables of routing information, link costs, distance in number of hops, and a message retransmission list in each node. Link cost is derived by counting the number of refresh periods required for a successful transmission. A maximum value indicates a broken link. WRP does not scale well in large networks and can also generate considerable overhead traffic (Hać, 2003).

X.25 – A packet-switched data protocol. X.25 is widely used in public and private packet switched networks to transfer data and interconnect networks (Newton, 2006).

ZHLS (Zone-based Hierarchical Link State) – A routing protocol that divides an ad hoc network into non-overlapping physical zones. The technique relies on a location-based service, such as the Global Positioning Satellite (GPS) system, to develop node associations. Intra-zone and inter-zone tables maintain routes within the node's zone, also called the inter-zone, and provide gateways to other zones (Haas et al., 2002).

ZRP (Zone Routing Protocol) – One of the first hybrid ad hoc routing protocols. ZRP was introduced in 1998 by Haas and Pearlman. ZRP integrates reactive and proactive route discovery techniques and requires a relatively small number of overhead packets to maintain route information. ZRP also has the advantage of discovering multiple routes to each destination (Haas, Pearlman, & Samar, 2003).

Summary

This chapter began with a general introduction to the field of wireless networks. The introduction covered the characteristics of ad hoc networks. In addition, the challenges and potential benefits of wireless technology were discussed.

The problem statement and goal of the research were stated. The field of wireless networking has unique requirements that cannot be accommodated by the existing single protocols developed for wireline networks. Prior researchers concluded that wireline protocols were unsuited for use in wireless networks, and wireless protocols performed well in only some wireless environments, thereby paving the way for this research (Boleng et al., 2002; Kargl et al., 2002; Viswanath & Obraczka, 2002).

The next section covered the relevance and significance of the proposed research. The research is relevant because of the considerable work that has already been done in the field. According to researchers such as Boleng et al. (2002), Marina and Das (2001), and Garcia-Luna-Aceves, Mosko, and Perkins (2003), the field of ad hoc routing protocol development remains an active area of research. As demonstrated by investigators such as Boleng et al. (2002), Denko (2003), Kargl et al. (2002), and Viswanath and Obraczka (2002), no single protocol can manage the many different scenarios presented by ad hoc environments. As a result, investigative efforts continue to improve the efficiency of wireless protocols and expand their capabilities.

The significance of this inquiry is reflected in findings of investigators such as Denko (2003), Abolhasan, Lipman, and Chicharo (2004); and Garcia-Luna-Aceves, Mosko, and Perkins (2003). Buszko et al. (2001) indicated that developing ad hoc networks can have a positive economic impact on the corporate environment. According to Hubaux et al. (2001), ad hoc networks have the potential to create a new paradigm in networking by changing the fundamental relationship between information sources and users of information. Ad hoc networks also play a critically important role in situations such as disasters. As the routing requirements in situations change, the use of multiple protocols provides a deployment advantage. The use of a wide mix of equipment with varying capabilities suggests that a number of protocols may be present. Building a basic understanding of the performance characteristics of multi-protocol networks is the first step in building more robust and flexible ad hoc networks. This inquiry provides data indicating the impact of multi-protocols on the performance of ad hoc wireless networks.

Outcomes from this investigation contribute to the body of data necessary to develop more flexible and adaptable protocols for use in heterogeneous wireless networks.

Barriers and issues to this research included the difficulty in obtaining hardware and software resources to perform the necessary tests to prove the research questions. Additional challenges centered on the researcher becoming proficient in using the *ns-2* simulator program. This included becoming adept at using the languages and accessory programs necessary to do the simulations. A significant barrier was the lack of simulators that support multi-protocol environments. Currently, the simulators available to this researcher supported only single protocol environments. This limitation required either modification to the simulation program itself or additional processing of single protocol simulation results outside the simulation environment.

The main research question was introduced. Specifically, this investigation characterized the effects on throughput and end-to-end delay in environments consisting of two different ad hoc routing protocols. Secondary questions were posed that provide the basis for future research. Limitations and delimitations were indicated. All branches of science include specialized languages and terms in order to precisely communicate information within scientific communities. The key terms presented in this dissertation are a subset of general terms used within the wireless networking community.

Chapter 2

Review of the Literature

Historical Overview

The current field of WLANs has its antecedents in early mobile packet radio experiments conducted by Abramson at the University of Hawaii in the early 1970s (Briesemeister, 2001). The outcome of these experiments led to the development of ALOHANET, the first significant network to use packet radio to connect computers on the Hawaiian Islands. ALOHANET also used satellite links to connect University of Hawaii networks to the mainland Advanced Research Projects Network (ARPANET) (Microsoft, 2002). Originally a wireline network, ARPANET was developed in the 1960s by the U.S. Department of Defense Advanced Research Programs Agency (DARPA). Initially designed to interconnect military and educational computers (Patil et al., 2003), ARPANET served as the foundation for the present-day Internet (Microsoft).

Work on radio relay in the 1970s sponsored by DARPA was directed primarily toward military applications. However, the results had wider applications (Hubaux et al., 2001). Research efforts contributed to the development of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite that serves as the framework for the present-day Internet and next-generation networks such as Internet2 (I2).

The ALOHA and slotted ALOHA wireless protocols were developed from findings in the fixed wireless ALOHANET experiments (Patil et al., 2003). The ALOHA wireless protocol approached contention and collision problems using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA/CA is currently employed in 802.11 wireless networks (Tseng et al., 2002). A variant of CSMA, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is widely used in Ethernet LANs (Newton, 2006).

Packet radio entered a new phase in 1978 when amateur radio groups in Canada and the U.S. began developing hardware and software to send and receive text messages between transceivers (Dubendorf, 2003). A small packet link-layer protocol, AX.25 was developed to enable these transmissions. AX.25 supports functions similar to the X.25 protocol used in wireline data networks (Newton, 2006). AX.25 still provides a transparent, error-free communications system for many amateur radio operators (Dubendorf).

With the development and popularization of PCs, new methods of networking were introduced and wireless transmission media including spread spectrum, microwave, satellite, and infrared were employed (Goleniewski, 2002). Spread spectrum and infrared were found to be practical in WLANs due to cost and flexibility (Hać, 2003).

Established by 160 companies in the early 1990s, the Infrared Data Association (IrDA) developed standards for infrared (IR) equipment interfaces (Vines, 2002). In 1994, the specifications for the first versions of the infrared (IR) link access protocol (IrLAP) and the IR link management protocol (IrLMP) were released (Santamaria & Lopez-Hernindez, 2001). These standards were subsequently incorporated in products such as laptop computers, printers, and PDAs, but found their most practical applications in connecting computers and printers in close proximity (Vines). Based on IrLAP and IrLMP specifications, the Infrared Communications (IrCOMM) protocol was developed which emulated serial or parallel port connections (Santamaria & Lopez-Hernindez).

In 1997, the IEEE released the initial wireless radio and infrared WLAN standards in the initial 802.11 specification (Hać, 2003). Wireless radio refers to the electromagnetic frequencies generally between 10 kHz and 3 GHz. Spread spectrum is a frequency jumping and modulation technique that is employed in 802.11 wireless implementations. Additional standards and refinements are discussed in detail in the sections that follow.

WLAN Technologies

Infrared

A WLAN technology, infrared is relatively inexpensive to implement and has a wide base of support. A line-of-sight technology, infrared operates when there are no obstructions between the transmitter and receiver or transceivers (Wheat et al., 2001). A relatively short range technology, infrared only functions reliably within a few feet and a narrow angle of transmission (Suvak, 2000). Data rates up to 16 million bits per second (Mbps) are supported but typically infrared solutions enable transmissions of 4 Mbps. Protocols for IR LANs were developed in the mid-1990s by the Infrared Data Association (IrDA) and designed mainly to support legacy systems (Vrana, 2001).

While infrared is specified as a transmission technology in the IEEE 802.11 standards for WLANs, this limited optical technology is being replaced by spread spectrum technologies that do not have the line-of-sight limitation and also provide higher bandwidth (Solis & Obraczka, 2004; Wheat et al., 2001). Infrared, nevertheless, remains a viable alternative as a replacement in applications involving connectivity between a laptop to a printer and for transporting small amounts of data between PDAs. IR technology is also used in remote control devices such as garage door openers and television remote controls (Solis & Obraczka; Suvak, 2000).

Spread Spectrum Technologies

Spread spectrum modulation technologies were originally developed by the United States military to provide secure, tamper resistant communications (Hu et al., 2002). Spread spectrum technologies such as direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS) are widely deployed today to comply with requirements set forth in the IEEE 802.11 standards for WLANs (Hać, 2003).

DSSS was commercialized and deployed in the Direct Broadcast Satellite industry (Hać, 2003). DSSS operates by distributing a signal at a given frequency across a band of frequencies where the center frequency of the band is the original frequency (Bruno et al., 2001). The frequency distribution changes over time thus making the signal appear random (Hać). As a consequence, DSSS has a high resistance to interference (Hać). According to Hać, the limited bandwidth available for use by IEEE 802.11 WLANs limits the effectiveness of DSSS making FHSS a more attractive long term solution.

FHSS operates by moving a signal within a band of frequencies based on a pseudorandom sequence of over 65,000 hops within 50 or more channels (Flickenger, 2002). Prior to transmission, the sender and receiver exchange the sequence of frequencies that the signal will use. Unlike DSSS, FHSS transmits the full signal strength on each frequency in the sequence; however, the frequency is changed multiple times a second making reception extremely difficult. As a consequence of the frequent changes in frequency, FHSS is relatively secure against unauthorized reception and interference (Hać, 2003).

Narrowband Ultra High Frequency (UHF)

Narrowband UHF data systems have been available since the early 1980s and typically operate in the 430 million hertz (MHz) to 470 MHz range (Kain, 2003). Because much of this bandwidth is controlled by the FCC, a license is required in most instances.

UHF solutions support relatively long distance communications with relatively low power, and have low to moderate data rates of 4.8 Kilobits per second (Kbps) to 19.2 Kbps. Due to the relatively directional nature and signal propagation properties of narrowband UHF, this technology is infrequently used in highly mobile environments (Kain, 2003). Currently, UHF licensed frequencies support automated data collection with higher frequencies in the 460 MHz to 470 MHz range enabling data-intensive and sophisticated communications such as video transport (Kain).

Unlicensed Radio Frequencies 900 MHz and above

FCC Part 15, Subpart C, Section 15.247, approved in 1985, allows unlicensed use of several frequency ranges in Industrial, Scientific, and Medical (ISM) bands (Flickenger, 2002). This provision also requires the use of FHSS or DSSS radio transmission technology. Spread spectrum technology allows additional users to send transmissions over the same bandwidth without significant interference compared to technologies such as IR and UHF (Hać, 2003). Of the two spread spectrum bandwidth distribution techniques, Hać indicates that FHSS is more stable and resistant to interference than DSSS. Consequently, FHSS is expected to become more widely deployed (Hać).

RF bands ranging from 902 MHz to 928 MHz and 2.4 MHz to 2.4835 GHz are available for ISM use. In addition, the 5.725 GHz to 5.850 GHz band, designated as the unlicensed national information infrastructure (UNII) band, is available (Hać, 2003).

According to Hać (2002), the 2.4 to 2.4835 GHz band is the most useful in the marketplace today. The range and capacity of the 2.4 to 2.4835 GHz band and the relatively lower cost of manufacturing devices that operate in this band contribute to the popularity of this band. In addition, this band is the only one of the three bands that is available in other countries such as Germany and Japan in addition to the U.S. These countries designate alternative uses for the 902 to 928 MHz range and 5.725 to 5.850 GHz RF bands that would require legislative changes and re-allocation of those bands. Governments in these countries independently assign RF bands for specific purposes such as for military, commercial, or educational use. As those bands are utilized, the difficulty and cost of re-allocating the RF bands becomes problematic. Those organizations such as the military that use the bands would be required to replace equipment to accommodate the changes.

The 902 to 928 MHz frequency range enables operation of devices such as wireless headsets, mobile telephones, and remote controls. By contrast, the 5.725 to 5.850

GHz frequency range has excellent data carrying capacity, but relatively poor range and higher costs than the 2.4 to 2.4835 GHz RF bands (Hać, 2003). Given this assessment by Hać, it is not surprising that the first popular home and small business WLAN technology released in the late 1990s was compliant with the IEEE 802.11b standard and operated in the 2.4 to 2.4835 GHz frequency range. In addition, newer 802.11 extensions such as 802.11g and IEEE 802.15 specification also referred to as Bluetooth, support services in the 2.4 to 2.4835 GHz frequency range as well.

IEEE 802.11 Wireless LAN (WLAN) Standards

In 1997 the IEEE 802.11 Working Group (WG) for Wireless Local Area Networks (WLANs) released the original 802.11 standard and its extensions (Ross, 2003). The 802.11 standard and its extensions cover spread spectrum and infrared transmission media and specify modulation and frequency distribution techniques (Ross). The initial data rates for 802.11 were specified as 1 Mbps and 2 Mbps with the 2 Mbps rate as an option and the 1 Mbps rate mandatory (Patil et al., 2003). 802.11 operated in the 2.4 to 2.483.5 GHz frequencies.

This release of WLAN specifications was timely for several reasons. Ohrtman and Roeder (2003) point out that with deregulation resulting from the Telecommunications Act of 1996, a new and vigorous atmosphere for the introduction of wireless media was created. The provisions of this Act opened wireline telephone and data networks to competitors and discouraged new investments in expensive wireline infrastructure (Ohrtman & Roeder). Wireless became a less expensive alternative to building out networks that were increasingly in demand.

The IEEE 802.11 suite of specifications describes standards and extensions for Medium Access Control (MAC) protocols. The MAC protocols provide either asynchronous or synchronous services, with asynchronous service mandatory and synchronous service an option (Hać, 2003). According to Hać, the asynchronous MAC basic access method implemented by the distributed coordination function (DCF) is CSMA/CA. This method tests for the presence of a carrier prior to initiating transmission on the shared frequency. If a carrier is present, transmission is delayed to avoid packet collisions. Synchronous services are provided by the point coordination function (PCF), a centralized, contention-free scheme normally implemented as a polling process (Anastasi & Lenzini, 2000).

In addition, two different network topologies are specified, namely, infrastructure-based and ad hoc (Hać, 2003). Infrastructure-based topologies consist of wireless terminals or nodes connecting through wireless access points (APs) that are linked to a wireline backbone (IEEE, 1999). The ad hoc network topology is formed without the use of APs that support connections to a wireline infrastructure. Instead, individual nodes communicate directly with other nodes, thereby eliminating the need for an established infrastructure. The 802.11 standard was followed by the release of extensions that included 802.11b and 802.11a in September, 1999. In addition, the 802.11g extension was released in 2003 and a task group was formed to develop 802.11n (Patil el al., 2003).

IEEE 802.11b

802.11b, also known as Wi-Fi for wireless fidelity, was the first wireless MAC and Physical Layer specification released by the IEEE 802.11 WG for WLANs and contributed to WLAN popularity in the public sector (IEEE, 1999b). IEEE 802.11b operates in the 2.4 to 2.4835 GHz range of the unlicensed ISM frequencies set aside by the FCC. 802.11b has an approximate operating range of 300 feet with an optimum throughput of 11 Mbps and uses the DSSS bandwidth distribution technique with complementary code keying (CCK) modulation (Ohrtman & Roeder, 2003). 802.11b devices implement dynamic rate shifting which allows lower transmission rates of 1, 2, and 5.5 Mbps in noisy conditions (Hać, 2003). Dynamic rate shifting allows devices to vary the transmission rate based on the success of prior transmissions. In keeping with the general 802.11 specifications, 802.11b operates either in infrastructure-based or ad hoc mode. Although implementations typically utilize the infrastructure-based mode, considerable interest in developing the ad hoc mode is evident (Williams & Camp, 2002).

802.11b signaling divides the available 2.4 to 2.483.5 GHz band into 14 channels each containing 22 MHz. Of the 14 channels, 11 channels partially overlap. This overlap requires built-in redundancy to accommodate data loss due to interference. The redundancy is accomplished through a technique called chipping (Ohrtman & Roeder, 2003). Chipping is the process of modulating a data stream with a second 11 bit binary sequence known as the Barker code. The encoded data objects resulting from this process are called chips (Ohrtman & Roeder).

IEEE 802.11a

The 802.11a specification was released in September 1999 at the same time as 802.11b. Unlike 802.11b which operates in the 2.4 to 2.4835 GHz range using DSSS, 802.11a operates in the UNII 5.725 to 5.850 GHz range and provides optimum rates up to

54 Mbps by using OFDM (Hać, 2003). The frequency range and modulation technique make 802.11a incompatible with 802.11b. The original implementation of 802.11a also restricted the range of transmission to about 150 feet or less in comparison to 802.11b. Like 802.11b, 802.11a has a number of fallback data rates that are used in noisy situations. The data rates of 6 Mbps, 12 Mbps, and 24 Mbps are mandatory transmission speeds while 9 Mbps, 18 Mbps, 26 Mbps, 48 Mbps, and 56 Mbps are optional (Patil et al., 2003). Patil et al. also point out that the 5.725 to 5.850 GHz range is available worldwide. As a consequence, implementations of 802.11a solutions that operate in the 2.4 to 2.483.5 GHz range make 802.11a more appealing globally.

In addition to enabling increased throughput, the 5.725 to 5.850 GHz operating frequency range allocated to 802.11a is not as heavily used by other devices. In contrast, the 2.4 to 2.4835 GHz range is heavily used by devices such as telephones and microwaves competing for the limited bandwidth (Wheat et al., 2001). 802.11a devices are more complex, expensive to produce, limited in range, and subject to signal blockage from environmental obstructions in comparison to devices using the 2.4 to 2.4835 GHz range, thereby making 802.11a deployments less attractive to the mass market (Wheat et al.).

IEEE 802.11g

Released by the IEEE 802.11 WG for WLANs in July 2003. 802.11g provides throughput similar to 802.11a while maintaining backward compatibility with 802.11b. Like 802.11b, 802.11g operates in the 2.4 to 2.483.5 GHz ISM band. 802.11g uses OFDM on the Physical Layer and either DSSS or FHSS. In addition, CCK modulation is supported (Hać, 2003). This flexibility allows backward compatibility with the popular 802.11b solution while still facilitating enhanced throughput for later generations of devices (Patil et al., 2003).

IEEE 802.11n

In September of 2003 the IEEE 802.11 Work Group formed the 802.11 Task Group N (TGn) to address the increasing demand for bandwidth in wireless applications (Gilbert, Choi, & Sun, 2005). The goal of the 802.11 TGn was to produce a new extension to the 802.11 wireless standard, designated 802.11n, that is capable of wireless data speeds greater than 100 Million bits per second (Mbps). While the 802.11n extension is intended to eventually replace the 802.11a/b/g extensions, one goal of the new extension is to maintain some backward compatibility with the current 802.11a/b/g standards (Gast, 2005).

The proposed 802.11n extension achieves the desired throughput by using multiple antenna arrays and receivers through a technology called multiple-input multiple-output (MIMO). MIMO creates multiple streams of data that are transmitted on an array of from one to four separate antennas. The data are then received on a like number of antennas and recombined at the receiving station producing throughputs as high as 600 Mbps according to the recently approved draft specification (Gast, 2005; IEEE, 2006). The current IEEE 802.11n draft standard provides throughput of 300 Mbps for two antenna systems and 600 Mbps for four antenna systems (IEEE, 2006).

The 802.11n draft standard was approved in January of 2006 and was submitted for final ratification. The draft version of the 802.11n extension approved combines two opposing technological approaches to achieve the goals of high data throughput and backward compatibility with the existing 802.11a/b/g extensions (IEEE, 2006).

IEEE 802.15 – Bluetooth

Bluetooth is a short range wireless technology that connects a variety of electronic devices together (Golmie, Van Dyck, Soltanian, Tonnerre, & Rebala, 2003). Bluetooth is defined in the IEEE 802.15 standard and operates in the 2.4 to 2.4835 GHz ISM band like 802.11b and 802.11g solutions. However, in contrast to 802.11b and 802.11g, Bluetooth functions at a very low power with a limited range of 30 feet (Golmie, Van Dyck, et al.). The technology operates at a 1 Mbps channel rate using a FHSS scheme that changes on a packet-by-packet basis (Soltanian & Van Dyck, 2001). Bluetooth uses Gaussian frequency shift keying (GFSK) as its modulation technique and a set of 79 frequencies or channels in normal operations (Peterson, Baldwin, & Raines, 2003).

The small networks formed by Bluetooth devices are called wireless personal area networks (WPANs). WPANs are distinguished from 802.11b networks by their small size and limited scope (Golmie et al., 2003). Bluetooth-based WPANs also support small ad hoc networks that in turn act as nodes on 802.11-based WLANs (Deb, Freburg, Surdu, Hall, & Maymi, 2002).

Bluetooth ad hoc peer-to-peer networks operations are based on proximity (Deb et al., 2002). The network topology is called a piconet with one Bluetooth device acting as a master node while the others act as slave nodes to the master node (Deb et al.). The master node provides the communications synchronization and frequency hopping pattern to the slave nodes. When more than one master node is within communications range of another, a scatternet is formed. However, the piconets that form the scatternet can retain their independence and separation by maintaining separate synchronization with their respective master nodes (Deb et al.). This capability allows multiple piconets to exist within the same scatternet space.

IEEE 802.16 - WiMAX

The IEEE 802.16 family of standards defines the features of a wireless metropolitan area network (WMAN). The characteristics of a WMAN include high-speed data transfer over fixed and mobile wireless links that provide an alternative to current wired broadband options such as digital subscriber line (DSL), optical fiber, and coaxial cable (Ghosh, Wolter, Andrews, & Chen, 2005). The current 802.16d specifications support the MIMO model used in the 802.11n extensions. According to Gilbert, Choi, and Sun (2005), MIMO systems provide greater spatial diversity than single antenna systems. Spatial diversity refers to the probability that as the number of antennas used increases, the number of antennas in poor receiving positions decreases, thereby improving overall performance (Gilbert et al., 2005). Multiple antennas allow for multiple data streams, thereby increasing the throughput as compared to single antenna systems.

Unlike the 802.11 WLAN standard and its extensions, 802.16 originally specified operations in the 10 to 66 GHz licensed areas of the radio spectrum. As the standard evolved however, the 2 to 10 GHz range became increasingly attractive. This was largely the result of the line-of-sight issues present in the 10 to 66 GHz range (IEEE, 2004). Current proposals include operation in both licensed and unlicensed spectrum. The unlicensed 5.725 to 5.850 GHz spectrum is the same spectrum that is currently used by 802.11a/g standards (Ghosh et al., 2005).

The 802.16e extension is currently under review by the development community and provides support for wireless mobility at speeds of 70 to 80 miles per hour. In addition, an asymmetrical wireless link structure is defined that will enable personal devices such as PDAs, cellular telephones, and laptop computers to access high-speed data links (Ghosh et al., 2005).

High Performance Radio Local Area Network (HIPERLAN)

The HIPERLAN specification was developed by the European Telecommunications Standards Institute (ETSI) Radio Equipment and Systems (RES)-10 Group (Hać, 2003). Four types of HIPERLANs were defined. HIPERLAN Types 1 and 2 most closely approximate the 802.11 WLAN standards (Wheat et al., 2001). HIPERLAN Type 1 (HIPERLAN/1), released in 1996 and also known as Wireless 8802, operates in the 5.725 to 5.850 GHz RF band and enables a throughput from 20 Mbps to 23 Mbps (Hać, 2003). HIPERLAN Type 2 (HIPERLAN/2) was released in 2000. HIPERLAN/2 operates in the same 5.725 to 5.850 GHz RF band as HIPERLAN/1, but supports data rates reaching 54 Mbps (Chandramouli, 2002; Hać; Vasilakopoulou, Karastergios, & Papadopoulos, 2003).

HIPERLAN/2 interoperates with asynchronous transfer mode (ATM), Ethernet, third generation global system for mobile communications (3GSM), Internet Protocol (IP), and other current technologies, thus providing a high degree of flexibility (Littman, 2002). In contrast to the IEEE 802.11a CSMA/CD mode of operation, HIPERLAN/2 operates in time-division multiplexing (TDM) mode (Littman). TDM avoids the delays inherent in CSMA/CD and allows HIPERLAN/2 to support connection-oriented transmissions providing enhanced QoS services (Bolinth et al., 2001; Wheat et al., 2001). In contrast to 802.11a, HIPERLAN/2 allows each connection to be assigned specific QoS parameters including delay, jitter, and bit error rate limits (Wang, Khokhar, & Garg, 2002). Flexibility is accomplished through operating multiple modes at the Physical Layer or Layer 1 of the OSI Reference Model (Wang et al., 2002). The Physical Layer operating modes allow any of four different modulation techniques to be used on each of the 52 available sub-channels.

Home Radio Frequency (HomeRF)

HomeRF is an open industry specification developed by the Home Radio Frequency Working Group (Chandramouli, 2002). This specification defines how various electronic devices within a home share voice, data, and other information. Like the IEEE 802.11b and 802.11g standards, HomeRF operates in the 2.4 to 2.4835 GHz ISM band. In contrast to IEEE 802.11b, HomeRF supports a theoretical data rate of only 1.6 Mbps with a maximum practical throughput of 650 Kbps (Wheat et al., 2001).

HomeRF transmissions are restricted to a relatively short range of 150 feet. As a consequence, HomeRF is comparable to Bluetooth (Wheat et al., 2001). Version 2 of HomeRF, released in 2001, enables data rates of 10 Mbps, thereby making this technology similar to IEEE 802.11b, but with the advantage of built-in quality of service (QoS) (Chandramouli, 2002). The HomeRF WG developed shared wireless access protocol (SWAP) to provide high-quality connectivity between voice and data devices in a home setting before disbanding in January 2003.

WLANs and ad hoc routing protocols

This investigation focuses on the 802.11 WLAN standards that cover ad hoc networks. Major areas of research associated with ad hoc networks include ad hoc routing protocols, agent technology, and network simulation. These domains are discussed in this section and the sections that follow. Initially, wireless medium access (MAC) protocols are described.

The real push for the popularization of WLAN technology began with the establishment of the IEEE 802.11 WGs in the early 1990s. Adler and Scheidler (1998) discussed the problems of contention, security, and energy consumption in their landmark presentation at the ACM Symposium on Parallel Algorithms and Architectures in Puerto Vallerta, Mexico. According to Adler and Scheidler, contention in ad hoc networks occurred when nearby nodes attempt to transmit at the same time. Nodes checked for active transmissions before beginning their transmissions using CDMA/CA.

With the release of the 802.11b and 802.11a extensions in 1999 and the proliferation of wireless-enabled notebook computers and PDAs, demand for wireless connections increased (Papapetrou & Pavlidou, 2003; Patil et al., 2003). As noted, 802.11 standards specify both infrastructure-based and infrastructure-less or ad hoc topologies (IEEE, 1999). Infrastructure-based WLANs are widely deployed. The accelerating interest in ad hoc or self-organizing, wireless networks that lack a permanent infrastructure is reflected in the number of research efforts underway (Li & Cuthbert, 2004; Papapetrou & Pavlidou; Solis & Obraczka, 2005).

As a consequence of the relatively new advancements in the mobile ad hoc networking field, researchers initially examined existing routing technologies and found them wanting (Abolhasan, et al., 2004; Baran & Sosa, 2001). Protocols and methodologies for establishing connectivity and determining paths, routing, and network control were typically developed for cellular devices and spread spectrum and wireline networks. Generally, these technologies utilized fixed hosts communicating with mobile nodes (Gast, 2005).

In contrast, ad hoc networks operate without fixed locations and dedicated hosts (Williams & Camp, 2002). Rather, each node must act as a host and contain the routing software necessary to discover and communicate with nearby nodes (Hubaux et al., 2001). According to Hać (2003), routing protocols designed for wireline networks cannot be used for ad hoc networks because of node mobility in ad hoc networks. Investigators in the relatively established industries of telecommunications and IT, including the mobile radio and cellular telephony sectors, continue to investigate strategies for implementing ad hoc networks (Huang, Lee, & Tseng, 2004).

However, some transmissions may go undetected (Hać, 2003) as a result of a hidden terminal problem (Tseng et al., 2002). Hidden terminal problems occur when a node attempts to transmit a message to another node that is already receiving a message. Normally, prior to transmitting, a node checks for a carrier indicating the channel is in use before starting a transmission. In some cases, another transmitting node located outside the receiving range goes undetected. The intended receiver located between the two transmitting nodes receives both messages now damaged. Adler and Scheidler (1998) and Wedde et al. (2005) examined the feasibility of using transmission power control as a means to reduce contention and energy consumption while improving security. Poon and Li (2003) proposed a method of reducing power requirements in ad hoc networks while dynamically adjusting transmission range.

Mobile Ad Hoc Routing Protocol Overview

Mobile ad hoc routing protocols are classified in a number of different ways. One popular classification method divides protocols into three general categories, specifically, proactive, reactive, and hybrid (Zhang & Jacob, 2003). Proactive protocols actively seek and maintain path information for surrounding nodes in tables. Nodes employing reactive protocols seek path information only when they are ready to send a message (Lee et al., 2002). Hybrid protocols combine aspects of both proactive and reactive protocols.

Lee et al. (2002) also point out that ad hoc routing protocols may be classified as either unicast or multicast. Unicast protocols transmit message packets to a single recipient at a time while multicast protocols transmit a single message to an authorized group of recipients.

Ad hoc routing protocols are classified as single hop or multi-hop. A hop is defined as a transmission between adjacent nodes (Newton, 2006). Multi-hop refers to transmissions that travel through intermediary nodes to reach a destination (Newton). Multi-hop protocols are designed to store path information for intermediary nodes to facilitate transmission of messages destined to nodes that are not adjacent or are outside the range of the originating node (Tseng et al., 2002). Single hop protocols are designed to transfer messages only to adjacent nodes and store path information only to adjacent nodes.

Protocols are also classified as position-based versus topology-based, link state versus distance vector, flat structure versus hierarchical structure, decentralized computation versus distributed computation, source routing versus hop-by-hop routing, and single path versus multiple path (Zou, Ramamurthy, & Magliveras, 2002; Valera et al., 2003; Prakash, 2001; Williams & Camp, 2002; Mauve, Widmer, & Hartenstein, 2001). An overview of the distinctive features of protocols included in these classification categories is presented in this segment.

Proactive or table-driven protocols that are well-suited for wireline environments do not perform well in ad hoc wireless environments (Hu et al., 2002; Valera et al., 2003). In wireless ad hoc networks, changing topography and node motion dramatically increase the table storage requirement as well as the bandwidth necessary to maintain information about routes (Marwaha et al., 2001). In wireline networks, there is no movement and infrequent changes in equipment. Consequently, route tables do not change rapidly. In addition, the equipment attached to wireline networks usually contains more storage and computational power than wireless networks. In contrast to wireless network elements, wireline network elements are also less subject to the power limitations of battery operation. As a result, the bandwidth and power required for constantly maintaining the routing tables for a proactive routing protocol in a wireless ad hoc network may be prohibitive (Haas et al., 2002). Roy and Garcia-Luna-Aceves (2002) also point out that proactive routing protocols do not scale well due to the potentially large tables required for large ad hoc networks.

Reactive or on-demand protocols do not proactively maintain a table of routes but determine the route to the receiver only when a message is ready to send. Consequently, these protocols tend to be more efficient in dynamic or rapidly changing ad hoc networks (Hu et al., 2002). As reactive routing protocols must determine the path to the intended recipient when a message is ready, route acquisition time can create significant negative impacts on the delay in transmitting the message (Tian et al., 2002). In multi-hop environments, delay is increased as route acquisition time is lengthened. The large end-to-end delays that can result with the use of reactive protocols make them unsuitable for real-time or time-sensitive applications (Marwaha et al., 2001).

Hybrid protocols resolve some problems associated with purely reactive or proactive protocols by adopting features of each. For example, in contrast to proactive protocols, a hybrid protocol with tables may find updates only on an as-needed basis. Examples of hybrid routing protocols are also described later in this section.

Proactive or Table-based Routing Protocols

Perkins and Bhagwat (1994) in their seminal article first proposed the destination sequenced distance vector (DSDV) routing protocol. Based on the conventional routing information protocol (RIP), DSDV was modified specifically for use in wireless ad hoc networks. This protocol utilized tables maintained in each node in the network and required active bidirectional links (Hać, 2003). In their classic work, Perkins and Bhagwat (1994) discussed the difficulties imposed by changing topology in mobile ad hoc networks as defined by the IEEE 802.11 WG for WLANs. At that time, the authors acknowledged that methods for enabling wireless ad hoc connections between mobile computers were not available. Available routing protocols were designed for static infrastructure-supported networks and placed too high a computational burden on existing mobile computers. The lack of computational and storage capacity remains a problem today especially with smaller PDAs (Hubaux et al., 2001). New generations of mobile equipment supplied with the necessary storage and computational power will be able to perform better in ad hoc networks provided that appropriate routing protocols exist (Hu et al., 2002; Kanter, 2003).

In 1995, another table-based distance vector protocol, the wireless routing protocol (WRP) was introduced by Murthy and Garcia-Luna-Aceves (Hać, 2003). Like the DSDV protocol, WRP maintained a distance vector table and a route table as well as a link-cost and a message retransmission table (Raju & Garcia-Luna-Aceves, 2000). According to the authors, the additional tables assisted in reducing misdirected and lost messages by maintaining information on unresponsive nodes and the number of timeouts between successful transmissions between nodes (Raju & Garcia-Luna-Aceves). Nodes utilize regular circulated messages between neighboring nodes to maintain fresh information in the stored tables (Hać).

In 1998, Chen and Gerla introduced global state routing (GSR), an improved version of WRP (Haas et al., 2002). Like WRP, GSR maintains a number of tables including link state tables and relies on flooding to continually maintain current information on surrounding nodes and extended paths (Choudhury et al., 2002). Flooding results in excessive management traffic and expansive table size especially in larger networks (Hać, 2003).

Fisheye state routing (FSR), an adaptation of GSR, limits the scope of the network traffic and table size by restricting updates to a limited number of hops from the table source (Pei, Gerla, & Chen, 2000). Accurate information is obtained for the nodes closest to the source while distant nodes store more general and less accurate information (Haas et al., 2002).

Optimized link state routing (OLSR) is a link state protocol that limits the scope of information that is collected and disseminated (Clausen & Jacquet, 2003). OLSR uses an efficient flooding technique to obtain link information from nodes within a two-hop distance unlike conventional link state protocols that propagate link information throughout the entire network (Haas et al., 2002). The two-hop subset of nodes surrounding a specific node is referred to as its multipoint relay (MPR) set. Typically, only those nodes within a MPR set participate in relaying messages, thereby eliminating the need for extensive tables of link data (Haas et al.). Development of the OLSR protocol continues. The hierarchical OLSR (HOLSR) protocol is a modified version of OLSR protocol designed to improve scalability in larger heterogeneous MANETs (Ge, Lamont, & Villasenor, 2005). In simulations the HOLSR protocol significantly reduced protocol overhead and improved scalability in large networks with high speed links such as those found in military environments (Ge et al.).

Hierarchical Routing Protocols

Hierarchical protocols are a subset of table-based protocols. They provide another method for reducing overall management traffic overhead and table size by forming clusters of nodes (Hać, 2003). The hierarchical state routing (HSR) protocol introduced by Iwata, Chaing, Pei, Gerla, and Chen (1999) partitions the ad hoc network into clusters. Each cluster has a specialized node called a cluster head that serves as a gateway and collection point (Hać). In this approach, the assigned cluster heads communicate frequently with the nodes within their assigned clusters. Cluster heads then exchange periodic summarized updates with surrounding cluster heads to maintain an accurate view of the overall network (Chaing et al.). Clustering and hierarchical schemes support network scalability and efficient multicast delivery (Kwon & Gerla, 2002). Kwon and Gerla also point out that active clustering requires significant amounts of cluster-dependent traffic. This process will not work properly with partial neighbor information, a situation that is common in highly mobile networks.

Kwon and Gerla (2002) promote the use of passive clustering to overcome active clustering drawbacks such as the use of specific control packets and separate clustering schemes. Rather than gaining information on the surrounding nodes through repeated exchange of information, passive clustering acquires necessary data by monitoring normal traffic between nodes (Kwon & Gerla).

According to Pei, Gerla, Hong, and Chaing (1999), the wireless hierarchical routing protocol with group mobility (WHIRL) can enable operations in large ad hoc environments. WHIRL divides the network into logical subnets. Each subnet has a Home Agent (HA). Communications from nodes within a subnet are transferred to the HA prior to being routed to the cluster head of the final destination. The cluster head then routes the message to the final node within its subnet. Experimental results show improvements with WHIRL in comparison to other hierarchical methods in reducing overhead traffic and maintaining low latency (Pei et al.). Pei at al. also indicated that WHIRL had some disadvantages in comparison to reactive or on-demand routing protocols. These limitations included dropped packets when routes become invalid as a consequence of mobility and increased complexity due to the addition of the HA (Pei et al.).

Additional examples of hierarchical routing protocols include core extraction distributed ad hoc routing (CEDAR), zone-based hierarchical link state (ZHLS), and landmark ad hoc routing (LANMAR) (Haas et al., 2002). Introduced in 1999, CEDAR employs core nodes. Each core node is within one hop of its neighbors (Sivakumar, Sinha, & Bharghavan, 1999). Link state information travels inward to the core node which stores data on stable, high-capacity, and nearby links (Haas et al.). Global route searching is done reactively. Core nodes, also called dominators, determine the shortest path to the receiving core node (Haas et al.).

Zone-based hierarchical link state (ZHLS) routing divides a network into a series of non-overlapping zones (Haas et al., 2002). This protocol also introduces the use of a global positioning system (GPS) to maintain information regarding a node's physical location as well as its logical location within a zone (Hać, 2003). Zones can be further divided into sub-zones. Every node maintains both intra-zone and inter-zone routing tables. Intra-zone tables enable communications to other nodes within the sender's zone while the inter-zone tables serve as gateways for communications outside the immediate zone (Haas et al.). In effect, the intra-zone table defines the detailed topology of a node's position within its zone while the inter-zone table provides an overview of the zone's position within the network (Hać).

Landmark ad hoc routing (LANMAR) is an adaptation of a wireline protocol first introduced by Tsuchiya in 1988 (Haas et al., 2002). Like the original, LANMAR creates a network of pre-defined logical subnets, each with a pre-selected landmark (Pei et al., 2000). This protocol actively develops and maintains routes to landmarks through the exchange of distance vectors among the nodes. Message packets specify the hierarchical address. These packets are sent to the landmark for distribution within their subnet (Haas et al.), thereby reducing the amount of overhead required to maintain accurate routing tables.

The hybrid ad hoc routing protocol (HARP) is a hybrid hierarchical protocol that is operationally very similar to the ZRP and the ZHLS routing protocols (Nikaein et al., 2001). HARP differs from both ZRP and ZHLS by restricting the amount of routing information stored to data about the path between the source node and the destination node and leaving topology definition to the distributed dynamic routing (DDR) protocol (Nikaein et al.). HARP reduces the need for widespread control packet flooding and bandwidth by restricting communications to a subset of the forwarding nodes in each zone.

On-demand or Reactive Routing Protocols

On-demand routing protocols are a recent class of routing protocols that provide a method for dealing with scalability problems in wireless ad hoc networks (Hong, Xu, & Gerla, 2002). A key advantage of on-demand routing protocols is that they do not actively maintain tables of routes to surrounding nodes. As noted, table size becomes unmanageable in large networks. Instead, these protocols use a route discovery method to determine the path at the time it is required (Hać, 2003). Dynamic source routing (DSR) is an on-demand routing protocol that allows an ad hoc network to be completely self-organizing and self-configuring, thus eliminating the need for any infrastructure or administration (Johnson, Maltz, Hu, & Jetcheva, 2002). Each message packet transferred by the DSR protocol contains addressing information in the message header to enable successful information transmission (Hać). As a consequence, intermediary nodes are not required to maintain up-to-date routing tables.

On-demand routing protocols consist of functional components for routing, route discovery, and route maintenance (Hać, 2003). Unlike proactive protocols with predetermined route tables, on-demand protocols determine the appropriate path only when necessary. This approach can lead to significant problems such as long delays in determining routes and excessive control packet traffic reducing the effective throughput of messages (Marina & Das, 2001).

Ad hoc on-demand distance vector routing (AODV) improves on DSDV by eliminating proactively maintained route tables based on repeated update requests (Hać, 2003). Instead, tables are created on-demand, thereby lowering the overhead of proactive route requests (Ye et al., 2003). Paths are defined with sequence numbers as in DSDV. However, the information is gathered on-demand through a process of flooding the network with route request (RREQ) messages. An intermediate or destination node stores the destination address in its local cache, responds with route reply (RREP) to the originator and provides the route information (Ye et al.).

The temporally ordered routing algorithm (TORA) protocol is an example of a distributed on-demand protocol that provides loop-free multi-path routing with reactive or proactive route establishment and maintenance (Park & Corson, 2001). The TORA protocol requires only information about adjacent routers. While maintaining path state information on a per-destination basis, the TORA protocol does not actively maintain shortest-distance metrics for use in establishing routes (Hać, 2003). Hać describes the protocol as a highly adaptive and scalable protocol. The TORA protocol also supports link reversal by providing feedback to the routing source on paths that become blocked or congested, thereby redirecting messages to alternate paths (Park & Corson). The TORA protocol is not necessarily efficient however. According to Ye et al. (2003), at least one study showed the TORA protocol generated 50 times the overhead of the AODV and DSR protocols, thereby raising concerns about TORA's effectiveness.

A recent entry into the reactive protocol field, the caching and multi-path (CHAMP) routing protocol creates a five packet cache in each node that is designed to reduce packets dropped as a result of path failure (Valera et al., 2003). In other reactive protocols such as DSR and AODV, these packets are normally discarded. The CHAMP protocol temporarily stores forwarded packets when a transfer error is encountered and forwards the packet along an alternate route. With the CHAMP protocol, multiple routes to each destination must be maintained (Valera et al.). Valera et al. demonstrated in a series of simulations in stressful environments that the CHAMP protocol supports significant gains in performance in contrast to DSR and AODV.

Hybrid Mobile Ad Hoc Routing Protocols

Introduced by Haas (1997), ZRP is a representative example of a hybrid ad hoc routing protocol. In an effort to overcome protocol limitations including poor scalability and management of dynamic network architecture, Haas created a new single hybrid protocol. ZRP integrates proactive and reactive elements of route discovery and maintenance, thereby combining the best features of distance vector and on-demand type protocols. Based on continued efforts at developing a converged routing protocol, Haas illustrates the need for a multi-protocol solution.

The geographic distance routing (GEDIR), the most forward with fixed radius (MFR), and the dynamic incremental routing (DIR) protocols combine positioned-based routing with loop free single paths and provide guaranteed delivery (Rangarajan & Garcia-Luna-Aceves, 2004; Stojmenovic & Lin, 2001; Woo & Singh, 2001). Guaranteed delivery is dependent on a collision-free and connected environment which is unlikely in any real ad hoc situation. These protocols depend on GPS to provide location information and use minimal hop counts. One advantage of the GEDIR, MFR, and DIR protocols is their reduction of overhead in large networks (Stojmenovic & Lin).

Ramasubramanian et al. (2003) developed a unique approach to hybrid protocols by using two separate protocols. One protocol consists of an existing on-demand protocol. The second protocol is a table-based proactive protocol called the sharp hybrid adaptive routing protocol (SHARP). With SHARP, an individual can select between reactive or proactive protocols based on application requirements (Ramasubramanian et al.).

Agent-based Routing Protocols

While still relatively new, agent-based routing protocols are popular research topic in the wireless ad hoc networking community. As an example, the multi-agent routing protocol (MARP) introduced in 2004 features an amalgam of several protocols and a mechanism for creating a topology-aware environment (Choudhury et al., 2004).

MARP can significantly reduce overhead associated with route maintenance in situations where the topology is rapidly changing.

According to Denko, (2003), mobile agents can perform clustering and maintain routing information in large hierarchical ad hoc networks. In situations with frequent changes in clusters, traditional methods of maintaining cluster information can result in high overhead. Moreover, mobile agents can significantly reduce bandwidth utilization and communications latency and minimize connection time (Denko). The architecture for mobile agents proposed by Denko includes two distinct agents, specifically, the routing mobile agent (RMA) and a clustering mobile agent (CMA). CMAs maintain cluster tables through the use of broadcast hello messages. RMAs maintain intra-cluster and inter-cluster route tables to known destinations.

Migas, Buchanan, and McArtney (2003) of the School of Computing at Napier University in Scotland investigate the use of mobile agents for routing, topology discovery, and automatic network configuration. According to these researchers, agents can solve numerous problems in ad hoc networks. Their research goals include maximizing ad hoc network performance, scalability, and reliability and reducing latencies.

Ant colony dynamics applied to routing is promoted in the ant colony routing algorithm (ARA) by Günes et al. (2002). According to Günes et al., no routing algorithm fits all cases in ad hoc networks. As a consequence, Günes et al. propose an algorithm based on swarm intelligence patterned after an ant colony. Günes et al. define two types of protocols, specifically, forward ants (FANTs) and backward ants (BANTs).

The FANT protocol supports the discovery of paths to the target node. In this process, the FANT creates an artificial pheromone trail along the path. Additional traffic increases the pheromone trail incrementally over a route. The same pheromone value is decreased over time just as the pheromone trail dissipates through disuse. ARA enables multiple path routing, features low overhead, is loop free, and facilitates on-demand operations (Günes et al.). In simulations performed by Günes et al., ARA performed well in comparison to DSR. In highly dynamic environments, DSR retained a slight advantage. ARA performance was clearly superior to AODV and DSDV in highly mobile environments.

An ant-based routing solution that consists of the topology abstracting protocol (TAP) and the mobile ants based routing (MABR) protocol is also in development (Heissenbüttel & Braun, 2003). TAP creates a two-layered hierarchical structure and logical routers in the upper layer. Each node falls within the domain of a logical router that is itself a node. Logical routers maintain two tables. The MABR protocol uses FANTs and BANTs as in the ARA protocol. As with ARA, the ants create an artificial pheromone trail to re-enforce well traveled paths. Once paths are determined, a technique called straight packet forwarding (SPF) facilitates message delivery (Heissenbüttel & Braun).

FANTs and BANTs are also used for supporting robust multicast routing (Shen & Jaikaeo, 2003). Shen and Jaikaeo utilized the swarm intelligence of the ants metaphor to develop the MANSI, (on-demand multicast for ad hoc network with swarm intelligence) protocol. The MANSI protocol creates multicast connections within defined groups using a forwarding set that consists of an intermediate set of nodes capable of facilitating communications in core-based hierarchies. Ant-like agents assist in the development and evolution of the forwarding set to identify economical communications paths. Cost is determined by measuring the number of hops and the delay in the paths to the nodes within the set. Paths with lower delay or fewer hops are lower cost routes. Lower cost routes are preferred for efficient transmissions. As with other swarm intelligence protocols, both FANTs and BANTs are employed (Shen & Jaikaeo).

Mobile Ad Hoc Protocol Performance

Boukerche (2004) described the use of simulation to determine the effectiveness of the DSDV, DSR, and AODV protocols. According to Boukerche, each protocol showed strengths and weaknesses. For example, Boukerche indicated the AODV protocol performed poorly when nodes were moving quickly but had very low throughput delay in more static environments. In contrast, DSR performed well in mobile environments but at the expense of high packet overhead. According to Boukerche, protocols such as these did not function optimally in multiple situations.

Lee et al. (2000) compared the performance of the ad hoc multicast routing (AMRoute) protocol, the on-demand multicast routing protocol (ODMRP), the ad hoc multicast protocol utilizing increasing identity numbers (AMRIS), and the core-assisted mesh protocol (CAMP). Basing their results on packet delivery ratio, or the number of packets sent divided by the number of packets delivered and the control bytes to transmitted bytes delivered ratio or the number of control and data packets transmitted divided by the total number of packets delivered, the authors concluded that in general mesh networks performed best in highly mobile environments. Lee et al. concluded that the ODMPR protocol performed best overall and suggested the modification of this protocol to improve performance in environments with high numbers of transmissions.

Prakash (2001) described the capabilities of the DSDV, DSR, AODV, and TORA protocols in supporting ad hoc networks with unidirectional links. Prakash found significant performance differences in tests of the aforementioned protocols with no single protocol performing well in multiple environments. A modified version of the DSDV protocol was recommended since this protocol performed well in a highly mobile environment with a strongly connected network. However, an efficient MAC sub-layer protocol was required for the modified protocol to operate effectively (Prakash). Prakash also recommended additional research to reduce increased overhead incurred in protocol modifications.

Mauve et al. (2001) presented a study of the performance of position-based routing protocols. These investigators indicated that position-based algorithms can provide performance improvements over topology-based algorithms by using a location service (LS) to define node position within a network. Mauve et al. examined different packet forwarding approaches and obtained contrasting results in the greedy perimeter stateless routing (GPRS), the distance routing effect algorithm for mobility (DREAM), and the location aided routing (LAR) protocols. Results demonstrated that GPRS may drop packets in large networks. In contrast, DREAM provided reliable transmission of small numbers of packets. They concluded none of the protocols examined resolved the problems inherent in MANET implementations (Mauve et al.).

Abolhasan and Wysocki (2003) compared the performance of the dynamic zone topology routing (DZTR) protocol to the AODV, LAR1, and LPAR protocols. The researchers noted that the DZTR protocol is advantageous over other zone-based protocols since the DTZR protocol creates dynamic rather than static zones, reduces information redundancy and potential single point failures, and utilizes numerous different location strategies to minimize overhead. Abolhasan and Wysocki concluded that under worst case conditions, the DZTR protocol produced fewer control packets and better packet delivery than the AODV, LAR1, and LPAR solutions. In light to moderate network conditions, the same protocols may produce better results than DZTR. These outcomes also indicate the inability of a single routing protocol to perform effectively in all MANETs.

Kassabalidis et al. (2001) discussed difficulties in developing new protocols to accommodate complexities in ad hoc networks. According to Kassabalidis et al., traditional static and dynamic routing protocols were unable to manage large-scale network operations, rapidly changing topology, or unstable linkages. Despite a concerted effort to develop new protocols to manage these problems, Kassabalidis et al. concluded that resolving the array of inherent problems in MANETs appeared unlikely with any single routing protocol solution. Based on their work, Kassabalidis et al. determined that an adaptive protocol is necessary to provide the greatest range of potential solutions. Kassabalidis et al. recommended the use of mobile agents belonging to the relatively new branch of artificial intelligence (AI) research called swarm intelligence.

Swarm intelligence is patterned after the behavior of social insects such as bees and ants (Kassabalidis et al., 2001; Sugar & Imre, 2001). Each individual insect is limited in intellectual scope and ability. However, when functioning as a swarm, social insects accomplish complex tasks (Arabshahi et al., 2001). In a similar manner, mobile software agents with limited individual capabilities can collaborate to manage more complex tasks than can be accomplished by any single software agent acting alone.

Software agents

Software agents are typically small mobile software programs that are environmentally aware and goal-oriented (Wooldridge, 2002; Kawaguchi & Inagaki, 2000; Denko, 2003). Mobility implies that the software agent can replicate or move itself from host-to-host (Kotz, Gray, & Rus, 2002). Some agents are designed with environmental awareness and the ability to make decisions or perform tasks based on pre-established rules. These qualities provide the flexibility necessary to cope with the highly dynamic environments in which ad hoc networks are formed. Scarce bandwidth, restricted battery power, and limited computing capacities are additional problems that must be addressed through proper adaptive selection of routing protocols (Sugar & Imre, 2001). Mobile software agents provide a vehicle to support an adaptive solution to these problems. By contrast, intelligent software agents are mobile, learn from the environment, and change behavior based on experience. The software agents proposed in this research are mobile, but not intelligent.

Mobile software agents are one method that can be used to address the highly variable performance of ad hoc routing protocols due to environmental factors (Kawaguchi et al., 2000). These agents are also proposed to support route discovery (Royer, Sun, & Perkins, 2001) and enable message carrier functions (Kawaguchi & Inagaki, 2000; Li & Rus, 2000). For example, Marwaha et al. (2002) combined agents and the AODV protocol to reduce end-to-end delay and route discovery latency in ad hoc networks. Kassabalidis et al. (2001) used mobile swarm-based agents to address scalability problems in ad hoc networks. Agents were also adapted to improve efficiency in slow and unreliable networks (Chacón et al., 2000).

Swarm-based agents, also called biologically inspired agents, follow the model of swarming insects such as ants (Baran & Sosa, 2001). There is considerable interest in swam intelligence within the ad hoc wireless research community (Arabshahi et al., 2001). One of the first biologically-based agent techniques applied to ad hoc networks was AntNet described by Di Caro and Dorigo (1998). These authors introduced the concept of stigmergy or indirect communications by mobile software agents in facilitating ad hoc routing. Other researchers that used ant-like agents and pheromone trails for position-based routing included Camara and Loureiro (2000), Jiang and Camp (2002), and Sugar and Imre (2001). These investigators examined mobile software agent capabilities in supporting location services for position-based routing operations.

Arabshahi et al. (2001) examined the use of intelligent software agents for adaptive routing in wireless networks and focused their research on using agents to detect and respond to dynamic traffic impacting events in order to maintain a specified quality of service (QoS). Arabshahi et al. concluded through their examination of the literature that swarm-intelligent routing can enhance MANET reliability and the effectiveness of data transfer in multi-node wireless networks. In addition, Arabshahi et al. determined that overhead resulting from increased network size, a problem associated with the use of table-based routing protocols, can be reduced through the use of intelligent agents.

Swarm-based routing is popular in developing wireless routing solutions for a number of reasons. First, biological systems represented by ants and bees produce very sophisticated behaviors based on a relatively small set of simple actors (Arabshahi et al., 2001). In terms of routing, ants, for example, can find optimal or near optimal pathways to food through stigmergy. Arabshahi et al. also inferred that swarm-like agents such as ants optimized the use of local information, thereby eliminating the need for long distance exchanges, and created scalable networks that were generally fault-tolerant.

Swarm-based agents have been applied to the problem of traffic prioritization in multi-path environments. White et al. (2002) developed architectures of multi-agent swarms that improved overall convergence of the address tables present in wireline network routing algorithms. This technique operates most efficiently when wireline connections are persistent. These operations are not supported in MANETs.

Yang, Zincir-Heywood, Heywood, and Srinivas (2002) developed an ant-based routing algorithm for wireline LANs using FANTs and BANTs that were subjected to constant reinforcement learning. Results of the experiment showed that the algorithm developed paths autonomously. Interestingly, heavy traffic loads created better dynamic reinforcement than lighter traffic loads (Yang et al.).

Extrapolation of rules of behavior demonstrated by biologically-based systems of agents such as swarms can resolve key problems in MANETs (Arabshahi et al., 2001; Wedde et al., 2005). These problems include route discovery, route optimization, and route repair. As a consequence, the author examined capabilities of mobile software agent technology as one potential method of managing operations and enhancing performance in multi-protocol ad hoc networking environments.

Following the biologically-based systems example, Wedde et al. (2005) introduced the BeeAdHoc mobile ad hoc routing protocol. The BeeAdHoc protocol uses two types of software agents called scouts and foragers. Through simulations, Wedde et al. showed that BeeAdHoc consumes less energy, a critical factor in MANETs, than previous mobile ad hoc routing protocols such as DSR, AODV and DSDV.

Network simulation

Researchers must test capabilities of TCP/IP and other routing protocols under a variety of conditions to determine their effectiveness and robustness in ad hoc wireless environments (Breslau et al., 2000). The cost of equipment for constructing test environments and the relative inflexibility of building test beds or laboratories to test protocols make protocol testing difficult (White, Lepreau, & Guruprasad, 2003). A multi-protocol network simulator such as *ns-2* enables efficient experimentation and validation of large-scale interaction studies in a controlled environment, along with streamlined comparison of results across the research community (Breslau et al.).

Scherpe and Wolf (2002) described their experience with a real-time network delay and loss simulator (RDLS) and suggested several enhancements for improving their existing model. According to Scherpe and Wolf, the aspects that should be taken into consideration in developing a simulation scenario included delays, losses, fulfillment, network load, mobility routing decisions, resources, and radio quality. The simulator developed by Scherpe and Wolf was proprietary and not publicly available. However, the different network aspects they describe such as delay, losses, network loads, and routing decisions are relevant to this investigation.

Johnson (1999) provided an excellent approach for validating wireless and mobile network simulators. Associated with the Carnegie Mellon University (CMU) Computer

Science Department that developed the mobile networking architectures (Monarch) ad hoc extensions for use with *ns-2*, Johnson was instrumental in developing the dynamic source routing (DSR) protocol. A novel on-demand ad hoc routing protocol, the DSR protocol does not require extensive maintenance of route information by intermediate nodes. In addition, DSR functions well as node mobility increases (Draves et al., 2004). Consequently, DSR was one of the protocols examined in this research. Moreover, extensive prior published research on the performance of DSR, DSDV, TORA, AODV, and ZRP protocols provided quantitative data that were used to check the baseline results of this research (Draves et al.).

Tian et al. (2002) also recommend implementation of a graph-based mobility model operating on *ns-2* platforms. Tian et al. documented the capabilities of DSDV, DSR, and AODV routing protocols; enhanced the *ns-2* environment with the CMU Monarch extensions; and provided tabulated results including average end-to-end delay, routing packet overhead, and packet delivery ratios for these protocols. These findings were useful in evaluating the DSR, DSDV, AODV, and ZRP protocol functions in this investigation.

Summary of the Known and Unknown

Ad hoc networks consist of temporary collections of nodes that can be mobile. These nodes route messages in an environment without a typical wireline infrastructure (Abolhasan & Wysocki, 2003). Ad hoc networks operate in the distributed coordination function (DCF) mode as defined by the IEEE 802.11 WLAN standard and its extensions (Acharya, Misra, & Bansal, 2002). DCF is a MAC layer protocol that implements physical and virtual carrier sense mechanisms to reduce the impact of hidden terminal collisions (Rangarajan & Garcia-Luna-Aceves, 2004). Ad hoc wireless networks present several unique and challenging problems such as mobility, node entry and exit, and changes in topography as well as limitations in battery power, computational power, bandwidth, and coverage that must be addressed in enabling seamless MANET operations (Hać, 2003). In contrast to wireline networks nodes that are stationary, ad hoc wireless nodes are mobile. As a consequence, routing protocols must accommodate nodes entering and exiting the network due to changes in the topography. Limitations in battery power reduce operating distance and restrict computation power. Conventional wireline routing protocols such as the routing information protocol (RIP) and the open shortest path first (OSPF) protocol are unable to manage these challenges and function effectively in ad hoc environments (Lee et al., 2002).

Studies by Lee et al. (2000), Mauve et al. (2001), and Williams and Camp (2002) indicate that ad hoc routing protocols such as DSR, DSDV, and TORA are ineffective in supporting operations in multiple demanding wireless environments although these protocols perform well in single situations. Purely on-demand or table-driven protocols such as DSR, AODV, LAR, and ZRP tend to perform best in specific scenarios, but degrade quickly outside of simulated environments (Williams & Camp, 2002). Hybrid protocols such as ZRP perform better in some instances than purely reactive or proactive protocols but feature serious limitations such as high packet overhead at the environmental extremes including situations of high mobility and high node density (Haas et al., 2002).

Software agents are useful in wireline and wireless network environments (Kawaguchi et al., 2000; Sugar & Imre, 2003). The flexibility, autonomy, and self-organization capabilities of these biologically-based or swarm-based software agents contribute to research involving the resolution of complex problems inherent in ad hoc networks (Arabshahi et al., 2001; Günes et al., 2002).

Contributions to the Field of Study

Ad hoc wireless networks are a growing field of interest in the networking domain. Ad hoc networks are temporary dynamic networks that form without a traditional infrastructure. This ability contributes to ad hoc application development in commercial and military sectors. The difficulties of creating and managing ad hoc networks are considerable due to factors that include topology, mobility, interference, and node additions and deletions (Arabshahi et al., 2001). Solutions developed to enable routing services in conventional wireline environments do not work well in ad hoc networks (Günes et al., 2002; Viswanath & Obraczka, 2002). Consequently, new approaches are in development.

Routing protocols are essential in facilitating effective functions in wireless ad hoc networks. To date, dozens of ad hoc routing protocols are available (Papapetrou & Pavlidou, 2003). Researchers such as Viswanath and Obraczka (2002) and Boleng et al. (2002) concluded that single routing protocols are unable to address challenges associated with ad hoc network deployment. In addition, hybrid protocols that combine the best aspects of on-demand and proactive protocols in support of acceptable solutions are also inefficient. As a consequence of the ineffectiveness of single protocol solutions in ad hoc networks, new approaches including the use of multiple protocols are in development.

According to Arabshahi et al. (2001), White et al. (2002), and Barán and Sosa (2001), software agent technology provides a viable option for augmenting protocol use in dynamic environments. According to Denko (2003), renewed interest in applying mobile software agent technology to ad hoc networks is evidenced by the efforts of Marwaha et al. (2002) and Sugar and Imre (2001).

The efforts of research pioneers such as Di Caro and Dorigo (1998) and more recent work by Arabshahi et al. (2001), Günes et al. (2002), Kassabalidis et al. (2002), Jiang and Camp (2002), and White et al. (2002) provide a framework and potential solution for managing a multi-protocol ad hoc network. According to Arabshahi et al. (p. 3), "Swarm-intelligent routing methods will enhance the reliability and timeliness of data transfer within a heterogeneous multi-node wireless communications network." Arabshahi et al. described the merits of using agent technology to address scalability problems and support robustness. Moreover, Arabshahi et al. explored successful applications of agent technology in traditional wireline networks and concluded that the same agent-based techniques used in wireline networks hold promise for resolving problems in ad hoc wireless networks.

Jiang and Camp (2002) used software agents to perform discrete routing tasks such as providing location information updates. White et al. (2002) employed biologically inspired agents to facilitate priority routing in traditional wireline networks. Agents can manage communications on handheld devices as well (Caire, Lhuillier, & Rimassa, 2002).

This researcher investigated the effects of multiple ad hoc routing protocols operating in the same IEEE 802.11b WLAN environment. Earlier studies focused on the performance of single protocols in homogeneous environments (Xu & Gerla, 2002). In the absence of a flexible and efficient protocol that enables operations in MANETs, multiple protocols were deployed and required to co-exist in the same environment. It is important to note that heterogeneous environments are also an emerging area of study (Ge, Lamont, & Villasenor, 2005; Xu & Gerla; Ahmed, Vanitchannant, & Dao, 2002). Heterogeneous environments contain a mix of different type of nodes such as computers, PDAs, and cellular telephones that operate on the same frequencies using different routing protocols.

For this inquiry, capabilities of agents in mediating protocol communications in a multi-protocol ad hoc wireless environment were also examined. Researchers such as Kassabalidis et al. (2001), Chacón et al. (2000), and Jiang and Camp (2002) applied agents to resolve specific flaws in single routing protocols. However, the use of agents to mediate protocols in heterogeneous and multi-protocol ad hoc wireless environments remains an open area of study.

Summary

This review of the literature featured an introduction and examination of advances in the field of ad hoc wireless networking. A historical perspective on WLAN technology as well as current developments was provided. A special emphasis was placed on exploring the IEEE 802.11 WLAN standard and its extensions and the capabilities of the MAC protocol.

Capabilities of ad hoc networks and the routing protocols that enable these networks to function were discussed in detail. Protocol classification methods including proactive, reactive, and hybrid, and single-hop and multi-hop were described. The features and functions of representative protocols including DSDV, WRP, OLSR, HSR, ZHLS, HARP, DSR, and AODV were reviewed. These protocols were also classified in reactive, proactive, and hybrid categories. Specifically, the reactive or on-demand protocols included DSR, AODV, TORA, and CHAMP. Proactive or table-driven protocols included DSDV, WRP, GSR, OLSR, and FSR. Characteristics of hybrid protocols such as HARP, ZRP, GEDIR, MFR, and DIR with both reactive and proactive characteristics were examined as well. Hierarchical protocols including HSR, WHIRL, CEDAR, LANMAR, and ZHLS were also described.

A discussion on the performance of routing protocols was included. Studies by Tseng et al. (2002) and Williams and Camp (2002) on the performance of different classification groups and specific routing protocols were examined. It is significant to note that Boleng et al. (2002), Kargl et al. (2002), and Viswanath and Obraczka (2002) concluded that an ad hoc routing protocol was incapable of performing effectively in diverse ad hoc environments. The author examined a series of heterogeneous environments consisting of varying populations of nodes operating with DSR or AODV protocols while measuring the effect on packet delivery ratio and latency, measures of network efficiency.

A discussion of agents and agent functions in traditional wireline communications networks was presented. According to Kawaguchi and Inagaki (2000) and Chacón et al. (2000), agents and agent systems are also ideal candidates for resolving problems inherent in wireless ad hoc networks. In particular, the capabilities of biologically-inspired swarm-based agents in resolving specific wireless routing challenges were highlighted.

This chapter also contains a brief discussion of network simulation. According to Lin, Noubir, and Rajaraman (2004), network simulation is an acceptable method for developing and testing network-related software such as routing protocols that are the subject of this research. Network simulators eliminate the cost and complexity of developing large-scale test bed environments and allow rapid prototyping of routing protocols (Jardosh, Belding-Royer, Almeroth, & Suri, 2003).

The contribution of this research to the field of multi-protocol ad hoc networks was also noted. The primary goal of this research was to characterize the behavior of a multi-protocol wireless ad hoc environment. Facilitating communications among multi-protocol nodes using mobile software agent technology was secondary to the primary goal of this research. In this investigation, mobile software agent technology was recognized as a potential mediator for protocol translation in multi-protocol ad hoc networks. Researchers such as Kawaguchi et al. (2000), Sugar and Imre (2003), and Denko (2003) utilized software agents to address only specific aspects of single routing protocols. This author examined the applicability of mobile software agent technology in multi-protocol environments and is based in part on the swarm intelligence work of Arabshahi et al. (2001) and Kawaguchi et al. (2001). In addition, this author expanded findings identified in the pioneering work of Abolhasan et al. (2004), Bhargava et al. (2004), and Cordeiro et al. (2004) in the field of heterogeneous wireless ad hoc networks.

This approach represented a significant change in direction of the current research in the field. Generally, researchers focused their efforts on developing hybrid protocols that accommodate a wide variety of conditions while sacrificing network operating efficiency (Haas & Perlman, 2001; Marwaha et al., 2002; Williams & Camp, 2002). Single protocols such as DSR, DSDV, TORA, and AODV that operate effectively in a narrow range of specific environments become inefficient in extreme environments (Williams & Camp). Environmental factors such as the number of hops and broken links, the size of the network in terms of the number of nodes, and the mobility of nodes continue to provide significant challenges for single routing protocol solutions.

Numerous wireless devices such as notebook computers and PDAs are available that supports the IEEE 802.11b protocol. IEEE 802.11b compliant devices feature capabilities that are useful in diverse situations including heterogeneous environments where different device types operate in conjunction with multiple ad hoc protocols (Xu & Gerla, 2002; Abolhasan et al., 2004). While current research typically focuses on ad hoc routing protocols in homogeneous environments, investigations on the use of ad hoc multi-function routing protocols and mobile software agents in heterogeneous environments are gaining in popularity (Draves et al., 2004; Huang, Lee, & Tseng, 2004; Xu & Gerla).

Chapter 3

Methodology

Approach

The initial phase in this research involved conducting an extensive literature search to ensure that the proposed research contributed to the body of knowledge and advancement of the practice in the field. The methodology required the simulation of an ad hoc network of nodes operating two different ad hoc routing protocols. The *ns-2* simulator enabled simulations for this investigation. The author determined that if the *ns-2* simulator proved to be inadequate, the global mobile simulator (GloMoSim) would be used. While not as widely used in the development community as the *ns-2* simulator, the GloMoSim simulator was developed specifically for evaluating wireless environments and for evaluating single protocol environments, thereby making post-simulation processing of simulation results necessary.

Next, a pair of existing ad hoc routing protocols, specifically DSR and AODV, was selected based on availability within the simulation environment and the large number of studies focusing on the current capabilities and function of these protocols in the literature (Hu, Das, & Pucha, 2002; Roy & Garcia-Luna-Aceves, 2002; Rangarajan & Garcia-Luna-Aceves, 2004). According to Garcia-Luna-Aceves, Mosko, and Perkins (2003), AODV, DSR, and OLSR are representative of the state-of-the-art in ad hoc routing protocols. Importantly, AODV and DSR protocols were also implemented in the current release of *ns-2*. These factors contributed to the use of these protocols for this research, thereby eliminating the need to program other protocols. As a consequence of using protocols built into *ns-2,* potential programming errors were reduced and repeatability and reliability were improved.

A series of simulation experiments were designed and conducted to measure the change in the dependent variables specifically packet delivery ratio also know as throughput and end-to-end delay or latency while manipulating the three independent variables. Independent variables or the number of subject nodes consisted of the introduced nodes and the number of target nodes that formed the base or starting environment. The velocity of the nodes was the third independent variable. Constants were the size of the topography, packet size, transmission radius, radio propagation method, and experimental runtime (Camp, Boleng, Williams, Wilcox, & Navidi, 2002).

The experimental topography consisted of a flat bounded area of 1000 feet by 500 feet. A rectangular area was selected to provide greater interaction opportunities between nodes (Camp et al., 2002). The topography size was similar to that used by Roy and Garcia-Luna-Aceves (2002) and Lee, Han, and Shin (2002) who evaluated similar numbers of nodes. Bounding referred to the inability of nodes within the area to move outside the area. In the bounded environment used in this study, nodes were reflected or bounced off the perimeter. Transmission distance was set to 250 feet, which was the default distance in *ns-2* and the GloMoSim simulators (Jardosh et al., 2003).

Widely employed in simulating movement in *MANETS*, the random waypoint mobility model is also used in this investigation (Hu et al., 2002; Tian et al., 2002;

Viswanath & Obraczka, 2002; Williams & Camp, 2002). According to Yoon, Liu, and Noble (2003), the random waypoint model is the de facto standard in mobile computing research. In terms of operations, the random waypoint model randomly places nodes within an assigned space (Jardosh et al., 2003; Marina & Das, 2001). Each node is assigned a point to move toward that is defined by direction, distance, and velocity. The nodes then move toward their assigned points at the assigned velocities. When a node arrives at the point, it waits for a defined delay period that may be set to zero to simulate continuous movement (Marina & Das). After the delay, the node is assigned a new direction, distance, and velocity and proceeds to the new position. This process continues until the experiment time measured in seconds expires.

The random waypoint model is not without its detractors. According to Yoon et al. (2003), the random waypoint model failed to provide a steady state by inadvertently decreasing velocity of nodes over time. Navidi and Camp (2004) implemented an auxiliary program specific to the *ns-2* simulator to provide a steady state distribution thus eliminating the concerns expressed by Yoon at al. Moreover, Lin, Noubir, and Rajaraman (2004) analyzed the steady state problem and described a framework indicating how a steady state can be achieved to provide accurate results. The auxiliary program developed by Navidi and Camp was used in creating the movement scripts used in this research. Additionally, Bai, Sadagopan, and Helmy (2003) indicated that the random waypoint model did not capture the effects of barriers or obstacles and temporal and spatial dependencies on node movements, whereas temporal dependencies reflect the changes in the network connectivity that varies as a function of time, spatial dependencies exist as relationships between the position of nodes and objects that restrict radio propagation. Barriers and obstacles within the movement area were not utilized in these experiments.

The following experimental parameters were utilized. The transmission radius was set at a constant 250 feet. This distance is commonly used in testing ad hoc protocols (Abolhasan, Lipman, & Chicharo, 2004; Cordeiro et al., 2004; Zhang & Jacob, 2003). The duration of each experiment was set to 200 seconds. Simulation times reported in other research varied from as little as 120 seconds to 9000 seconds (Lo, Liu, & Chen, 2004; Zhang & Jacob).

A run time of 200 seconds was selected for this experimental series to generate sufficient data for throughput analysis without generating excessive data file size (Abolhasan & Wysocki, 2003; Sheu & Chen, 2002). The radio propagation model was two-ray ground reflective commonly used in ad hoc simulations (Shen & Jaikaeo, 2003; Valera et al., 2003). The packet payload size was 512 bytes. This payload size was also used by ad hoc protocol researchers including Hu and Johnson (2000), Kong and Hong (2003), and Rangarajan and Garcia-Luna-Aceves (2004). The number of sending nodes was set to half the total of nodes.

Researchers developing and testing ad hoc routing protocols use a broad range of simulation sets to evaluate their work (Abolhasan et al., 2004; Hu et al., 2002; Marina & Das, 2001; Roy & Garcia-Luna-Aceves, 2002; Williams & Camp, 2002). In each case, the numbers of simulation runs were selected to provide an adequate number of data points and thereby reveal meaningful results. After evaluating the experimental parameters established by investigators such as Abolhasan and Wysocki (2003), Boukerche (2004), Camp et al. (2002), and Zhang and Jacob (2003) who performed studies similar to this inquiry, the author selected a series of six simulation sets.

The first experimental series began with an environment of 10 nodes, 5 sending and 5 receiving featuring the DSR ad hoc routing protocol. The velocity was set at 1 foot per second (fps) and the packet delivery ratio or throughput (packets received to packets sent) and latency or end-to-end delay were measured. The first series of simulations established the baseline performance at the 1 fps velocity and 10 node density. This experiment was repeated using the same mobility model but with velocities of 2 fps, 3 fps, and 4 fps to establish additional baselines. The second and third experimental series followed the same pattern, but increased the number of target nodes to 20 and 30 respectively. The fourth, fifth and sixth experimental series used AODV as the target ad hoc routing protocol but otherwise followed the same pattern.

Once the baseline data points were acquired through these series of 24 simulations, the series were re-evaluated introducing the subject nodes. In the case of the first series, 2 nodes (one sending and one receiving) operating the AODV protocol replaced 2 of the 10 nodes operating the DSR protocol. The same mobility and movement model used in the original baseline series was re-used and packet delivery ratio and latency were measured for the duration of the experiment at 200 seconds. Upon completion, the experiment was repeated at the second and third velocities. Two additional AODV nodes were introduced and the series was evaluated again at each of the four velocities.

The process of replacing DSR nodes with AODV nodes was repeated until the total number of subject nodes equaled the number of target nodes, specifically 10 in the case of the first experimental series. The second and third experimental series followed the same pattern. However, the second and third series increased the starting target nodes to 20 and 30 respectively. Likewise, the subject nodes were introduced until they reached 20 and 30 nodes respectively. The fourth, fifth, and sixth experimental series followed the same pattern while reversing the target and subject protocols in order to determine if there was any asymmetry to the measured effects.

Simulation environment

The use of simulators is critical to the development and design of networking protocols (Walsh & Sirer, 2003). As a consequence, the University of Southern California (USC) Information Sciences Institute (ISI) developed a discrete event network simulation tool called *ns* (Cavin, Sasson, & Schiper, 2002). The current *ns-2* version is formally known as *ns-2.28* and includes a number of enhancements. Since DSR, AODV, and DSDV ad hoc routing protocols are embedded in this release, the *ns-2.28* is appropriate for simulating MANET environments (Jardosh et al., 2003). *ns-2* is a relatively mature simulator with roots in the realistic and large (REAL) network simulator of 1989 (Cavin et al.). White et al. (2003) indicate that *ns-2* is the simulator most frequently used by members of the wireless networking development community.

ns-2 is available to the research community as source code. The source code must be compiled on the computer platform it will be used on. The simulator is available to the public without charge, thereby making this simulator especially appealing for independent research work. *ns-2* operations are well documented (*ns-2*, 2005). However, this simulator lacks a user-friendly interface (*ns-2*), thereby creating a steep learning curve for the new user. A significant amount of research time was required for developing the necessary programming skills in the *ns-2* scripting language. Importantly, researchers such as Stoica (2000) make some simulation scripts for their proposed ad hoc protocols available on the Internet. As noted, routing protocols such as DSDV, TORA, AODV, and DSR are built into the *ns-2* program. In addition, an active community of *ns-2* users is available for consultation. Information is freely shared among users although response time and accuracy of answers may be hit and miss. *ns-2* has been in use for several years and archives of problem resolution threads are available via the Internet. Since search functions are poor, finding specific answers to relatively obscure problems was difficult. Generally, common problem solutions were easy to locate.

Other simulation environments were available in addition to *ns-2*. The ad-hoc network simulator (ANSim) was a new entry into the simulation field. A limited Web-based online version of ANSim was available free of charge. Unfortunately, the Web-based online version did not allow sufficient collection of detailed simulation data such as delay and throughput. While ANSim was able to generate *ns-2* scripts, considerable time programming each simulation session was required with very limited ability to save configurations. In addition, the author found the online version of ANSim was relatively slow and unreliable at completing simulations consisting of more than 20 nodes. Lastly, programming heterogeneous populations of ad hoc protocols was unsupported making ANSim unsuitable for this inquiry.

Another new network simulator, Network Emulator (NE) was available at a moderate cost (Liu & Song, 2002). Like ANSim, NE was designed specifically for investigating challenges in ad hoc networks. The author found NE to be unsuitable for this research because NE was written for older versions the Microsoft Windows operating system and had not been updated. Support was available only with the purchased version.

The global mobile simulator (GloMoSim) developed at the University of California at Los Angles (UCLA) was also available (Jardosh et al., 2003). As with *ns-2*, GloMoSim was designed to simulate a single routing protocol environment. Also as with *ns-2*, GloMoSim was supported by an active user community and was featured in a number of ad hoc wireless research papers (Abolhasan, Lipman, & Chicharo, 2004; Abolhasan & Wysocki, 2003; Lundberg, 2004). As a consequence of its support and acceptance by the research community, GloMoSim was selected by the author to serve as a backup simulator if *ns-2* proved unusable for this research.

Initial experiments performed indicated that *ns-2* provided the data required for this inquiry. Although designed for a single protocol environment, *ns-2* scripts were sufficiently flexible to allow programming a heterogeneous environment. Since *ns-2* was widely accepted in the ad hoc wireless protocol development community by researchers such as Viswanath and Obraczka (2004); Hu et al. (2002); Williams and Camp (2002); Zhang and Jacob (2003), and Al-Shurman et al. (2004) the author used this simulator for this investigation.

A network simulator such as *ns-2* has a number of auxiliary programs such as the Network Animator (Nam) that enhance its function (Nam, 2003). These auxiliary programs also assist in analyzing the data produced through simulation runs. Developed at the USC School of Engineering and ISI, the Nam program processes the log files generated by *ns-2* simulations into animations. Both the *ns-2* and Nam programs are under constant development and review by the user community. Modifications are frequent. As a consequence, changes that potentially impacted *ns-2* results were continually monitored by the author.

Research Assumptions

Assumptions were necessary to constrain the scope of the research. The following represents a high level listing of the major assumptions for this investigation.

- The simulated physical communications medium was the 2.4 to 2.483.5 GHz range of radio frequencies as described by the 802.11b standard and implemented by the *ns-2* simulator. Although other 802.11 standards are currently available, the 802.11b standard is the most widely deployed (Hać, 2003).
- Transmission power was fixed with no variable power output.
- A transmission range of 250 feet was assumed. This range is used by researchers that include Boukerche (2004), Hu and Johnson (2000), Ji and Corson (2001), Hu et al. (2002), and Marwaha et al. (2002).
- Power consumption is another active area of research and can significantly influence the formation and effectiveness of *MANETS* (Buttyán & Hubaux, 2003). In order to reduce the number of variables that may skew research data, power constraints were not considered in this inquiry.
- Mobile devices such as laptop computers and PDAs were assumed to be of the same capability regardless of type.
- Computational power was assumed to be sufficient to support all proposed routing protocols.

- Mobile node storage related to routing caches was assumed to be sufficient to accommodate the environments tested.
- Agent size was assumed to have minimal impact on the performance of host nodes. Minimal impact means that computational power and storage capacity in mobile nodes were not taxed (Denko, 2003). Bandwidth between nodes was, however, a major concern because of the scarcity of this resource. Bandwidth was also measured in the simulation environment.
- Security was assumed. Nodes were assumed to be benign. Buttyán and Hubaux (2003) point out that nodes may not be willing to participate in the forwarding of packets. Some nodes may in fact have malicious intent. The author assumed that the nodes belong to a common authority and were equally motivated to provide forwarding services in a secure environment (Buttyán & Hubaux).
- A two-dimensional, flat experimental environment was utilized without barriers. Real-world scenarios contain many objects that can impair message transference and *MANETS* may exist in three dimensions (Lundberg, 2004). Topology can adversely affect the range and effectiveness of transmissions (Yoon et al., 2003). These factors were not considered in this research.

Resources

The primary resources required for this research were the following.

- Computer system. An HP Media Center PC model 854n with an Intel Pentium 4 processor operating at 2.54 GHz with 1 Gigabyte (GB) of memory and the Microsoft Windows XP operating system.

- *cygwin*, a UNIX emulator program. *cygwin* runs on the Microsoft Windows XP operating system, simulates a relatively standard version of UNIX, and can host the *ns-2* network simulation package. The emulator and auxiliary programs such as a C and C+ compilers, ED and EMACS text editors, and graphical user interfaces (GUI) were available for free at the *cygwin* Web site.
- *ns-2*. Network simulation package. As noted, the current package version is 2.28 and supports ad hoc wireless additions. The software and instructions on its installation are available at the *ns-2* Web site. Importantly, a complete set of the required software packages was assembled for use under the *cygwin* UNIX emulator by Nicolas Christin (2005) at the University of California Berkley (UCB). This prepared set of software packages was used to install the *ns-2* simulator in the *cygwin* environment used in this investigation.
- Nam. A simulation results graphing program. Version is 1.9 was available at no cost from the developers (Nam, 2003).
- Tool control language (tcl) and other UNIX scripting languages for creating scripts that are used by the *ns-2* simulator. Tcl was available at no cost from the developers (Tcl Developers Exchange, 2005).
- X-Windows, a UNIX windowing environment required for Nam use. X-Windows was also useful for running multiple programs within the same desktop environment. A user developed version of X-windows is freely available (Xfree86, 2005).
- Other UNIX utilities such as awk, sed, and grep as required.

- C and C+ programming languages for developing protocols for inclusion in the *ns-2* suite of protocols. C and C+ programming compilers are included in the *cygwin* package as part of the standard set of UNIX utilities.

Reliability and Validity

Reliability was established through the use of standard simulation tools and published versions of existing protocols. Lui and Song (2002) indicated that simulation allows repeatable results in a controlled environment. The tests were repeated to ensure that they produced similar results given similar settings to provide a level of internal validation. The nature of mobile environments is subject to variation in direction, velocity and initial position of nodes. This variation is due to the use of random numbers used to create the mobility script files. Consequently, some variation in individual simulation results was expected. Repeating experiments using the same test parameters allowed convergence of the data to a repeatable norm.

The random waypoint mobility model utilized the stationary distribution program created by Navidi and Camp (2004). This program established a steady state distribution, thereby eliminating the need for arbitrary disposal of initial data points (Navidi & Camp).

The utilization of an unmodified *ns-2* simulation environment allows other researchers in the field to duplicate results in a known simulation environment. The *ns-2* simulation environment continues to be extensively tested and developed by an active MANET research community with a vested interest in accuracy.

The installation process for the *ns-2* simulation environment involved an extensive set of 33 suites of validation tests on different routing protocols with different environmental settings. In all there were over 93 individual tests. The output of each of these tests was compared against a reference set of results that were validated and accepted by the user community (*ns-2*, 2005). The aforementioned validation tests were conducted in the test environment created by the author. Test results matched the user community provided validation results. No errors were generated. Therefore, it was reasonable to assume that extensively tested simulation environment scenarios developed in this research can perform in a similar manner in environments that also were subjected to the same extensive validation tests.

The definitive test of repeatability and external validity is to open the developed test scripts to the research community for testing and comments. Therefore the researcher will make the developed scripts and results available to the active *ns-2* community upon publication of the dissertation at the conclusion of this investigation.

Summary

The intent of this investigation was to advance knowledge and practice in the field of wireless ad hoc networking. As a consequence, this research examined the throughput and delay characteristics in heterogeneous wireless ad hoc networks consisting of nodes operating DSR and AODV routing protocols. Heterogeneous ad hoc networks are an emerging area of study (Abolhasan et al., 2004). Implementation problems of wireless ad hoc networks are challenging for many reasons including the lack of a permanent infrastructure, node mobility, and the frequent adding and dropping of nodes (Hu et al., 2002; Lee et al., 2002). Previous efforts to meet these challenges focused on identifying a single ad hoc routing solution (Viswanath & Obraczka, 2002). Ad hoc routing protocols such as DSR, TORA, DSDV, and HSR provide efficient operation in a limited number of environments (Prakash, 2001). Unfortunately, the same routing protocols lose their effectiveness as the environment changes. As a consequence, researchers such as Boleng et al. (2002), Günes et al. (2002), Kargl et al. (2002), and Viswanath and Obraczka concluded that single ad hoc routing protocol solutions cannot support efficient operations in the diverse environments.

This research is based on findings from previous ad hoc protocol investigations including those conducted by Abolhasan et al. (2004), Park and Park (2004), Arabshahi et al. (2001), and Calafate et al. (2003). Heterogeneous ad hoc networks are an emerging area of study. The author contributed to the body of knowledge by quantifying and analyzing the effects of heterogeneous multi-protocol 802.11b-based environments consisting of DSR and AODV ad hoc protocols. Prior research focused primarily on quantifying performance in single ad hoc routing protocol environments (Ge et al., 2005). Performance measures described in this investigation can serve as the foundation for development of flexible ad hoc routing protocols that coexist in multi-protocol environments. In addition, the author examined the use of an agent mediator as one possible solution for enabling efficient and reliable routing services in heterogeneous multi-protocols environments.

The author conducted an extensive literature search to identify appropriate wireless ad hoc routing protocols and mobile software agents to utilize in this effort. Capabilities of wireless ad hoc routing protocols were examined. Two protocols, specifically DSR and AODV were selected for inclusion in the simulation study as a consequence of their ability to perform well in extreme operational environments (Garcia-Luna-Aceves, Mosko, & Perkins, 2003).

Agent technology was investigated to determine the applicability of agents as mediators in multi-protocol environments. An examination of the literature revealed several applications of agent technology in the field of ad hoc routing. Denko (2003) and Dunne (2001) used agents to organize resources in mobile ad hoc networks. Günes et al. (2002) and Marwaha et al. (2002) introduced new routing protocols based on agent technology. Pirzada and McDonald (2004) used agents to establish trust and security in ad hoc networks. Importantly, Choudhury et al. (2004); Migas et al. (2003); Mochocki and Madley (2005), and Wedde et al. (2005) applied agent technology to improving network efficiency. Each of the agent applications added a layer of complexity and overhead to the ad hoc environment. As a consequence, agent technology is typically used in extreme environments unsupported by existing ad hoc routing protocols.

Direct performance comparisons of agent-based ad hoc routing protocols with existing ad hoc routing protocols showed mixed results. For example, Wedde et al. (2005) compared their BeeAdHoc routing protocol to DSR, AODV and DSDV protocols. Their findings indicated significant improvement in delay when compared to DSR, however improvements compared to AODV and DSDV were insignificant. In addition, BeeAdHoc packet throughput percentages showed minor improvement compared to DSR, but were significantly lower than both AODV and DSDV. As velocity of the nodes increased, the packet throughput improvements compared to DSR became insignificant while both AODV and DSDV outperformed BeeAdHoc. Based on this assessment, an agent protocol framework as a potential solution for facilitating enhanced operations in multi-protocol environments as used in this investigation was not recommended.

For this inquiry, the experiments were programmed and simulated using *ns-2*. Simulation results were compiled and compared against the performance of single protocol solutions established through baseline experiments also performed by this researcher.

Chapter 4

Results

Introduction

This chapter describes the outcomes of the experiments performed in this research. The chapter begins with a detailed examination of the software programs and scripts that were used to conduct the experiments. The processes used to create the movement and communications script files are described. In addition, the specific parameters that were used by the *ns-2* simulation program are presented.

The data gathered from the 10, 20, and 30 node series of experiments are presented graphically. The process used to evaluate the data is described. The analysis of the data is discussed and the results are evaluated. Findings from the outcomes are discussed followed by the chapter summary.

Data analysis

The experimental sequences began with the creation of a number of necessary programs and script files. At the outset, a series of movement script files were developed using the program mobile-ss.exe. Mobile-ss.exe supports creation of movement script files also known as movement scenarios that begin in a steady state (Navidi, Bauer, & Camp, 2003). Yoon et al. (2003) found the random waypoint model failed to establish and maintain continuous steady state movement leading to unreliable results. Movement scripts created with the mobile-ss program begin in a steady state and maintain consistent average velocities throughout the simulation (Navidi & Camp, 2004).

The movement script files specified the number of nodes, the configuration and size of the area or topography for node movement, the duration of the simulation, the velocity and variance of the node movement, the pause time, and the pause variance. A total of 12 movement files that were created used the process outlined in Figure 1. Prior to execution, the required parameters for mobile-ss and the output file were specified on the command line using the format: mobile-ss.exe <number of nodes> <max-x> <max-y> <duration> <velocity mean> <velocity delta> <pause time> <pause time delta> <N> >> mov1-10.txt. With the numeric parameters the command line was: mobile-ss.exe 10 1000 500 200 1 0.5 0 N >> mov1-10.txt. This specified 10 nodes, an area of 1000 feet by 500 feet, a duration of 200 seconds, a mean speed of 1 fps, a speed variance of 0.5 fps, a pause time of 0, *ns-2* compatible script output, and the results saved in file mov1-10.txt.

Movement script files were created for each of the four velocities used by each of the three different populations of 10, 20, and 30 nodes. The four velocities varied plus or minus one-half of the median velocity. For example, with a velocity of 4 fps, the velocity was allowed to vary between 2 and 6 fps.

The topography used throughout these experiments consisted of a rectangle measuring 1000 feet by 500 feet that was a flat space devoid of barriers. In addition, the simulation time was specified as 200 seconds with no pause time between movement transitions and a zero pause time variance to maintain relatively constant movement of nodes within the specified area.

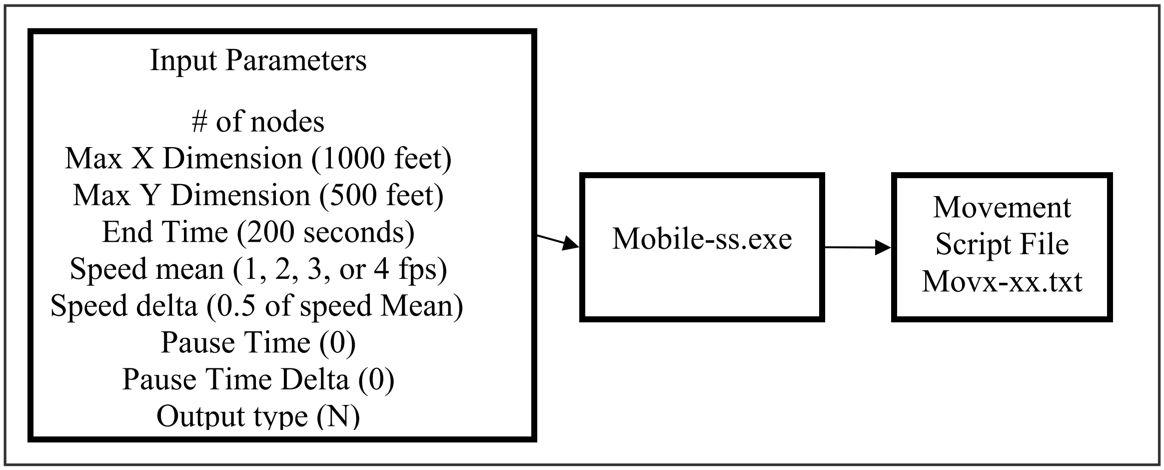

Figure 1. Movement script file creation process

Each movement script file was utilized multiple times in specific sequences of experiments. For example, the movement file mov1-10.txt was established the baseline results of both the AODV and DSR series for 10 nodes with a velocity of 1 fps and a variance of 0.5 fps. In addition, the same file was used in the creation of the series of mixed 10 node AODV and DSR simulations. This process led to an additional four data runs in the AODV and four in the DSR. Printouts of the movement files mov1-10.txt, mov1-20.txt, and mov1-30.txt are presented in Appendices B, C, and D respectively.

Subsequently, communication pattern script files were created using cbrgen.tcl, a program that was included in the *ns-2* suite of auxiliary files. The communications script files created by cbrgen.tcl defined the communications characteristics used between nodes in the simulation environment. In this case, a Constant Bit Rate (CBR) was used with a packet size of 512 bytes and a transmission frequency or rate of four times per second. A seed value of one was used in the random number generator for each run. The maximum number of connections was set to one-half the total populations or 5 in the case of the 10 node series, 10 for the 20 node series, and 15 for the 30 node series.

Examples of the communications script files for each of the three different node populations are presented in Appendices E, F, and G. Figure 2 details the process used to create the communications scripts. Cbrgen.tcl accepts command line parameters prior to interpretation by the *ns-2* simulator. The general format used was: *ns-2.exe* cbrgen.tcl [-type type] [-nn number of nodes] [-seed seed] [-mc maximum connections] [-rate rate] >> output.txt. Using the numeric values the command line would be: *ns-2.exe* cbrgen.tcl –type cbr –nn 10 –seed 1 –mc 5 –rate 4 >> cbr10-1-5-4.txt. This command line specifies 10 nodes, a random number seed of 1, a maximum of 5 connections, a rate of 4 packets per second, and the output file of cbr10-1-5-4.txt.

Once the required input parameters were provided, the cbrgen.tcl program was interpreted by the *ns-2.exe* program. The results were redirected into a text file. The resultant text file contained the communications patterns that were subsequently used by the control scripts. Communications pattern scripts specified the individual nodes that transmitted data packets to specific receiving nodes. In addition, the timing of these transmissions was specified. For example, a communications script indicated that 23.005693 seconds into the simulation node 4 would transmit a 512 byte packet to node 7. The results of this transmission were recorded in the trace and nam output files.

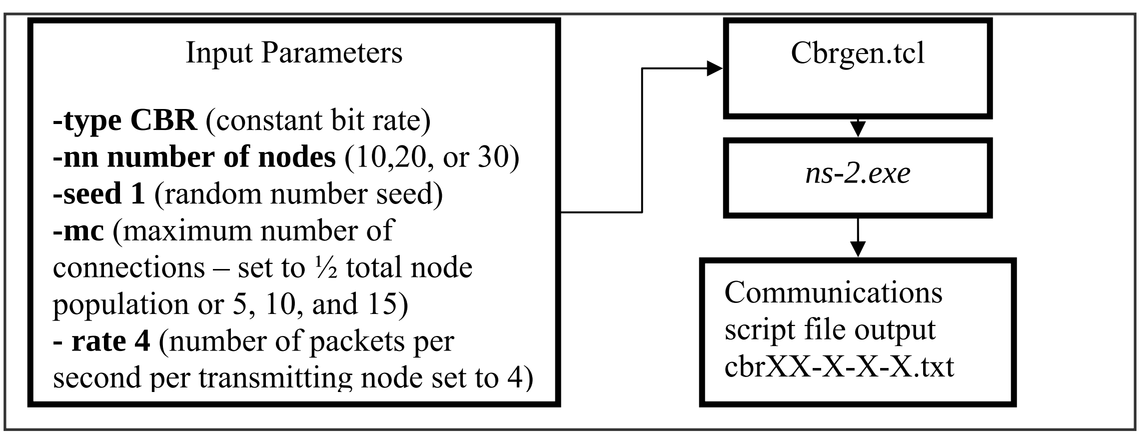

Figure 2. Process for creating communications scripts using cbrgen.tcl

The third step in the process was the development of the *ns-2* control scripts. Control scripts were written by the researcher in tcl and constituted the instruction set that was input into the *ns-2* simulator for each simulation. The movement and communications pattern script files described above were used by the *ns-2* control scripts and in turn by the simulator to generate data in the form of nam and trace files. *ns-2* control scripts included information on the type of network, the radio propagation model, the medium access control (MAC) model, antenna type, topography, ad hoc routing protocols used, and additional data required by the *ns-2* simulator program. The radio propagation model used in this investigation was the *ns-2* default, two ray ground reflection. This model considers both line of sight and ground reflected radio waves (*ns-2*, 2005). The type of network was set to wireless with a MAC model set to 802.11. In addition, the control programs specified the trace and nam output files. An example of the *ns-2* control scripts used in these experiments is included in Appendix H.

The fourth step involved running the control scripts through the *ns-2* simulator. In each simulation, two output files were created, one used specifically by nam and the second, a trace file used by other programs. Nam output files are used to view the movement and transmission patterns of the simulations. Nam files were not directly used to obtain data relating to delay and throughput for evaluating the hypotheses proposed in this research. However, viewing the movement and communication patterns was helpful to the researcher in evaluating results. Both nam and trace files contain detailed structured information from the simulations such as movement and communications data between nodes. As a consequence, these files can be very large in size making storage and manipulation difficult.

Figure 3 indicates the process flow used in running the simulations. As indicated, movement and communications script files were loaded by the control script. The control script was then interpreted by the *ns-2* simulation program. The results of the simulation were output into two files, specifically, the nam and trace files.

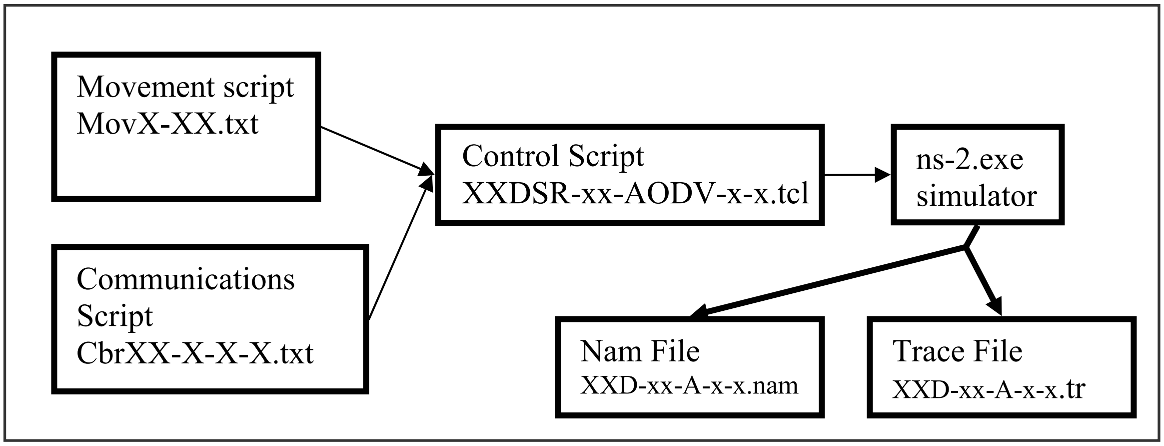

Figure 3. Processing of control scripts

Trace files contained the most detailed information and served as the primary source of data used to obtain results from which average data packet throughput and average end-to-end delay was calculated. Direct evaluation of the data contained within trace files was problematic given the volume of information and the cryptic nature of the entries. Consequently, additional processing of the trace files using auxiliary programs and scripts was necessary to obtain usable information.

One of the programs used to evaluate the trace files was Tracegraph. Tracegraph was developed by Malek (2003) specifically for the analysis of *ns-2* trace files. Tracegraph provided a wide array of information including graphs based on the data present in the trace files. However, the program proved too cumbersome in extracting the specific data required for this research. As a consequence, Tracegraph was used primarily for validating the data derived from programs written by the author.

One of the primary tools developed by the author for processing *ns-2* trace files was A-stat. A-stat was written in awk, a Unix scripting language developed by Aho, Weinberger, and Kernighan (Dougherty & Robbins, 1997). A-stat processed the structured data of the trace files and extracted information such as the number of data packets sent and received by each node, the time sent and the time received, and whether the packet arrived successfully or was dropped in transit. The number of sent and received packets formed the basis for the data packet delivery ratio or throughput, while the send and receive times were used to calculate the end-to-end delay or latency. A copy of the A-stat program is presented in Appendix I. An abbreviated example of the output text file from the A-stat program is shown in Appendix J.

The text output of each run of the A-stat program was imported into Microsoft Excel spreadsheets. The spreadsheets enabled easy manipulation and presentation of the data and featured statistical tools used for data analysis. In addition, Microsoft Excel provided graphing tools for graphically representing the data in the form of figures presented in this report.

10 Node Series Results

A total of 48 *ns-2* traces constituted the data set used to develop average data packet throughput and end-to-end delay in the 10 node series. A baseline was established with four simulation runs in each of the four velocities. Each baseline consisted of two runs for pure AODV environments and two for pure DSR environments consisting of 10 nodes each. After the baselines were established, additional data traces were created substituting the target protocol nodes for the subject protocol nodes while maintaining a total of 10 nodes. For example, the baseline of 10 AODV nodes had 2 ADOV nodes replaced with 2 DSR nodes creating an environment with 8 AODV and 2 DSR nodes. This process was repeated until all of the AODV nodes were replaced with DSR nodes leaving an environment of 10 DSR nodes. The accumulated raw results of this series of simulations are displayed in Appendix K.

As discussed, each data series was run twice with the results averaged to remove any effects that might result from the ordered assignment of specific protocols to individual nodes. The results of these paired runs were averaged to produce the results displayed in Figure 4. The horizontal or x axis indicates the protocol mix with A as the AODV protocol and D as the DSR protocol. The number following the A or D indicates the number of nodes assigned to that protocol.

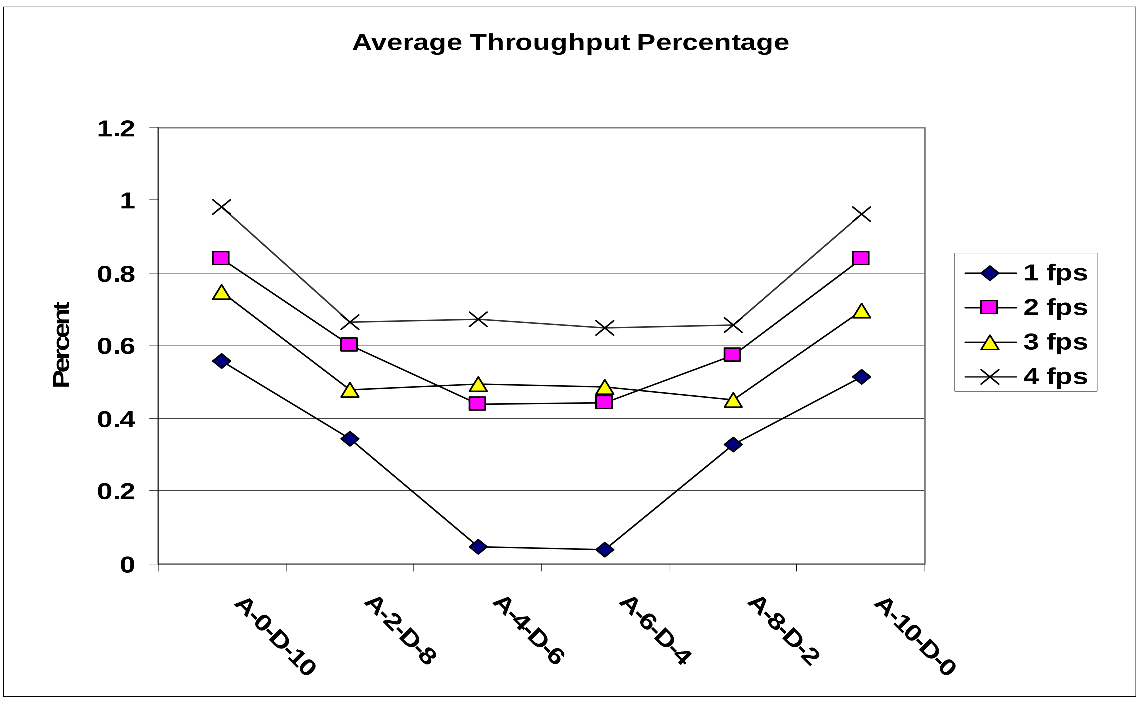

Figure 4. 10 node average data packet throughput percentage

Analysis of the data revealed a general decrease in the average data packet throughput percentage as the mix of subject and target protocols increased. As noted in the Figure 4 graph, the throughput baseline percentage as indicated by the first and last data points of each line decreased at lower velocities and conversely increased at higher velocities. For example, at 4 fps the baseline values represented by the first and last points on the topmost line were nearly 100% indicating that very few data packets were dropped. Conversely, at the 1 fps velocity, the baseline values were close to 50% indicating that almost half of the data packets were dropped in the non-mixed environment. The velocities of 2 fps and 3 fps fell between these extremes.

A statistical analysis of the baseline values for each velocity was performed using a confidence level of 95% or an alpha value of .05. The results are presented in Table M1 in Appendix M. The left legend of Table M1 in Appendix M indicates the velocities in fps. The mean, standard deviation (SD), standard error of the mean (SEM), and the high and low limits were calculated. The SEM was calculated using the standard statistical formula: SEM = $\bar{x} \pm \frac{\sigma}{\sqrt{s}}$, where $\bar{x}$ is the absolute value of the mean, σ is the standard deviation, and s is the sample size. In each case, the sample size was 2. The mean and standard deviation varied with each data set. The SEM was added to the mean to obtain the high limit and subtracted from the mean to obtain the low limit. The high and low values constituted the confidence interval used for evaluating the mixed protocol average packet throughput percentage. Values within the confidence interval were assumed to be a random occurrence or due to chance. Values outside of this range were considered significant from a statistical standpoint given the confidence level of 95% in this instance.

Average values from each of the additional data runs were compared against the high and low values for each of the four velocities. Significance was determined by evaluating each data point gathered from the test series against the high and low limit values or confidence interval using an Excel spreadsheet. The general format of the Excel formula was =OR(data =>High,data<=Low). Where the data fell outside the high and low values, the formula returned TRUE, otherwise the formula returned FALSE. The results were compiled in Table M2 of Appendix M. In each case, the numeric measured results were well below the minimum or low limit values computed at the 95% confidence level.

These data indicated that significant degradation occurred as populations of AODV and DSR nodes became increasingly mixed. As populations approached either all AODV or all DSR populations, the number of dropped data packets decreased, thereby increasing the average data packet throughput percentage. The left column in Table M2 of Appendix M indicates the mix of protocols. For example, A2-D8 indicates a mix of 2 AODV nodes and 8 DSR nodes. The data columns are arranged by velocity. The (T/F) next to each of the numeric data elements indicates the result of the confidence interval evaluation where (T) is true indicating statistical significance and (F) is false indicating a lack of significance or a probable random occurrence. In these results, all measured values were outside the expected limits of random occurrences.

Average latency or end-to-end delay was evaluated through the same process that was used to evaluate throughput. Throughout each data run, data packet send and receive times were accumulated and averaged as the simulation ended. An overview of the data obtained is summarized in Figure 5. The lines represent the four velocities tested. The horizontal or x axis indicates the protocol mix while the vertical or y axis shows the average end-to-end delay or latency in milliseconds. As with the throughput evaluation, a baseline was established using four data points from the data runs consisting of all AODV on the right and all DSR nodes on the left. The average of the AODV and DSR baseline data was represented by the end points of each line.

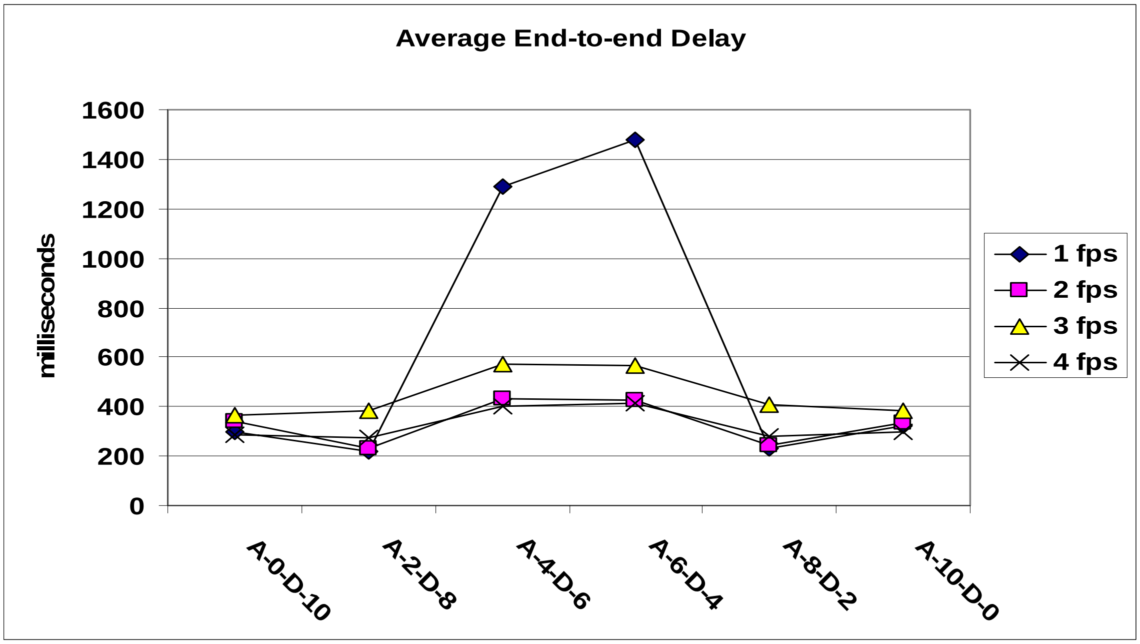

Figure 5. 10 node average end-to-end delay

The baseline averages expressed in milliseconds were used to establish the mean, standard deviation, SEM and high and low limits following the same process as the throughput evaluation. The results are indicated in Table M3 of Appendix M.

The high and low limits or confidence interval were calculated using the same process as the average throughput percentage calculations. These limits were applied through an Excel spreadsheet using the same evaluation formula previously described with the results indicated in Table M4 of Appendix M. The columns are arranged by velocity with rows arranged by the protocol mix. Result of the confidence interval evaluation are indicated with a (T) for true or (F) for false. All numeric values are expressed in milliseconds.

An evaluation of the data showed a significant increase in the average end-to-end delay or latency as the mix of protocols increased. This result was most pronounced at a velocity of 1 fps. Results from simulations at higher velocities indicated a reduced delay effect. The delay remained significant in all cases except the 3 fps two AODV and eight

DSR node results. This anomalous finding resulted from the initial positioning of the two AODV nodes in close proximity at the beginning of the simulation, thereby causing very low delays. When these finding were averaged into the DSR delays, the result was a latency value that fell slightly under the upper limit of the confidence interval.

The overall results of the data analysis of the 10 node series of simulations were strongly supportive of the research hypothesis proposed by this the author. Evaluation of the simulation results using the 95% confidence level showed that there were significant differences in average throughput percentages and latency or end-to-end delay as the mix of protocols increased within this simulation environment.

20 Node Series Results

The second stage in this research involved execution of the series of 20 node simulations. Like the 10 node series, baseline values were established for each of the four tested velocities first using environments with 20 AODV nodes and no DSR nodes and then another using 20 DSR nodes with no AODV nodes.

Programming bias resulted from the sequential assignment of specific protocols to specific node identifiers (IDs). For example, in the 20 node series with 10 DSR and 10 AODV, the first 10 nodes would be assigned to operate using DSR with the second 10 operated AODV. Node IDs were used by movement and communication script files. Both scripts created randomly distributed patterns which might produce more movement or communication within one group or the other. As a consequence, the data produced would be biased toward one group. Running the entire series twice while swapping the starting positions of DSR and AODV distributed any bias evenly. The results from each simulation using the same mix of protocols were then averaged to eliminate any bias.

Once the baselines were established, the replacement series of simulations were run starting with 2 AODV nodes and 18 DSR nodes at each of the four velocities. Next, 2 DSR nodes were replaced with 2 AODV nodes, thereby maintaining a total of 20 nodes; and these simulations were re-run for each of the four velocities tested. This process continued until all nodes utilizing AODV were replaced with DSR nodes.

As with the 10 node series, the entire sequence was repeated starting with 2 DSR nodes and 18 AODV nodes. In all, 88 simulations were run in the 20 node series. The results obtained from the AODV starting sequence and the DSR starting sequence was averaged. For example, the results of 2 AODV and 18 DSR nodes were averaged with 18 DSR and 2 AODV node results. As discussed, this was necessary to eliminate any bias introduced by the assignment of protocols to specifically ordered nodes.

The Figure 6 graph shows the overall results of the averaged data packet throughput percentage analysis for the series of 20 node simulations. The data packet throughput percentage is the ratio of received data packets and sent data packets. The x axis of the chart indicates the protocol mix with A representing the number of nodes assigned the AODV protocol and D the number of nodes assigned the DSR protocol. The y axis is the averaged data packet throughput ratio as a percentage. The four lines represent the four velocities that were tested in the simulations.

The data represented in Figure 6 indicated a general decrease in the average data throughput percentage as the mix of protocols increased. In addition, there was some decrease in the average data throughput percentage as the velocity increased. Averaging the throughput percentages for 1 fps yielded 0.7964; at 2 fps the result was 0.7453, and 3 fps and 4 fps yielded 0.6597 and 0.7341 respectfully. Observing differences in the throughput percentage provided insight for evaluating the general effect of velocity as opposed to effects resulting from the protocol mix. While a linear relationship between velocity and average throughput was not obvious from an analysis, an observed trend toward reduced average data packet throughput as velocity increased was noted. Some variation in specific data points was expected due to the semi-random nature of the random waypoint model used in these experiments (Yoon, Lui, & Noble, 2003). As a consequence, direct observation of linear relationships was somewhat obscured by variations in position and velocities established within the random waypoint model.

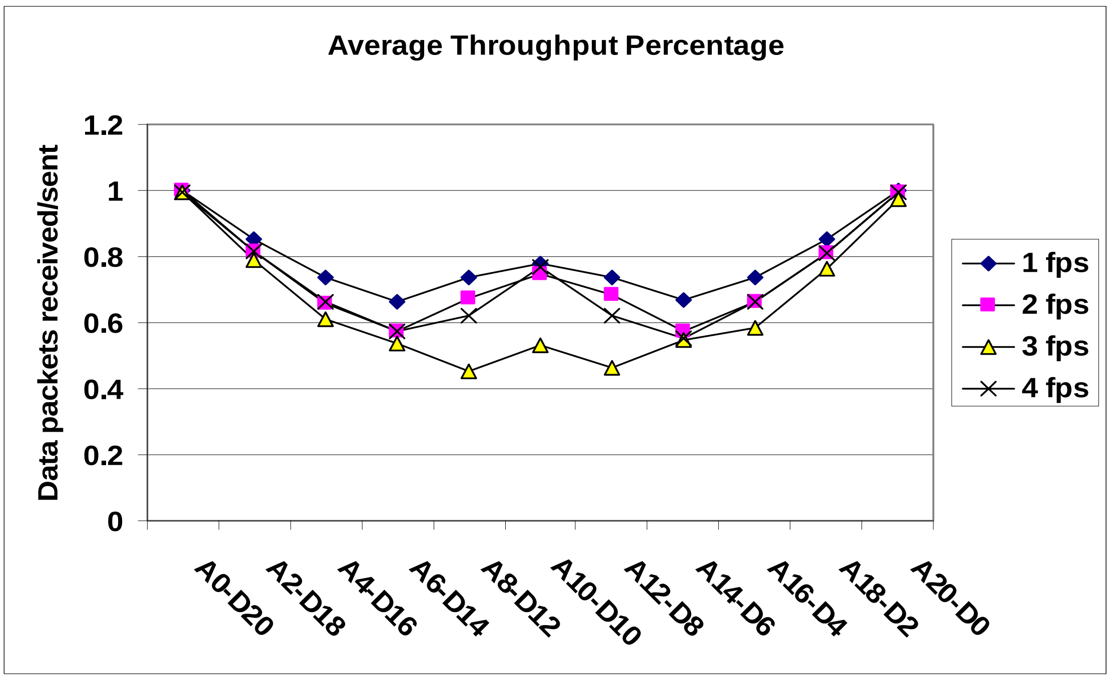

Figure 6. 20 node average data packet throughput percentage

The author applied the same statistical analysis to the average throughput data obtained from the 20 node series of simulations that was used in the 10 node simulations. Table N1 of Appendix N shows the statistical values obtained through an analysis of the baseline simulation results of the 20 node series.

The confidence interval formed from the high and low limit values in Table N1 of Appendix N was applied to the simulation data obtained for each of the mixed 20 node protocol environments. Evaluations were made using the same formula already described using a 95% confidence level. The results of the extended analysis of the significance of the data are tabulated in Table N2 of Appendix N. The confidence interval evaluations for the 20 node simulations showed statistically significant results in all the mixed protocol node simulations as indicated by the (T) or true next to the numerical results. Results supported the hypothesis proposed by this author.

Evaluation of the 20 node average end-to-end delays followed the same process as the throughput evaluation in both the 10 and 20 node series and the end-to-end analysis for the 10 node series. Table N3 of Appendix N reflects the statistical results obtained from evaluating the baseline series for each of the four velocities tested. Each of the four velocities tested are arranged by rows in Table N3 of Appendix N.

The confidence intervals indicated by the high and low limits in Table N4 of Appendix N were evaluated against the average end-to-end delay simulation results for each of the protocol mixes and each velocity in the 20 node series. The results are tabulated in Table N4 of Appendix N. Like the evaluation of the average data packet throughput percentage calculations of the 20 node series, the confidence interval evaluation of the average end-to-end delay revealed statistically significant differences indicated by the (T) next to the numerical results in all mixed node environments.

The mix of protocols is indicated in the far left column of Table N4 of Appendix N. The number of nodes operating AODV is indicated by an A followed by the number of nodes running AODV. Likewise, D represents nodes operating with the DSR protocol followed by the actual number of nodes running DSR. All the values in Table N4 of Appendix N were measured in milliseconds. As in the previous 10 node average end-to-end delay Table, the values presented are the averaged of results obtained from running two series of simulations.

An analysis of data in Table N4 in Appendix N showed significant variation from the baseline data of the pure AODV and DSR environments. However, the relationship between the measured end-to-end delay and the protocol mix was not as clearly indicated as in the results for the 10 node series. Whereas the 10 node results showed clear increases in the end-to-end delay as the protocol mix increased, the 20 node results were both higher and lower than the high and low limits of the confidence level with little apparent relationship to the mix of protocols in the 1 fps results. Figure 7 shows the average end-to-end delay of the four velocities charted against the protocol mix.

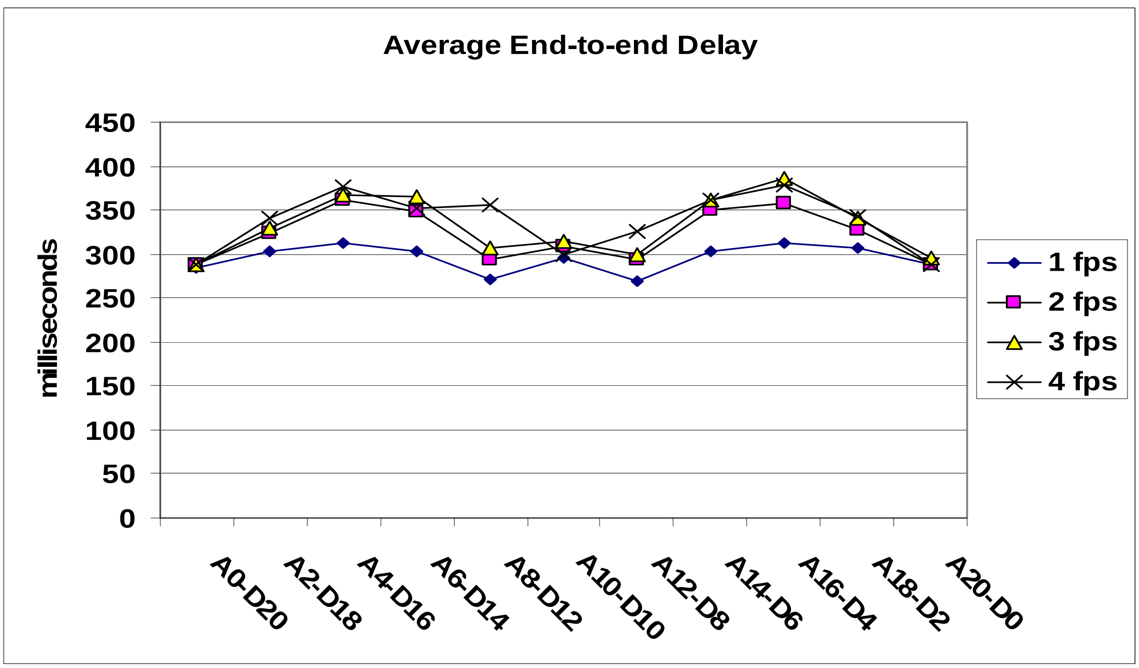

Figure 7. 20 node average end-to-end delay

It is interesting to note that at higher velocities there was a tendency toward higher delays when the protocol mix was dominated by one protocol. The specific dominant protocol seemed to make little difference. As the protocol mix approached an equal mix of AODV and DSR nodes, however, the delay generally declined and became stable. These results appeared somewhat contrary to the 10 node series results. Based on an examination of the graphed results of the 10 node series, the author determined that the 1 fps velocity results indicated high delays as the protocol mix increased. Higher velocities in the 10 node series were less extreme. A comparison of the average baseline means for the 10 and 20 nodes series delays showed a 12% average decline in baseline delay in the 20 node series over the four velocities. Individual difference percentages were 8%, 12%, 22%, and 2% for velocities 1 fps through 4 fps respectively.

30 Node Series Results

The baseline data runs for the 30 node series were established using the same procedure that was used in the 10 and 20 node series. As with the previous series, there were four runs for each of the four velocities tested yielding 16 data points that were used to establish the baselines. These baseline data points also established the confidence interval which was then used to evaluate the subsequent data results. Table O1 of Appendix O details the analysis of the baseline data for average throughput for the 30 node series.

Following the development of the baseline data and the subsequent statistical analysis, the remaining series of simulations were run. In total, there were 14 additional simulations for each of the four velocities forming an additional 56 data sets. As with the previous 10 and 20 node series, these simulations were run twice and the results were averaged together. These results are presented in Table O2 of Appendix O. As with the previous average throughput analysis tables, the (T) next to the numerical value indicates probable statistical significance with (F) indicating a lack of significance. As with the previous evaluation, a confidence level of 95%, or an alpha value of .05 established the confidence interval used in these evaluations. The same formula was used that was employed for the 10 and 20 node series.

The average throughput percentage data is represented graphically in Figure 8. In comparison to the previous graphs, the representation of the 30 node data showed significant flattening in the graphed results. This outcome was a continuation of the trend evident in the differences observed by the author between the 10 node and 20 node throughput series. As noted, there was a tendency toward flattening as the number of nodes and the velocity increased.

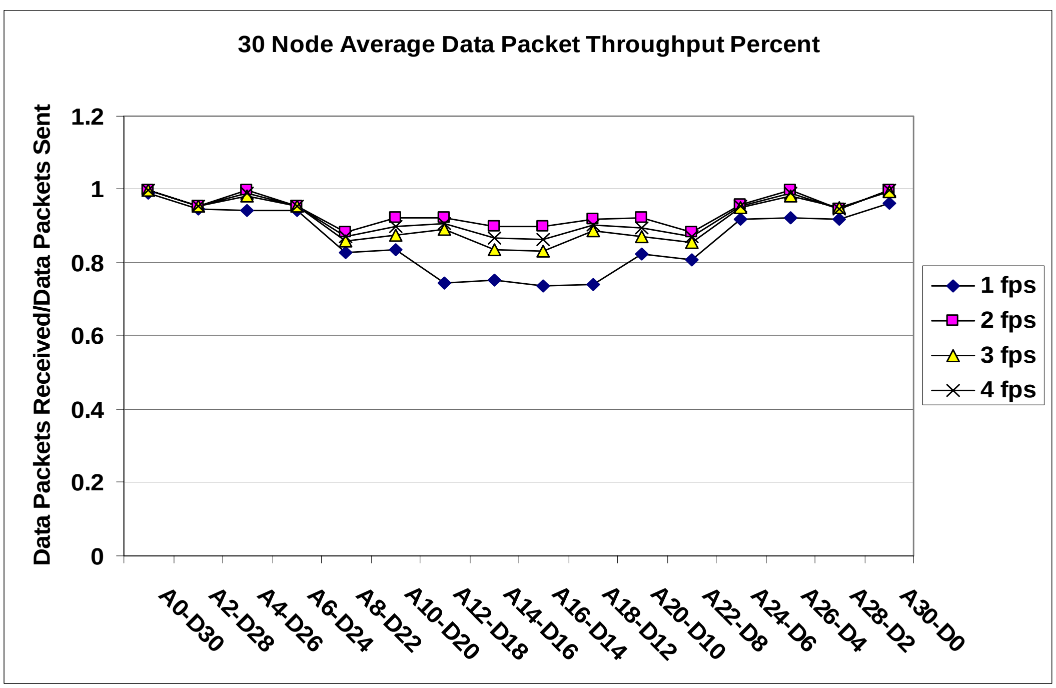

Figure 8. 30 node average data packet throughput percentage

While the results from the data analysis remained consistently significant for velocities of 1, 2, and 3 fps, at 4 fps, two of the 16 data measurements failed to show significance. Both of the readings that failed to show significance occurred in a dominant protocol mix where there were only four nodes of the target protocol. An examination of the graph indicated that the throughput percentage decreased as the mixed protocol environment increased. However, the decline in throughput percentage was less apparent when one protocol dominated the environment.

The statistical analysis of the end-to-end delay or latency of the 30 node series followed the same process that was used in the preceding evaluations. The results of the analysis for the baseline simulation results are shown in Table O3 of Appendix O.

This analysis of the end-to-end delay showed higher average delays for both the 1 fps and the 4 fps baseline series. In addition, the standard deviation was higher, thus resulting in wider confidence intervals. As a consequence, Table O4 of Appendix O indicates that some data points fell within of the 95% confidence interval. In contrast to the 10 and 20 node series, the average delay for the 30 node series showed a considerable reduction in delay times. Whereas a 15% overall decrease in average end-to-end delay was identified from the 10 to the 20 node simulations, a 49% improvement was observed when comparing the 30 and 20 node series. This improvement was the result of the increased density of the nodes. With the greater the node density, a greater probability existed of sending nodes finding a suitable path to the receiving nodes through nodes operating with the same protocol (Lo, Liu, & Chen, 2004). The measured results are presented in Table O4 of Appendix O. All values are indicated in milliseconds.

The far left column in Table O4 of Appendix O showed the mix of protocols with an A indicating AODV and D indicating DSR. A and D are followed by the number of nodes for each protocol. For example, A28-D2 means that there were 28 AODV nodes and 2 DSR nodes in the simulation environment. In each case, 30 nodes were used for this experimental series. The results of the confidence interval evaluations showed the majority of readings remain significant at the 95% confidence level.

Figure 9 demonstrates the relationship between the average end-to-end delay and the protocol mix for each of the four velocities tested in the 30 node series. While there was less correlation of statistically significant delays in relationship to the protocol mix, there was a significant improvement in the overall average delays in comparison to the 10 and 20 node series. The graph also indicated that the delay values were in a narrower range in comparison to the 10 and 20 node series.

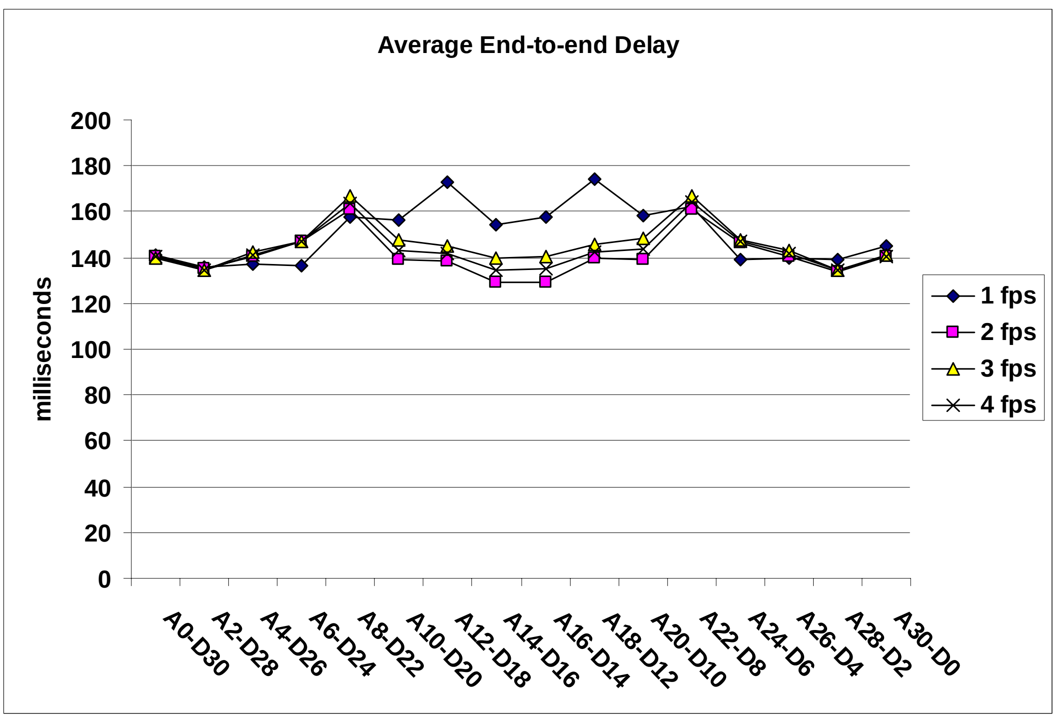

Figure 9. 30 node average end-to-end delay

Findings

The data obtained through the simulations supported the hypothesis proposed by this researcher and allowed rejection of the null hypothesis. The data indicated that there were significant differences in data packet throughput and end-to-end delay or latency averages in mixed AODV and DSR ad hoc routing protocol environments as opposed to environments with only one of these protocols. The differences observed generally increased as the mix of protocols increased and diminished when the majority of nodes operated with either one of the protocols.

Both the average data packet throughput and the average end-to-end delay results were most pronounced when node densities were low and when velocities were low. As node density and velocity increased, average data packet throughput percentages remained consistently significant until the 4 fps 30 node series. At 4 fps, the 30 node series showed two of the 14 data points fell within the 95% confidence interval while the remaining 12 remained significant

A general narrowing of the range of graphed data was observed as node density and velocity increased. This result indicated that the observed differences in average data packet throughput and delay, while remaining significant through the experimental series, diminished in the intensity of the effect. One method of characterizing this effect was by evaluating the percentage change or range of the average data throughput percentage.

The average data packet throughput percentage graphs revealed a relative narrowing of the range of the throughput percentage values as the velocity increased within each of the series of experiments. For example, in the 10 node series, the 1 fps average data packet throughput percentage range varied from a high of 55.8% to a low of

3.3%, thus yielding a range of 55.8%. In contrast, at 4 fps, the 10 node series highest percentage was 98.2% with a low of 64.8%, thus making the range 33.4%. In the 30 node series, the 1 fps high percentage was 98.0% and the low was 86.1%, thereby creating a range of 11.9%. The 30 node 4 fps high percentage was also 98% with a low of 85%, thus yielding a range of 12%. Ranges for the 20 node series fell between the 10 and 30 node figures.

An examination of the average data throughput means further underscores the significance of this result. The 10 node series data packet throughput percentage average of the means was 0.7686 indicating that on average 23.14% of the data packets were dropped in transit. This finding was in stark contrast to the 0.53% and 1.47% dropped packet rate of the 20 and 30 node series respectively. In sparse node environments, many packets were dropped when receiving nodes were out of range of either the sending or relaying nodes (Al-Shurman, Yoo, & Park, 2004). As the data indicated, this problem was diminished as the node density increased.

Lo, Liu, and Chen (2004) indicated that environments with low node densities reduced the probability of connected pairs and thus increased the likelihood of dropped packets. These researchers also indicated that increasing the velocity of a mobile node produced a similar result. As node density increased, the probability of connections increased, thereby effectively reducing the number of dropped packets. Williams and Camp (2002) also noted that node density was a major factor in the performance of ad hoc routing algorithms with poor relative performance at low densities. The observed overall flattening of the data ranges obtained through the 10, 20 and 30 node series simulations in this investigation can be explained by the increasing probability of successful transmissions due to the increased density of nodes.

Latency or end-to-end delay was significantly different through the 10 and 20 node series of simulation. In the 30 node series, however, some instances occurred where the data failed to show significant differences from probable random occurrences. For example, in the 30 node 1 fps and 2 fps series of 28 simulations, 6 results failed to show significant readings. The 3 fps series had only two data points that were not significant. By contrast, in the 4 fps 30 node series, 5 of the 14 data points failed to prove significant at the 95% confidence level.

The author determined that the end-to-end delay improved as node density increased. The 10 node series showed an overall average delay of 328 milliseconds through the four velocities. The 20 node series showed a drop to 288 milliseconds. The 30 node series dropped to only a 141 milliseconds average end-to-end delay through the four velocity series. These results were consistent with the operation of the DSR and AODV protocols. Both DSR and AODV protocols determined the routes to send data packets on demand. Routes were cached once they were determined, thereby reducing the need to continually re-discover the paths (Prakash, 2001). As noted, in environments with dense node populations, the number of possible paths or connections to a destination increased along with the number of nodes that possessed transit information (Al-Shurman, Yoo, & Park, 2004).

The reduction in observed overall end-to-end data packet delay as node density increased was consistent with the results presented in the work of Abolhasan and Wysocki (2003) and Al-Shurman et al. (2004). As with throughput, end-to-end delay was reduced as node density increases. The data showed that the greater the node density in these test environments, the greater the number of possible nearby nodes with paths to a destination. Consequently, the requirement for path discovery was reduced and a higher probability of multiple paths leading to reductions in end-to-end delay was observed.

Throughout this research, significant differences in the measured end-to-end delay were attributed to the mixed protocol environment. As with the throughput analysis, the delay analysis showed a stronger correlation in the node density simulations consisting of 10 and 20 nodes. In the 30 node simulations, several measurements failed to show significance. This outcome was most pronounced at 4 fps where 5 of the 14 data points failed to show significance.

Summary

This chapter began with a detailed examination of the process leading to the execution of the experimental simulations. A discussion of the programs and scripts that were necessary to perform the experiments was presented. The function of each program and script was explained. In addition, an explanation of the programs that were required to analyze the data produced through the simulations was provided. A discussion of the execution of the simulations followed.

The chapter continued with an examination of the results of the 10 node series of simulations. The results of this series of experiments clearly indicated significant differences in both average data packet throughput and end-to-end delay in a mixed protocol environment. Outcomes supported the research hypothesis and allowed the null hypothesis to be rejected. The results also revealed a relationship between the average data packet throughput and the velocity. Specifically, higher velocities improved the baseline throughput percentage and narrowed the range of results.

Next, the results of the 20 node series of simulations were discussed. Like the 10 node series, the average data packet throughput and end-to-end delay results of the 20 node simulations remained significant and were strongly supportive of the research hypothesis allowing rejection of the null hypothesis. The trend toward narrowing of the data range as a result of node density was discussed. It was noted that with the 20 node density in the 1000 foot by 500 foot environment, the number of dropped data packets declined significantly over the 10 node series. In addition, a decline in the end-to-end delay was observed. However, in terms of the average data packet throughput and the average end-to-end delay, the mixed protocol environment data showed statistically significantly differences from the baseline data.

A discussion of the 30 node series of simulations followed. It was noted that there was a continuation of the narrowing of the data range as a result of higher node densities. While the majority of data points remained significantly different through most of the series, there were points that failed to show significance especially at the 4 fps velocity.

Lastly, the findings of the research were discussed in detail. Findings supported the research hypothesis that differences exist in average data packet throughput and end-to-end delay in mixed protocol versus single protocol environments. The data supported rejection of the null hypothesis. A detailed examination of the outcomes revealed direct relationships between the mix of protocols and the differences in average data packet throughput and average end-to-end delay from the baseline values. These differences were in addition to those produced by node density and node velocity.

Differences in the baseline values for each of the series of simulations were examined and found to be the result of the effect of node density and velocity. These results were consistent with previous findings (Abolhasan & Wysocki, 2003; Al-Shurman et al., 2004; Prakash, 2001; Lo et al., 2004). The simulation results were evaluated and shown to have significance in consideration of the effects of node density and velocity. It was noted however that at the highest node density and velocity tested in this research, instances occurred where individual readings did not prove to be significant.

Chapter 5

Conclusions, Implications, Recommendations, and Summary

Conclusions

The data in these simulations supported the conclusion that there were statistically significant differences in the average data packet throughput and end-to-end delay in mixed wireless ad hoc AODV and DSR routing protocol environments as compared to single AODV or DSR routing protocol environments. The data also indicated that the measured delay increased as the protocol mix increased and conversely decreased as the protocol mix decreased. Node density and node velocity were significant factors in average data packet throughput and end-to-end delay as revealed by comparing the baseline values in the simulations. However, the mix of wireless ad hoc routing protocols produced significant results beyond those accounted for by velocity and node density in this series of simulations. The observed variance in throughput and end-to-end delay not attributed to velocity and node density were the result of protocol interaction in the mixed protocol environment.

The data supported rejection of the null hypothesis which stated that there would be no difference in delay and packet throughput between single protocol environments and mixed protocol environments. Moreover, the data also revealed that the effects on throughput and end-to-end delay in AODV and DSR ad hoc routing protocol environments diminished as node density and velocity increased regardless of the protocol mix. In sparse environments, high percentages of packets were dropped primarily because of a lack of transfer or receiving nodes. In these simulations two probable causes were identified. Specifically, no node was within range to receive a packet from the transmitting node or the node within range was operating with a different routing protocol and therefore unable to recognize the packet. In either case, the packet was dropped.

An increase in the velocity of nodes also increased the probability of a transmitting node reaching a suitable receiver even in sparse environments. This outcome was evident in the 10 node baseline results where throughput at 1 fps was nearly 50% while at 4 fps throughput was nearly 100%. By doubling the node density to 20 nodes the positive effect obtained from increasing velocity was obscured as indicated by the baseline throughput values of nearly 100% for pure AODV and DSR protocol environments at all velocities. Similarly, this outcome was observed at 30 node series of simulations where baseline throughput was nearly 100% in all velocities tested. It is likely that these effects were independent of the ad hoc routing protocols implemented; however, proof of the relationship between throughput, velocity, and node density and additional routing protocols requires additional experimentation outside the scope of this research.

Mobile software agents were investigated as a possible method of mediating communications between different ad hoc routing protocols and improving throughput and end-to-end delay in multi-protocol environments. In this inquiry, the agent could act as a third protocol layer taking requests from both protocols in the environment and providing the expected return for the sending protocol as well as forwarding on packets.

In this instance, the agent would not need to self-replicate frequently and transferring new protocols among nodes would not be required. The use of an agent translation layer was expected to increase the control packet overhead (Arabshahi et al., 2001; Wooldridge, 2002).

An examination of the literature indicated that mobile agents are typically used in situations were traditional ad hoc routing protocols fail to perform effectively (Baran & Sosa, 2001; Günes et al., 2002). Agent technology used to improve network efficiency in terms of throughput and delay had limited success (Wedde et al., 2005). In this investigation, the observed opportunity for significant improvements in throughput and delay diminished rapidly as node density and velocity increased. As a consequence, application of agent technology would be effective only in low density and low velocity environments. Based on the limited success producing significant improvements in throughput and delay using agent technology (Wedde et al., 2005), the author determined that an agent technology layer would not increase efficiency in the environments tested in this investigation.

Implications

Results from these experiments indicated that mixed AODV and DSR wireless ad hoc routing protocol environments experienced degradation in data packet throughput and end-to-end delay. Degradation was most pronounced in sparse, low velocity environments consisting of a mix of protocols. As node velocity or node density increased, the negative effects of mixed node protocol operations diminished.

In terms of throughput and delay, observed results from this series of experiments imply that multi-protocol environments consisting of relatively dense populations of nodes can operate effectively as distinct, separate networks without communication between the different protocol populations. Protocol interaction and the hidden terminal problem reduce throughput and increase delay overall, however, these effects were observed to diminish as node density and velocity were increased. In this investigation, high density and high velocity multi-protocol node populations approach the operational effectiveness of single protocol populations.

Results indicate that ad hoc networks in environments such as large conferences or battlefields can operate efficiently with different ad hoc routing protocols provided communication between the different populations is not required. For example, conference participants from France could establish an ad hoc network using the DSR protocol while participants from Germany could use AODV. The efficiency of communication within each group would vary with the individual population density as well as the average velocity of the nodes. Communications between populations of nodes operating DSR and AODV in this example would require the implementation of a bridging technology. Haung et al. (2004) described a two tiered system for bridging mobile and fixed wireless. A similar system could also serve to bridge two mobile populations of differing protocols.

Recommendations

The author examined mixed populations of AODV and DSR ad hoc routing protocols and their effect on throughput and end-to-end delay. Analysis of results indicated that varying the number of nodes and the node velocity while using the random waypoint model did not completely isolate the effect of protocol mix on throughput and end-to-end delay. While the results clearly indicated that the mix of protocols did have an adverse effect on throughput and end-to-end delay, node density, velocity, and node paths also had an effect. Future simulations designed with static environments or highly structured movement patterns could eliminate the indeterminate effect of movement and position produced by the random waypoint movement model.

The author examined the effect of pairing of two of a wide variety of available wireless ad hoc routing protocols. The two protocols utilized in this research, AODV and DSR were both on demand protocols. Additional research pairing different available protocols including combinations of on demand protocols, proactive, and hybrid protocols may result in more efficient multi-protocol environments. Experiments creating more complex environments with more than two wireless ad hoc routing protocols could simulate more realistic real world situations.

Researchers investigating heterogeneous ad hoc networks utilize the power and range of fixed wireless stations to bridge dissimilar networks (Haung et al., 2004). Future development in this area of research could also serve to bridge wireless ad hoc networks operating different routing protocols. Software agents represent another potential technology that could be used to bridge networks operating different protocols. Although agent technology had limited opportunity to improve throughput and delay in this investigation, environments operating different protocols may offer more potential for improvement.

While revealing improvements in end-to-end delay and throughput as node density and velocity increased, this investigation was not designed to fully characterize these effects. Additional research may reveal limits to improvement or degradation in delay and throughput as either node density or velocities reach specific thresholds. Additional research may determine that delay and throughput improvements vary substantially depending on the protocols being tested.

Finally, the author focused on a mixed protocol network using an 802.11b WLAN. Additional experimentation using 802.11a, 802.11g, or the new 802.11n extension may produce different results.

Summary

The IEEE 802.11 standards for WLANs specify two operating modes, infrastructure-based and ad hoc. Infrastructure-based IEEE 802.11 WLANs require a wired component to support discovery, routing, and connectivity management for the wireless nodes. Infrastructure-based WLANs are a common feature in today's connected society. Products supporting the IEEE 802.11a/b/g extensions are readily available and businesses providing wireless connectivity for their customers are commonplace. Wireless ad hoc networks, based on IEEE 802.11 standards are less common. Wireless ad hoc networks, also called mobile ad hoc networks or MANETs are able to form and operate without a wired infrastructure. As a consequence of a lack of wired infrastructure, each device operating in an ad hoc network must have discovery, routing, and connectivity management capabilities. These capabilities are typically the domain of the routing protocol.

Ad hoc routing protocols are an active area of research (Abolhasan et al., 2004; Choudhury et al., 2004; Mochocki & Madey, 2005). The difficulties associated with MANETs such as discovery, connectivity management, power conservation, node adds and drops due to mobility, and security are formidable (Al-Shurman et al., 2004; Lo et al., 2004). As a consequence, many ad hoc routing protocols were introduced in the past decade. Increasingly, researchers such as Viswanath and Obraczka (2002) and Boleng et al. (2002) have determined that single routing protocol solutions able to operate effectively in many different ad hoc environments are unlikely. As a consequence, heterogeneous environments consisting of nodes operating multiple protocols are probable.

Research on heterogeneous networks is an emerging field of study (Bhargava et al., 2004; Cordeiro et al., 2004; Xu & Gerla, 2002). Previous research on ad hoc routing protocol performance focused on single protocol environments (Xu & Gerla, 2002). In this investigation, the author examined the effect multiple ad hoc protocols operating in the same environment had on throughput and end-to-end delay, two factors often used to determine network efficiency (Marina & Das, 2001; Lee et al., 2002; Lin et al., 2004).

The complexity of creating a test environment consisting of 30 or more wireless ad hoc devices was cost prohibitive for this investigation. As a consequence, simulation was selected to develop the data for multi-protocol environments. Simulation is often used by ad hoc routing protocol researchers such as Denko (2003); Hu et al. (2002); Viswanath and Obraczka (2002), and Wedde et al. (2005). Several simulators were examined including *ns-2*, GloMoSim, and ANSim. The *ns-2* simulator was selected based on its broad acceptance by the research community, ability to operate on the computer systems used by the author, and its cost-free availability to the research community. Although *ns-2* was initially designed to simulate homogeneous networks, the author determined that multi-protocol heterogeneous networks could be simulated using *ns-2* scripts.

An extensive literature search examining and documenting the historical foundations and current state of the art in ad hoc networks was presented in chapter 2. WLAN technologies and standards including infrared, narrowband ultra high frequency, IEEE 802.11a/b/g/n, IEEE 802.15, IEEE 802.16, HIPERLAN/1, HIPERLAN/2, and HomeRF were examined in this chapter.

Different approaches used to classify mobile ad hoc routing protocols were investigated as well. The proactive, reactive, hybrid, and agent-based classifications were discussed and representative examples of ad hoc routing protocols from each class were described. Software agents were researched as a possible technology to improve performance in multi-protocol ad hoc networks since these software agents have been applied successfully in wireline networks (Arabshahi et al., 2001; Denko, 2003). As a consequence of their ability to perform complex tasks using simple rules, software agents based on biological systems such as ants and bees were of particular interest in supporting route discovery and maintenance (Baran & Sosa, 2001; Wedde et al., 2005).

The significance of simulation in ad hoc routing protocol research was examined. Researcher such as Abolhasan et al. (2004); Kim et al. (2004); Lo et al. (2004), and White et al. (2003) used simulation to evaluate the performance of new ad hoc routing protocols. In addition, researchers such as Boukerche (2004); Camp et al. (2002), and

Draves et al. (2004) used simulation to compare and contrast performance of existing protocols.

Chapter 3 described the methodology used in this investigation. The author designed experiments to expose potential variations in throughput and delay in multi-protocol environments. Simulation was selected as the means of generating the data required. After an extensive review of the literature, the author used DSR and AODV in this investigation. The selection of DSR and AODV was based on the availability of performance data from prior studies and the inclusion of DSR and AODV in the standard suite of protocols in *ns-2* (Garcia-Luna-Aceves et al., 2003; Wedde et al., 2005).

The author used a flat, featureless topography containing populations of nodes operating the DSR or AODV protocols for experiments in this investigation. Populations consisting of 10, 20, and 30 total nodes were tested. In each total population, different numbers of nodes were assigned to use either the DSR or AODV protocol. For example, in the series of 10 total nodes, 2 nodes were assigned to operate with the DSR protocol while 8 nodes operated with the AODV protocol. The next experiment in the series then used 4 DSR nodes and 6 AODV nodes. The pattern was repeated in this series until all 10 nodes used the AODV protocol. The experiments were repeated reversing the assignment order and the results were averaged.

The random waypoint mobility model was used in this investigation to simulate node movement at velocities of 1 fps, 2 fps, 3 fps, and 4 fps. According to Loon et al. (2003), the random waypoint model is the de facto standard for developing mobility patterns in mobile computing research. Each series of 10, 20, and 30 node experiments were repeated at the four velocities indicated. Finally, research assumptions and the resources required for this investigation were discussed.

Results from this investigation were presented in chapter 4. Details of the processes followed in developing the scripts necessary to run the experiments were described. In addition, the function of each script was indicated. Examples of the scripts and programs developed by the author for this inquiry are featured in Appendices B through I.

The accumulated data on throughput and delay were presented graphically for each of the three populations tested. Results from the 10 node series of experiments showed that mixed populations of nodes operating DSR and AODV protocols had degraded throughput and delay as opposed to either all DSR or all AODV populations. Results from the 20 and 30 node series of experiments also showed significant declines in throughput and delay as a result of the mixed populations.

In addition, the author observed improvements in throughput that were the result of increased node velocity and node density. Specifically, as node velocity increased, throughput and delay improved. This finding was most apparent when observing the baseline values for both throughput and delay in the 10 node series. In the 10 node series, nodes moving a 1 fps dropped almost half of the transmitted packets. At higher velocities, the throughput was dramatically improved.

A statistical analysis of the results was performed. The analysis indicated that mixed protocol environments negatively impacted throughput and delay. As a consequence, the null hypothesis presented in chapter 1 was rejected. The data supported the hypothesis that there would be significant differences in throughput and delay between single protocol environments and multi-protocol environments.

Chapter 5 began with a discussion of the conclusions derived from this series of experiments. It was found that the outcomes from the simulations supported the research hypothesis. The hypothesis stated that there would be significant differences in data packet throughput and end-to-end delay between environments consisting of a single wireless ad hoc routing protocol and environments with two wireless ad hoc routing protocols. Experimental outcomes clearly indicated reductions in data packet throughput and increases in end-to-end delay as a result of the mixed protocol environment.

The significance of node density and velocity on throughput and delay was subsequently examined. Outcomes from this investigation indicated that increases in node density and velocity had a positive impact on the measured end-to-end delay and throughput within the constraints of this set of simulations. However, negative impacts on throughput and delay as a result of protocol interaction in the mixed environment remained significant.

The author then examined implications of the findings. While the data showed significant degradation in throughput and delay as a result of the mixed protocol environment, node density and velocity tended to moderate the effect. At higher levels of node density or at high velocities, both throughput and delay improved significantly. As a consequence of the limited opportunity for improvement to throughput and delay in this investigation, introduction of a mediating mechanism such as agent technology was not required.

Recommendations and suggestions for future experiments completed the chapter. Researchers Lundgren et al. (2006) found that limitations exist in both wireless simulation and testbed environments. Experiment design and monitoring can produce unexpected effects requiring modification of the experimental environment (Lundgren et al.). Findings from this investigation also indicated improvements to the design of the experiments. As a consequence, the author recommended additional experiments pairing more diverse wireless ad hoc routing protocols as an area of future research. In addition, the author recommended modifying experimental parameters such as using a static environment rather than a dynamic random waypoint model and proposed conducting experiments structured with more than two wireless ad hoc routing protocols. The author also recommended conducting additional experimentation using fixed wireless as a mediating mechanism bridging dissimilar mobile networks.

The findings from this investigation add to the current research efforts in wireless ad hoc networks in several ways. First, the author builds upon the work of researchers such as Al-Shurman et al. (2004), Choudhury et al. (2004), and Samar et al. (2004) by providing quantitative data on ad hoc routing protocol performance in a multi-protocol environment. In addition, the author has introduced a scripting technique whereby *ns-2*, originally designed for single protocol simulation, can simulate multi-protocol environments. This provides an additional tool for heterogeneous network researchers using *ns-2* such as Abolhasan et al. (2004), Ge et al. (2005), Huang et al. (2004), and Mochocki and Madley (2005). Finally, a significant finding of this investigation is that multi-protocol environments can function effectively as independent networks given sufficient node density and velocity.

Appendix A

Definitions of Acronyms

Acronym	Definition
3GSM	Third Generation Global System for Mobile Communications
ACM	Association for Computing Machinery
TDM	Time Division Multiplexing
AODV	Ad hoc On-demand Distance Vector
AMRIS	Ad hoc Multicast Protocol Utilizing Increasing Identity Numbers
AMRoute	Ad hoc Multicast Routing Protocol
ANSim	Ad hoc Network Simulator
AP	Access Point
ARA	Ant Colony Based Routing Algorithm
ARPANET	Advanced Research Projects Agency Network
ATM	Asynchronous Transfer Mode
BANT	Backward Ant
BSS	Basic Service Set
CAMP	Core-assisted Mesh Protocol
CCK	Complementary Code Keying
CEDAR	Core Extraction Distributed Ad hoc Routing
CHAMP	Caching and Multi-path Routing Protocol
CMA	Clustering Mobile Agent
CMU	Carnegie Mellon University
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DCF	Distributed Coordination Function
DDR	Distributed Dynamic Routing Algorithm
DIR	Dynamic Incremental Routing
DREAM	Distance Routing Effect Algorithm for Mobility
DSDV	Direct Sequenced Distance Vector
DSR	Dynamic Source Routing
DSSS	Direct Sequenced Spread Spectrum
DTZR	Dynamic Zone Topology Routing Protocol
FANT	Forward Ant
FCC	Federal Communications Commission
FHSS	Frequency Hopping Spread Spectrum
FSR	Fisheye State Routing
GEDIR	Geographic Distance Routing
GFSK	Gaussian Frequency Shift Keying

Acronym	Definition
GHz	Gigahertz
GloMoSim	Global Mobile Simulator
GPRS	Greedy Perimeter Stateless Routing
GPS	Global Positioning System
GSR	Global State Routing
HARP	Hybrid Ad hoc Routing Protocol
HIPERLAN	High Performance Radio Local Area Network
HOLSR	Hierarchical Optimized Link State Routing
HSR	Hierarchical State Routing
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol
IR	Infrared
IrCOMM	Infrared Communications Protocol
IrDA	Infrared Data Association
IrLAP	Infrared Link Access Protocol
IrLMP	Link Management Protocol
ISI	Information Science Institute
ISM	Industrial, Scientific and Medical
IT	Information Technology
LAN	Local Area Network
LANMAR	Landmark Ad hoc Routing
LAR	Location Aided Routing
LS	Location Service
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MANSI	Multicast for Ad hoc Network with Swarm Intelligence
MARP	Multi-agent Routing Protocol
MFR	Most Forward with Fixed Radius
MHz	Megahertz
MPR	Multi-point Relay
NEC	Nippon Electric Corporation
ns-2	Network Simulator version 2
NSU	Nova Southeastern University
ODMRP	On-demand Multicast Routing Protocol
OFDM	Orthogonal Frequency-division Multiplexing
OLSR	Optimized Link State Routing
OSI	Open System Interconnection
PCF	Point Coordination Function
PDA	Personal Digital or Data Assistant
QoS	Quality of Service
RMA	Routing Mobile Agent

Acronym	Definition
RREP	Route Reply
RREQ	Route Request
SD	Standard Deviation
SEM	Standard Error of the Mean
SHARP	Sharp Hybrid Adaptive Routing Protocol
SPF	Straight Packet Forwarding
SS	Spread Spectrum
SWAP	Shared Wireless Application Protocol
Tcl	Tool Control Language
TCP	Transmission Control Protocol
TORA	Temporally Ordered Routing Algorithm
UCB	University of California Berkeley
UHF	Ultra High Frequency
UNII	Unlicensed National Information Infrastructure
USC	University of Southern California
WG	Working Group
WHIRL	Wireless Hierarchical Routing Protocol with Group Mobility
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WRP	Wireless Routing Protocol
ZHLS	Zone-based Hierarchical Link State
ZRP	Zone Routing Protocol

Appendix B

Movement file: mov1-10.txt

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#	Steady-state Random Waypoint Model
#	numNodes  =     10
#	maxX      =   1000.00
#	maxY      =    500.00
#	endTime   =    200.00
#	speedMean  =    1.0000
#	speedDelta =    0.5000
#	pauseMean  =      0.00
#	pauseDelta =      0.00
#	output     =      N
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# output format is NS2
#	Initial positions:
$node_(0) set X_ 377.061589936636
$node_(0) set Y_ 142.950025009604
$node_(0) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(0) setdest 213.580172142750 337.332695181171
1.044557786525"
$node_(1) set X_ 428.222648469468
$node_(1) set Y_ 311.078216077202
$node_(1) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(1) setdest 467.113770296384 339.464847855021
1.239701798020"
$node_(2) set X_ 716.314995811274
$node_(2) set Y_ 26.926685442366
$node_(2) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(2) setdest 751.980183064928 3.176781117533
0.924389606534"
$node_(3) set X_ 550.474972651754
$node_(3) set Y_ 136.376394336778
$node_(3) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(3) setdest 314.189445839352 428.433692282268
1.420711371202"
$node_(4) set X_ 688.377959627763
$node_(4) set Y_ 365.817681435573
$node_(4) set Z_ 0.000000000000
```

```
$ns_ at 0.000000000000 "$node_(4) setdest 961.619300284246 498.728191013787
0.989928157468"
$node_(5) set X_ 840.866588309453
$node_(5) set Y_ 211.150794404886
$node_(5) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(5) setdest 922.182274946097 202.090946818744
0.816870118728"
$node_(6) set X_ 478.463605476334
$node_(6) set Y_ 166.330734047208
$node_(6) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(6) setdest 760.819041058803 239.259250340638
0.762830684322"
$node_(7) set X_ 904.733215141626
$node_(7) set Y_ 461.986875504181
$node_(7) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(7) setdest 505.783993055012 383.626181810920
0.697878558850"
$node_(8) set X_ 136.565695559339
$node_(8) set Y_ 278.707067154826
$node_(8) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(8) setdest 5.835372491663 75.647388154477
0.544364311192"
$node_(9) set X_ 381.566046394972
$node_(9) set Y_ 114.617571092920
$node_(9) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(9) setdest 228.945312662490 84.076566195151
0.774320438426"
#       Movements:
$ns_ at 243.155294820286 "$node_(0) setdest 213.580172142750 337.332695181171
0.000000000000"
$ns_ at 38.839128872981 "$node_(1) setdest 467.113770296384 339.464847855021
0.000000000000"
$ns_ at 38.839128872981 "$node_(1) setdest 385.990210988554 392.668717257990
1.376636833361"
$ns_ at 46.354168933199 "$node_(2) setdest 751.980183064928 3.176781117533
0.000000000000"
$ns_ at 46.354168933199 "$node_(2) setdest 779.594728620534 41.793385307208
1.270500747380"
$ns_ at 83.720741809237 "$node_(2) setdest 779.594728620534 41.793385307208
0.000000000000"
$ns_ at 83.720741809237 "$node_(2) setdest 181.313381614775 273.184748959348
1.046847091777"
$ns_ at 100.161375652251 "$node_(5) setdest 922.182274946097 202.090946818744
0.000000000000"
$ns_ at 100.161375652251 "$node_(5) setdest 758.362679629756 36.567062855031
0.984820822945"
```

$ns_ at 109.310751905800 "$node_(1) setdest 385.990210988554 392.668717257990 0.000000000000"
$ns_ at 109.310751905800 "$node_(1) setdest 411.490526707606 455.340557943723 1.409657455939"
$ns_ at 157.309014924890 "$node_(1) setdest 411.490526707606 455.340557943723 0.000000000000"
$ns_ at 157.309014924890 "$node_(1) setdest 384.659518201211 95.409220128976 0.988870522235"

Appendix C

Movement file: mov1-20.txt

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#	Steady-state Random Waypoint Model
#	numNodes  =    20
#	maxX      =   1000.00
#	maxY      =    500.00
#	endTime   =    200.00
#	speedMean =    1.0000
#	speedDelta =   0.5000
#	pauseMean =      0.00
#	pauseDelta =     0.00
#	output    =      N
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# output format is NS2
#	Initial positions:
$node_(0) set X_ 609.713953566970
$node_(0) set Y_ 94.765015483578
$node_(0) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(0) setdest 32.993479647205 62.798523373342
1.435818447012"
$node_(1) set X_ 503.629374976061
$node_(1) set Y_ 271.798450680344
$node_(1) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(1) setdest 483.239852582682 234.592109096512
1.112178609515"
$node_(2) set X_ 783.783162802574
$node_(2) set Y_ 459.927193749873
$node_(2) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(2) setdest 554.606046320221 467.129013718632
0.515043772021"
$node_(3) set X_ 693.932879391633
$node_(3) set Y_ 187.304402171786
$node_(3) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(3) setdest 894.838210611994 125.351898896206
0.924290645610"
$node_(4) set X_ 154.271257548765
$node_(4) set Y_ 204.187443869336
$node_(4) set Z_ 0.000000000000
```

```
$ns_ at 0.000000000000 "$node_(4) setdest 235.008241718173 310.813499293669
0.684910140937"
$node_(5) set X_ 251.529768161337
$node_(5) set Y_ 168.393214026418
$node_(5) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(5) setdest 978.610490438813 166.718272569924
0.909044020066"
$node_(6) set X_ 294.390269748288
$node_(6) set Y_ 236.963110656178
$node_(6) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(6) setdest 518.918178285900 34.585376519051
0.511929324408"
$node_(7) set X_ 664.020930425334
$node_(7) set Y_ 356.373998317001
$node_(7) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(7) setdest 88.653523516214 430.474404678901
1.289265460848"
$node_(8) set X_ 301.558454509317
$node_(8) set Y_ 243.426792727554
$node_(8) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(8) setdest 845.152732378409 115.187869228044
0.745741574344"
$node_(9) set X_ 386.650526592591
$node_(9) set Y_ 197.037455372650
$node_(9) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(9) setdest 716.033334711582 4.784336083003
1.420554145625"
$node_(10) set X_ 879.053167433224
$node_(10) set Y_ 379.910667594698
$node_(10) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(10) setdest 989.238327829744 411.808145889923
1.038892519299"
$node_(11) set X_ 420.580865470209
$node_(11) set Y_ 463.674678170427
$node_(11) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(11) setdest 276.329505851646 151.493877243015
0.550488799184"
$node_(12) set X_ 648.326841801771
$node_(12) set Y_ 417.177024312360
$node_(12) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(12) setdest 199.247563350595 242.845282304494
1.057275658237"
$node_(13) set X_ 723.287466809567
$node_(13) set Y_ 191.155641126374
$node_(13) set Z_ 0.000000000000
```

```
$ns_ at 0.000000000000 "$node_(13) setdest 506.458271996332 323.496634756912
0.542175067413"
$node_(14) set X_ 177.668913835320
$node_(14) set Y_ 5.950802877886
$node_(14) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(14) setdest 174.171036190433 5.902941341467
1.222986248623"
$node_(15) set X_ 839.610800541788
$node_(15) set Y_ 430.126675054608
$node_(15) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(15) setdest 429.495510845210 299.763786047587
0.835373030110"
$node_(16) set X_ 400.847261465932
$node_(16) set Y_ 162.865929343057
$node_(16) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(16) setdest 959.527865033377 434.229535485725
1.362682834465"
$node_(17) set X_ 792.446071178900
$node_(17) set Y_ 416.557502843634
$node_(17) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(17) setdest 62.854505173329 16.240878038221
0.558880996080"
$node_(18) set X_ 651.592752268853
$node_(18) set Y_ 22.760874376739
$node_(18) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(18) setdest 212.814784242220 269.180033481298
0.733026945497"
$node_(19) set X_ 532.086256715655
$node_(19) set Y_ 234.466501321593
$node_(19) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(19) setdest 992.163338694798 265.256396152664
1.498173360818"
#       Movements:
$ns_ at 402.283252387275 "$node_(0) setdest 32.993479647205 62.798523373342
0.000000000000"
$ns_ at 2.860379729312 "$node_(14) setdest 174.171036190433 5.902941341467
0.000000000000"
$ns_ at 2.860379729312 "$node_(14) setdest 965.334832652162 216.120232230108
0.523848679859"
$ns_ at 38.147587696042 "$node_(1) setdest 483.239852582682 234.592109096512
0.000000000000"
$ns_ at 38.147587696042 "$node_(1) setdest 780.474657090602 401.025453303487
0.992837455353"
$ns_ at 110.414962942456 "$node_(10) setdest 989.238327829744 411.808145889923
0.000000000000"
```

$ns_ at 110.414962942456 "$node_(10) setdest 652.789135767514 385.307159687070 0.534101866667"
$ns_ at 195.272979670240 "$node_(4) setdest 235.008241718173 310.813499293669 0.000000000000"
$ns_ at 195.272979670240 "$node_(4) setdest 559.369591325228 442.483012770528 1.297743815835"

Appendix D

Movement file: mov1-30.txt

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#         Steady-state Random Waypoint Model
#         numNodes   =     30
#         maxX       =   1000.00
#         maxY       =    500.00
#         endTime    =    200.00
#         speedMean  =    1.0000
#         speedDelta =    0.5000
#         pauseMean  =      0.00
#         pauseDelta =      0.00
#         output     =      N
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# output format is NS2
#         Initial positions:
$node_(0) set X_ 613.712027039135
$node_(0) set Y_ 407.781502118027
$node_(0) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(0) setdest 778.178365797819 10.599654405657
0.867165048443"
$node_(1) set X_ 848.222321956188
$node_(1) set Y_ 358.401099268853
$node_(1) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(1) setdest 961.529760137913 355.493049302834
1.058234459467"
$node_(2) set X_ 680.607118547142
$node_(2) set Y_ 169.646111779022
$node_(2) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(2) setdest 740.555793391800 27.772419866068
0.652226246514"
$node_(3) set X_ 373.110890421037
$node_(3) set Y_ 327.706941209160
$node_(3) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(3) setdest 67.360786752478 241.658799928454
0.650866535248"
$node_(4) set X_ 841.815707709850
```

```
$node_(4) set Y_ 413.176368828150
$node_(4) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(4) setdest 858.915223208682 411.970602540286
1.180877481358"
$node_(5) set X_ 217.096397190350
$node_(5) set Y_ 421.859324029156
$node_(5) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(5) setdest 259.328397577316 484.964666881116
1.294193585055"
$node_(6) set X_ 256.424182260918
$node_(6) set Y_ 413.430079555596
$node_(6) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(6) setdest 113.676609058714 426.616670995307
0.801168741819"
$node_(7) set X_ 780.053246273399
$node_(7) set Y_ 362.394090331604
$node_(7) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(7) setdest 352.890131693748 454.746367388752
0.810368453896"
$node_(8) set X_ 476.768083413225
$node_(8) set Y_ 278.397004549892
$node_(8) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(8) setdest 360.469018276999 252.719828045331
1.427930509782"
$node_(9) set X_ 293.709557046073
$node_(9) set Y_ 102.440067302821
$node_(9) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(9) setdest 664.158614195957 374.388675845409
0.858536866585"
$node_(10) set X_ 586.068842944261
$node_(10) set Y_ 246.132104897799
$node_(10) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(10) setdest 687.022038124046 226.469482633504
1.146952281162"
$node_(11) set X_ 78.467013181860
$node_(11) set Y_ 92.662363200449
$node_(11) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(11) setdest 39.278301428667 93.915270918009
0.993153529604"
$node_(12) set X_ 300.856625817404
$node_(12) set Y_ 305.144488019547
$node_(12) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(12) setdest 876.229434682163 403.959264701213
1.005302994653"
$node_(13) set X_ 434.185267501895
$node_(13) set Y_ 19.008160708186
```

```
$node_(13) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(13) setdest 225.755109556837 21.822365243836
0.761356920880"
$node_(14) set X_ 409.204766980916
$node_(14) set Y_ 129.759715831470
$node_(14) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(14) setdest 236.586925683816 67.601430261322
1.301051191699"
$node_(15) set X_ 852.123465215560
$node_(15) set Y_ 388.048226109769
$node_(15) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(15) setdest 606.701148490748 453.372526892169
1.354423522421"
$node_(16) set X_ 218.646035061125
$node_(16) set Y_ 358.179864557683
$node_(16) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(16) setdest 382.919258616361 128.381972959443
1.294380157538"
$node_(17) set X_ 277.434944928625
$node_(17) set Y_ 203.792318526688
$node_(17) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(17) setdest 288.884230558241 187.089915940114
1.041813340580"
$node_(18) set X_ 312.554225908812
$node_(18) set Y_ 370.195332553450
$node_(18) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(18) setdest 971.238921848703 56.077140409535
1.331946957662"
$node_(19) set X_ 492.127314765973
$node_(19) set Y_ 145.622585889827
$node_(19) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(19) setdest 908.521630293001 303.313515290298
0.756391690467"
$node_(20) set X_ 313.934362963425
$node_(20) set Y_ 434.871876422744
$node_(20) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(20) setdest 76.229062432530 454.880705082268
1.377265706732"
$node_(21) set X_ 229.416013865152
$node_(21) set Y_ 304.574308506223
$node_(21) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(21) setdest 122.208689396367 332.885948863293
0.837461672922"
$node_(22) set X_ 839.424728234321
$node_(22) set Y_ 256.693663263651
$node_(22) set Z_ 0.000000000000
```

```
$ns_ at 0.000000000000 "$node_(22) setdest 568.448169421613 485.468710765927
1.428051638795"
$node_(23) set X_ 939.287827806221
$node_(23) set Y_ 233.735920967497
$node_(23) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(23) setdest 986.509317991561 242.531218213277
1.389589496200"
$node_(24) set X_ 689.943106825287
$node_(24) set Y_ 55.013711285647
$node_(24) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(24) setdest 43.066920732645 438.742113271142
1.429589013598"
$node_(25) set X_ 590.255274915271
$node_(25) set Y_ 334.893862143365
$node_(25) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(25) setdest 577.952300001845 355.300805929723
1.113933907117"
$node_(26) set X_ 656.417569998854
$node_(26) set Y_ 119.888292494474
$node_(26) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(26) setdest 279.813360087487 0.206849537886
0.958166218084"
$node_(27) set X_ 328.572574221616
$node_(27) set Y_ 105.758041479865
$node_(27) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(27) setdest 872.432790637218 445.216093885347
0.949508138595"
$node_(28) set X_ 500.559105983323
$node_(28) set Y_ 209.575088050017
$node_(28) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(28) setdest 492.269182806960 70.614737258579
0.978418373489"
$node_(29) set X_ 485.023573289753
$node_(29) set Y_ 150.804713275158
$node_(29) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(29) setdest 99.956948356683 65.496281052705
0.969152319116"


#       Movements:
$ns_ at 495.738067676752 "$node_(0) setdest 778.178365797819 10.599654405657
0.000000000000"
$ns_ at 14.516302619213 "$node_(4) setdest 858.915223208682 411.970602540286
0.000000000000"
$ns_ at 14.516302619213 "$node_(4) setdest 919.863053094532 58.045198003783
1.292372898102"
```

$ns_ at 19.437118373500 "$node_(17) setdest 288.884230558241 187.089915940114 0.000000000000"
$ns_ at 19.437118373500 "$node_(17) setdest 912.991756067142 75.594786124115 0.891694721483"
$ns_ at 21.391477099142 "$node_(25) setdest 577.952300001845 355.300805929723 0.000000000000"
$ns_ at 21.391477099142 "$node_(25) setdest 643.458755055190 313.704017463002 0.920494727055"
$ns_ at 34.566752693662 "$node_(23) setdest 986.509317991561 242.531218213277 0.000000000000"
$ns_ at 34.566752693662 "$node_(23) setdest 171.588039105566 491.383518088322 1.364870510932"
$ns_ at 39.479027070750 "$node_(11) setdest 39.278301428667 93.915270918009 0.000000000000"
$ns_ at 39.479027070750 "$node_(11) setdest 851.983278455205 350.976389297739 1.216181742827"
$ns_ at 58.672085979910 "$node_(5) setdest 259.328397577316 484.964666881116 0.000000000000"
$ns_ at 58.672085979910 "$node_(5) setdest 328.439371813293 186.194702138284 0.813231100940"
$ns_ at 83.407361205134 "$node_(8) setdest 360.469018276999 252.719828045331 0.000000000000"
$ns_ at 83.407361205134 "$node_(8) setdest 762.877216452210 204.424979493220 0.959431933919"
$ns_ at 89.672617398958 "$node_(10) setdest 687.022038124046 226.469482633504 0.000000000000"
$ns_ at 89.672617398958 "$node_(10) setdest 456.280221443754 131.537186508783 0.865286000243"
$ns_ at 105.691369067463 "$node_(25) setdest 643.458755055190 313.704017463002 0.000000000000"
$ns_ at 105.691369067463 "$node_(25) setdest 379.164769490792 420.231991876025 0.876143274073"
$ns_ at 107.107407779596 "$node_(1) setdest 961.529760137913 355.493049302834 0.000000000000"
$ns_ at 107.107407779596 "$node_(1) setdest 991.653556000280 371.650952320383 0.792156525092"
$ns_ at 132.403238904440 "$node_(21) setdest 122.208689396367 332.885948863293 0.000000000000"
$ns_ at 132.403238904440 "$node_(21) setdest 370.709939101110 137.187726626726 1.135112311055"
$ns_ at 141.015339869062 "$node_(14) setdest 236.586925683816 67.601430261322 0.000000000000"
$ns_ at 141.015339869062 "$node_(14) setdest 257.142488964434 487.961781438422 1.042297978672"
$ns_ at 142.277996704476 "$node_(28) setdest 492.269182806960 70.614737258579 0.000000000000"

$ns_ at 142.277996704476 "$node_(28) setdest 677.876341472322 265.456088476561 0.609125766954"
$ns_ at 150.260034810076 "$node_(1) setdest 991.653556000280 371.650952320383 0.000000000000"
$ns_ at 150.260034810076 "$node_(1) setdest 903.301581229689 49.938698555314 0.506315712820"
$ns_ at 173.202551893527 "$node_(20) setdest 76.229062432530 454.880705082268 0.000000000000"
$ns_ at 173.202551893527 "$node_(20) setdest 508.727356562730 325.609420810644 1.166178797216"
$ns_ at 178.932777437676 "$node_(6) setdest 113.676609058714 426.616670995307 0.000000000000"
$ns_ at 178.932777437676 "$node_(6) setdest 914.293215104515 94.561847669334 1.465007578006"
$ns_ at 187.509499694718 "$node_(15) setdest 606.701148490748 453.372526892169 0.000000000000"
$ns_ at 187.509499694718 "$node_(15) setdest 29.587678625056 54.494423584311 0.793457984130"

Appendix E

Communications file: cbr10-1-5-4.txt

```
# nodes: 10, max conn: 5, send rate: 0.25, seed: 1
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.25
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
```

```
$ns_ attach-agent $node_(6) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.25
$cbr_(2) set random_ 1
$cbr_(2) set maxpkts_ 10000
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 146.96568928983328 "$cbr_(2) start"
#
# 6 connecting to 7 at time 55.634230382570173
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.25
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3) start"
#
# 7 connecting to 8 at time 29.546173154165118
#
set udp_(4) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(4)
set null_(4) [new Agent/Null]
$ns_ attach-agent $node_(8) $null_(4)
set cbr_(4) [new Application/Traffic/CBR]
$cbr_(4) set packetSize_ 512
$cbr_(4) set interval_ 0.25
$cbr_(4) set random_ 1
$cbr_(4) set maxpkts_ 10000
$cbr_(4) attach-agent $udp_(4)
$ns_ connect $udp_(4) $null_(4)
$ns_ at 29.546173154165118 "$cbr_(4) start"
#
#Total sources/connections: 4/5
#
```

Appendix F

Communications file: cbr20-1-10-4.txt

```
#
# nodes: 20, max conn: 10, send rate: 0.25, seed: 1
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.25
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(2)
```

```
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.25
$cbr_(2) set random_ 1
$cbr_(2) set maxpkts_ 10000
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 146.96568928983328 "$cbr_(2) start"
#
# 6 connecting to 7 at time 55.634230382570173
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.25
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3) start"
#
# 7 connecting to 8 at time 29.546173154165118
#
set udp_(4) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(4)
set null_(4) [new Agent/Null]
$ns_ attach-agent $node_(8) $null_(4)
set cbr_(4) [new Application/Traffic/CBR]
$cbr_(4) set packetSize_ 512
$cbr_(4) set interval_ 0.25
$cbr_(4) set random_ 1
$cbr_(4) set maxpkts_ 10000
$cbr_(4) attach-agent $udp_(4)
$ns_ connect $udp_(4) $null_(4)
$ns_ at 29.546173154165118 "$cbr_(4) start"
#
# 7 connecting to 9 at time 7.7030203154790309
#
set udp_(5) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(5)
set null_(5) [new Agent/Null]
```

```
$ns_ attach-agent $node_(9) $null_(5)
set cbr_(5) [new Application/Traffic/CBR]
$cbr_(5) set packetSize_ 512
$cbr_(5) set interval_ 0.25
$cbr_(5) set random_ 1
$cbr_(5) set maxpkts_ 10000
$cbr_(5) attach-agent $udp_(5)
$ns_ connect $udp_(5) $null_(5)
$ns_ at 7.7030203154790309 "$cbr_(5) start"
#
# 8 connecting to 9 at time 20.48548468411224
#
set udp_(6) [new Agent/UDP]
$ns_ attach-agent $node_(8) $udp_(6)
set null_(6) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(6)
set cbr_(6) [new Application/Traffic/CBR]
$cbr_(6) set packetSize_ 512
$cbr_(6) set interval_ 0.25
$cbr_(6) set random_ 1
$cbr_(6) set maxpkts_ 10000
$cbr_(6) attach-agent $udp_(6)
$ns_ connect $udp_(6) $null_(6)
$ns_ at 20.48548468411224 "$cbr_(6) start"
#
# 9 connecting to 10 at time 76.258212521792487
#
set udp_(7) [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp_(7)
set null_(7) [new Agent/Null]
$ns_ attach-agent $node_(10) $null_(7)
set cbr_(7) [new Application/Traffic/CBR]
$cbr_(7) set packetSize_ 512
$cbr_(7) set interval_ 0.25
$cbr_(7) set random_ 1
$cbr_(7) set maxpkts_ 10000
$cbr_(7) attach-agent $udp_(7)
$ns_ connect $udp_(7) $null_(7)
$ns_ at 76.258212521792487 "$cbr_(7) start"
#
# 9 connecting to 11 at time 31.464945688594575
#
set udp_(8) [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp_(8)
set null_(8) [new Agent/Null]
$ns_ attach-agent $node_(11) $null_(8)
```

```
set cbr_(8) [new Application/Traffic/CBR]
$cbr_(8) set packetSize_ 512
$cbr_(8) set interval_ 0.25
$cbr_(8) set random_ 1
$cbr_(8) set maxpkts_ 10000
$cbr_(8) attach-agent $udp_(8)
$ns_ connect $udp_(8) $null_(8)
$ns_ at 31.464945688594575 "$cbr_(8) start"
#
# 11 connecting to 12 at time 62.77338456491632
#
set udp_(9) [new Agent/UDP]
$ns_ attach-agent $node_(11) $udp_(9)
set null_(9) [new Agent/Null]
$ns_ attach-agent $node_(12) $null_(9)
set cbr_(9) [new Application/Traffic/CBR]
$cbr_(9) set packetSize_ 512
$cbr_(9) set interval_ 0.25
$cbr_(9) set random_ 1
$cbr_(9) set maxpkts_ 10000
$cbr_(9) attach-agent $udp_(9)
$ns_ connect $udp_(9) $null_(9)
$ns_ at 62.77338456491632 "$cbr_(9) start"
#
#Total sources/connections: 7/10
#
```

Appendix G

Communications file: cbr30-1-15-4.txt

```
#
# nodes: 30, max conn: 15, send rate: 0.25, seed: 1
#
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.25
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
```

```
$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.25
$cbr_(2) set random_ 1
$cbr_(2) set maxpkts_ 10000
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 146.96568928983328 "$cbr_(2) start"
#
# 6 connecting to 7 at time 55.634230382570173
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.25
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3) start"
#
# 7 connecting to 8 at time 29.546173154165118
#
set udp_(4) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(4)
set null_(4) [new Agent/Null]
$ns_ attach-agent $node_(8) $null_(4)
set cbr_(4) [new Application/Traffic/CBR]
$cbr_(4) set packetSize_ 512
$cbr_(4) set interval_ 0.25
$cbr_(4) set random_ 1
$cbr_(4) set maxpkts_ 10000
$cbr_(4) attach-agent $udp_(4)
$ns_ connect $udp_(4) $null_(4)
$ns_ at 29.546173154165118 "$cbr_(4) start"
#
# 7 connecting to 9 at time 7.7030203154790309
#
set udp_(5) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(5)
```

```
set null_(5) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(5)
set cbr_(5) [new Application/Traffic/CBR]
$cbr_(5) set packetSize_ 512
$cbr_(5) set interval_ 0.25
$cbr_(5) set random_ 1
$cbr_(5) set maxpkts_ 10000
$cbr_(5) attach-agent $udp_(5)
$ns_ connect $udp_(5) $null_(5)
$ns_ at 7.7030203154790309 "$cbr_(5) start"
#
# 8 connecting to 9 at time 20.48548468411224
#
set udp_(6) [new Agent/UDP]
$ns_ attach-agent $node_(8) $udp_(6)
set null_(6) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(6)
set cbr_(6) [new Application/Traffic/CBR]
$cbr_(6) set packetSize_ 512
$cbr_(6) set interval_ 0.25
$cbr_(6) set random_ 1
$cbr_(6) set maxpkts_ 10000
$cbr_(6) attach-agent $udp_(6)
$ns_ connect $udp_(6) $null_(6)
$ns_ at 20.48548468411224 "$cbr_(6) start"
#
# 9 connecting to 10 at time 76.258212521792487
#
set udp_(7) [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp_(7)
set null_(7) [new Agent/Null]
$ns_ attach-agent $node_(10) $null_(7)
set cbr_(7) [new Application/Traffic/CBR]
$cbr_(7) set packetSize_ 512
$cbr_(7) set interval_ 0.25
$cbr_(7) set random_ 1
$cbr_(7) set maxpkts_ 10000
$cbr_(7) attach-agent $udp_(7)
$ns_ connect $udp_(7) $null_(7)
$ns_ at 76.258212521792487 "$cbr_(7) start"
#
# 9 connecting to 11 at time 31.464945688594575
#
set udp_(8) [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp_(8)
set null_(8) [new Agent/Null]
```

```
$ns_ attach-agent $node_(11) $null_(8)
set cbr_(8) [new Application/Traffic/CBR]
$cbr_(8) set packetSize_ 512
$cbr_(8) set interval_ 0.25
$cbr_(8) set random_ 1
$cbr_(8) set maxpkts_ 10000
$cbr_(8) attach-agent $udp_(8)
$ns_ connect $udp_(8) $null_(8)
$ns_ at 31.464945688594575 "$cbr_(8) start"
#
# 11 connecting to 12 at time 62.77338456491632
#
set udp_(9) [new Agent/UDP]
$ns_ attach-agent $node_(11) $udp_(9)
set null_(9) [new Agent/Null]
$ns_ attach-agent $node_(12) $null_(9)
set cbr_(9) [new Application/Traffic/CBR]
$cbr_(9) set packetSize_ 512
$cbr_(9) set interval_ 0.25
$cbr_(9) set random_ 1
$cbr_(9) set maxpkts_ 10000
$cbr_(9) attach-agent $udp_(9)
$ns_ connect $udp_(9) $null_(9)
$ns_ at 62.77338456491632 "$cbr_(9) start"
#
# 11 connecting to 13 at time 46.455830739092008
#
set udp_(10) [new Agent/UDP]
$ns_ attach-agent $node_(11) $udp_(10)
set null_(10) [new Agent/Null]
$ns_ attach-agent $node_(13) $null_(10)
set cbr_(10) [new Application/Traffic/CBR]
$cbr_(10) set packetSize_ 512
$cbr_(10) set interval_ 0.25
$cbr_(10) set random_ 1
$cbr_(10) set maxpkts_ 10000
$cbr_(10) attach-agent $udp_(10)
$ns_ connect $udp_(10) $null_(10)
$ns_ at 46.455830739092008 "$cbr_(10) start"
#
# 13 connecting to 14 at time 83.900868549896813
#
set udp_(11) [new Agent/UDP]
$ns_ attach-agent $node_(13) $udp_(11)
set null_(11) [new Agent/Null]
$ns_ attach-agent $node_(14) $null_(11)
```

```
set cbr_(11) [new Application/Traffic/CBR]
$cbr_(11) set packetSize_ 512
$cbr_(11) set interval_ 0.25
$cbr_(11) set random_ 1
$cbr_(11) set maxpkts_ 10000
$cbr_(11) attach-agent $udp_(11)
$ns_ connect $udp_(11) $null_(11)
$ns_ at 83.900868549896813 "$cbr_(11) start"
#
# 14 connecting to 15 at time 155.17211061677529
#
set udp_(12) [new Agent/UDP]
$ns_ attach-agent $node_(14) $udp_(12)
set null_(12) [new Agent/Null]
$ns_ attach-agent $node_(15) $null_(12)
set cbr_(12) [new Application/Traffic/CBR]
$cbr_(12) set packetSize_ 512
$cbr_(12) set interval_ 0.25
$cbr_(12) set random_ 1
$cbr_(12) set maxpkts_ 10000
$cbr_(12) attach-agent $udp_(12)
$ns_ connect $udp_(12) $null_(12)
$ns_ at 155.17211061677529 "$cbr_(12) start"
#
# 15 connecting to 16 at time 39.088702704333095
#
set udp_(13) [new Agent/UDP]
$ns_ attach-agent $node_(15) $udp_(13)
set null_(13) [new Agent/Null]
$ns_ attach-agent $node_(16) $null_(13)
set cbr_(13) [new Application/Traffic/CBR]
$cbr_(13) set packetSize_ 512
$cbr_(13) set interval_ 0.25
$cbr_(13) set random_ 1
$cbr_(13) set maxpkts_ 10000
$cbr_(13) attach-agent $udp_(13)
$ns_ connect $udp_(13) $null_(13)
$ns_ at 39.088702704333095 "$cbr_(13) start"
#
# 15 connecting to 17 at time 43.420613009212822
#
set udp_(14) [new Agent/UDP]
$ns_ attach-agent $node_(15) $udp_(14)
set null_(14) [new Agent/Null]
$ns_ attach-agent $node_(17) $null_(14)
set cbr_(14) [new Application/Traffic/CBR]
```

```
$cbr_(14) set packetSize_ 512
$cbr_(14) set interval_ 0.25
$cbr_(14) set random_ 1
$cbr_(14) set maxpkts_ 10000
$cbr_(14) attach-agent $udp_(14)
$ns_ connect $udp_(14) $null_(14)
$ns_ at 43.420613009212822 "$cbr_(14) start"
#
#Total sources/connections: 10/15
#
```

Appendix H

Control file: AODV-10-1-5-4-0.tcl

```
# 30AODV-10-DSR-20-1.tcl
# Makes use of 'standard' movement and communications
# pattern files (sc) and (cp)
#
======================================================================
=
# Define options
#
======================================================================
=
set val(chan)       Channel/WirelessChannel   ;# Channel model
set val(prop)       Propagation/TwoRayGround          ;# radio propagation model
set val(netif)      Phy/WirelessPhy                ;# physical layer (wireless)
set val(mac)        Mac/802_11                     ;# MAC model is set to 802.11
set val(ifq)      Queue/DropTail/PriQueue      ;# Queue type for DSDV & AODV
# set val(ifq)        CMUPriQueue                  ;# Use this Queue for DSR
set val(ll)         LL
set val(ant)        Antenna/OmniAntenna                   ;# Antenna type to use
set val(x)              1000                           ;# X dimension of the topography
set val(y)              500                            ;# Y dimension of the topography
set val(ifqlen)         10                  ;# max packet in ifq
set val(seed)           1.0
set val(adhocRouting)   AODV                                ;# this is the protocol to use
set val(nn)             30                  ;# how many nodes are simulated
set val(nn1)                10
set val(nn2)                30
set val(cp)          "cbr30-1-15-4.txt"   ;# this is the transmission pattern
set val(sc)          "mov1-30.txt"                 ;# this is the movement pattern file
set val(stop)           200.0               ;# simulation time to run in seconds

#=====================================================================
==
# Main Program
#=====================================================================
==
#
```

```
# Initialize Global Variables
#
# create simulator instance

set ns_         [new Simulator]

# setup topography object

set topo        [new Topography]

# create trace object for ns and nam
# these are the files that record the results

set tracefd     [open 30A-10-D-20-1.tr w]
set namtrace    [open 30A-10-D-20-1.nam w]

# tell the program to trace all the events

$ns_ use-newtrace
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# define topology

$topo load_flatgrid $val(x) $val(y)

#
# Create god
#
set god_ [create-god $val(nn)]

#
# define how node should be created
#
# global node setting

set val(adhocRouting)   AODV
set val(ifq)      Queue/DropTail/PriQueue
$ns_ node-config -adhocRouting $val(adhocRouting) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
```

```
            -channelType $val(chan) \
                  -topoInstance $topo \
                  -agentTrace ON \
                  -wiredRouting OFF \
            -routerTrace ON \
            -macTrace ON
#
#  Create the specified number of nodes [$val(nn)] and "attach" them
#  to the channel.

for {set i 0} {$i < $val(nn1) } {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0          ;# disable random motion
}
#

set val(adhocRouting)   DSR
set val(ifq)      CMUPriQueue
$ns_ node-config -adhocRouting $val(adhocRouting) \
            -llType $val(ll) \
            -macType $val(mac) \
            -ifqType $val(ifq) \
            -ifqLen $val(ifqlen) \
            -antType $val(ant) \
            -propType $val(prop) \
            -phyType $val(netif) \
            -channelType $val(chan) \
                  -topoInstance $topo \
                  -agentTrace ON \
                  -wiredRouting OFF \
            -routerTrace ON \
            -macTrace ON
#
#  Create the specified number of nodes [$val(nn)] and "attach" them
#  to the channel.

for {set i 10} {$i < $val(nn2) } {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0          ;# disable random motion
}

# Define node movement model
# In this case, a movement model is loaded from the file indicated above (cp)
#

puts "Loading connection pattern..."
```

```
source $val(cp)

#
# Define traffic model
# In this case, the traffic model is loaded from the file indicated above (sc)

puts "Loading scenario file..."
source $val(sc)

# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

   # 20 defines the node size in nam, must adjust it according to your scenario
   # The function must be called after mobility model is defined

   $ns_ initial_node_pos $node_($i) 20  ;# value indicates size of circle from 20 to 10
}

#
# Tell nodes when the simulation ends
#

for {set i 0} {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop).0 "$node_($i) reset";
}

# End the simulation

$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"

# store the first four lines of the trace data

puts $tracefd "M 0.0 nn $val(nn) nn1 $val(nn1) nn2 $val(nn2)"
puts $tracefd "M 0.0 x $val(x) y $val(y) rp $val(adhocRouting)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

# Run the simulation

puts "Starting Simulation..."
$ns_ run
```

Appendix I

A-stat.awk

```
# USE:   awk -f allc.awk tracefile_Name
# This is an awk program that summarizes information about Network Simulation
#

BEGIN {
  highest_packet_id = 0;
  highest_flow_id = 0;
  highest_node_id = 0;
  duration_total = 0;
  flow_number = 0;
  dropped = 0;
  pkt = 512;
  simtime = 200;
  node_id = 0;
  max_nodes = 20;
  sendtime = 0;
  sent = 0;
  fwdnode = 0;
  pktcount = 0;
}

####### Packet Delivery Fractions ##########################################
//              { total++ }                            # Total gross packets
/AGT/||/RTR/||/MAC/ || /ARP/     { totalNL++ }         # total packets (Network trace Level)
/AGT/           {agt++}                                # total packets  AGT
/RTR/           {rtr++}                                # total packets  RTR
/MAC/           {mac++}                                # total packets  MAC
/ARP/           {arp++}                                # total packets  ARP
/AGT/ && /-It cbr/ {data_agt++}                        # total data packets
/^s/                {sent++}                           # sent packets total
/^s/ && /AGT/    {agt_sent++}                          # sent packets AGT
/^s/ && /RTR/    {rtr_sent++}                          # sent packets RTR
/^s/ && /MAC/    {mac_sent++}                          # sent packets MAC
/^s/ && /AGT/ && /-It cbr/ {data_agt_sent++}           # sent data packets
/^r/                {rec++}                            # received packets total
/^r/ && /AGT/    {agt_rec++}                           # received packets AGT
/^r/ && /RTR/    {rtr_rec++}                           # received packets RTR
```

```
/^r/ && /MAC/    {mac_rec++}                        # received packets MAC
/^r/ && /AGT/ && /-It cbr/ {data_agt_rec++}         # received data packets
/^d/               {drop++}                         # dropped packets total
/^d/ && /AGT/    {agt_drop++}                       # dropped packets AGT
/^d/ && /RTR/    {rtr_drop++}                       # dropped packets RTR
/^d/ && /MAC/    {mac_drop++}                       # dropped packets MAC
/^d/ && /AGT/ && /-It cbr/ {data_agt_drop++}        # dropped data packets
/^d/ && /IFQ/    {ifq_drop++}          # ROP_IFQ_QFULL i.e no buffer space in IFQ.
/^f/            {forw++}                            # forwarded packets total
/^f/ && /AGT/    {agt_forw++}                       # forwarded packets AGT
/^f/ && /RTR/    {rtr_forw++}                       # forwarded packets RTR
/^f/ && /MAC/    {mac_forw++}                       # forwarded packets MAC
/^f/ && /AGT/ && /-It cbr/ {data_agt_forw++}        # forward data packets

######## Throughput Analysis  ####################################

{  event = $1;
   time = $3;
   node_id = $5;
   packet_size = $37;
   flow_id = $39;
   packet_id = $41;
   flow_t = $45;

   # Determine the highest packet ID
        if ( packet_id > highest_packet_id ) highest_packet_id = packet_id;
   # Determine the highest flow ID
        if ( flow_id > highest_flow_id ) highest_flow_id = flow_id;
   # Determine the highest node ID
        if ( node_id > highest_node_id ) highest_node_id = node_id;

        if ( ($19 == "AGT" ) && (start_time[flow_id] == 0) ){
          start_time[flow_id] = time;   }

# Determine receive times of data packets
   if ( ($1 == "r") && ($19 == "AGT") && ($35 == "cbr")) {
     end_time[flow_id] = time;
     nodert[node_id] = time;
     rnode[node_id]++;
     }

##### Begin individual node analysis ####################################

# Store packets send time of data packets
if (sendTime[packet_id] == 0 && (event == "+" || event == "s") && packet_size >= pkt)
{
```

```
            sendTime[packet_id] = time;
            nodest[node_id] = time;
            }
# Count dropped packets
      if (event == "d" && packet_size >= pkt ) {
            dnode[node_id]++;
            dropped ++;
            }
# Count total sent packets
      if ((event == "s" || event == "+") && $19 == "AGT" && packet_size >= pkt) {
            snode[node_id]++;
            }
      if ((event == "f" ) && ($19 == "AGT" && $35 == "cbr") && packet_size >=
pkt) {
            fnode[node_id]++;
            fwdnode ++;
            }
# Update total received packets' size and store packets arrival time
      if (event == "r" && $19 == "AGT" && packet_size >= pkt) {
      # Rip off the header
            hdr_size = packet_size % pkt;
            packet_size -= hdr_size;
      # Store received packet's size
            recvdSize += packet_size;
      # Store packet's reception time
            recvTime[packet_id] = time;
      # Set individual nodes receive time
            nodert[node_id] = time;
      # Set individual nodes bytes received
            noderecsize[node_id] += packet_size;
          }
# Compute average delay
      delay = avg_delay = recvdNum = 0
      for (i in recvTime) {
            if (sendTime[i] == 0) {
              printf("\nError in delay.awk: receiving a packet that wasn't sent %g\n",i)
              }
            delay += recvTime[i] - sendTime[i];
            recvdNum ++;
        }
      if (recvdNum != 0) {
            avg_delay = delay / recvdNum;
        } else {
            avg_delay = 0;
        }
      for (node_id = 0; node_id < max_nodes; node_id ++){
```

```
        if ((nodert[node_id] - nodest[node_id]) >= 0) {
                node_delay[node_id] = (nodert[node_id] - nodest[node_id])}
        if (noderecsize[node_id] != 0) {
          anode_delay[node_id] = node_delay[node_id]/rnode[node_id]}
        }
}
##### End of individual node analysis ################
END {

#### Print Packet Delivery Fractions ###################################
    cp_sent = rreq + rrep + rerr;                               # Control Packets sent
    cp_lost = cp_sent - cp_rec;                                 # Control Packets lost
    if (data_agt_sent > 0) pdf =(data_agt_rec/data_agt_sent)*100;    # Packet Delivery %
    if (data_agt_rec > 0) nrl = ((cp_sent +cp_forw)/data_agt_rec)*100;   # Load
    lost = sent - rec;                                          # Packets lost
    data_lost = data_agt_sent - data_agt_rec;                   # Data Packets lost
    agt_lost = agt_sent - agt_rec;                              # Packets lost by Agent
    mac_lost = mac_sent - mac_rec;                              # Packets lost by Mac
    rtr_lost = agt_lost + cp_lost;                              # Packets lost by Router
    print "Data Analysis:";                                     # Print results.
    printf ("   Packets total:     %2d\n",totalNL);
    printf ("   Packets RTR:    %2d\n",rtr);
    printf ("   Packets MAC:   %2d\n",mac);
    printf ("   Packets Data AGT:       %2d\n",data_agt);
    printf ("\n");
    printf ("   Packets sent by Agent:  %2d\n", agt_sent);
    printf ("   Packets received by Agent:      %2d\n", agt_rec);
    printf ("   Packets forwarded by Agent:  %2d\n", agt_forw);
    printf ("   Packets dropped by Agent:     %2d\n", agt_drop);
    printf ("   Packets lost by Agent:             %2d\n", agt_lost);
    printf ("\n");
    printf ("   Data_Packets sent:                 %2d\n",data_agt_sent);
    printf ("   Data_Packets received:           %2d\n", data_agt_rec);
    printf ("   Data_Packets forwarded:         %2d\n", data_agt_forw);
    printf ("   Data_Packets dropped:            %2d\n", data_agt_drop);
    printf ("   Data_Packets lost:                 %2d\n", data_lost);
    printf ("\n");
    printf ("   Packet_Delivery Precent:        %3.2f\n", pdf);
    printf ("   Normalized_Routing Load(Percentage):        %3.2f\n", nrl);
    printf ("   Highest flow number:          %3.2f\n", highest_flow_id);
    printf("%6s %6s %8s %8s %8s %10s %12s %12s \n",   \
              "Node","sent","recpkts","forward","dropped ",  \
              "avgTput-kbps","avgDelay-ms",           \
              "totdatabytes")
        for (node_id = 0; node_id < max_nodes; node_id ++){
        printf("%6g %6s %8d %8g %8g %10g %12g %12g \n",      \
```

```
            node_id,snode[node_id],rnode[node_id],fnode[node_id],dnode[node_id],\
            (noderecsize[node_id]/simtime)*(8/1000),anode_delay[node_id]*1000,\
            noderecsize[node_id])
        }
        printf ("%6s %8s %8s %10s %10s %10s %12s %8s \n",      \
            "Totals","Sent","Received","Forwarded","Dropped ",   \
            "avgTput-kbps","avgDelay-ms", "Packets")
        printf("      %8g %8s %8d %10g %10g %12g %8g \n",       \
            data_agt_sent,data_agt_rec,fwdnode,dropped,             \
            ((recvdSize/simtime)*(8/1000)),(avg_delay*1000), pktcount)

######### Throughput ##############################################

   for ( flow_id = 0; flow_id <= highest_flow_id;flow_id ++ ) {
     start = start_time[flow_id];
     end = end_time[flow_id];
     if ( start <= end ){
          packet_duration = end - start;          # single distance
          duration_total += packet_duration;   # total duration
          flow_number ++;                              # flow number
       bits_total += $37;                             # Bits Total
     }
  }
  printf ("Total Number of Flows: %d\n",flow_number);
  thrgputPackets = flow_number / duration_total;
  thrgputBits    = bits_total / duration_total;

  printf ("Average Duration of packets per sec.:%f\n", thrgputPackets);
  printf ("Average Duration of Bits per sec.: %f\n", thrgputBits);
}
```

Appendix J

Sample Output of A-stat

20D-0-A-20-2		
Data	Analysis:	
Packets	total:	55172
Packets	RTR:	10642
Packets	MAC:	37575
Packets	AGT:	6954
Packets	Data_AGT:	6954
Data_Packets	sent:	3478
Data_Packets	received:	3476
Data_Packets	forwarded:	0
Data_Packets	dropped:	14
Data_Packets	lost:	2
Packet_Delivery	Precent:	99.94
Normalized_Routing	Load(%):	0

Node	sent	recpkts	dropped	avgTput-kbps	avgDelay-ms	Recdatabytes
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	770	0	15.7696	259.611	394240
3	0	0	0	0	0	0
4	712	0	0	0	0	0
5	462	363	10	7.43424	0.0312498	185856
6	0	811	0	16.6093	246.457	415232
7	308	0	0	0	0	0
8	0	308	0	6.30784	648.801	157696
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	189	0	0	0	0	0
13	0	189	0	3.87072	1057.95	96768
14	0	0	0	0	0	0
15	285	0	0	0	0	0
16	218	285	0	5.8368	0.884873	145920
17	0	218	0	4.46464	917.013	111616
18	816	0	2	0	0	0
19	488	532	2	10.8954	0.0463096	272384

Totals	Sent	Received	Dropped	avgTput-kbps	avgDelay-ms	TotBytecount
	3478	3476	14	71.1885	8.63691	0

Total	Number_of_Flows:	20		
Average_Duration	(packets	per	sec).:0.152995	
Average_Duration	(Bits	per	sec).:	6.731793

Appendix K

10 Node Series Data Collection

Run #	Target Protocol	# Target Protocol Nodes	Subject Protocol	# Subject Protocol Nodes	Mobility Model	Velocity	Total Packets Sent	Total Packets Received	Delivery Ratio	End-toEnd Delay (Latency)	Average E-2-E Delay
1	DSR	10	AODV	0	RW	1	2118	1182	55.81%	951	1001
2	DSR	10	AODV	0	RW	2	2106	1768	83.95%	278	617
3	DSR	10	AODV	0	RW	3	2100	1574	74.95%	145	1098
4	DSR	10	AODV	0	RW	4	2117	2079	98.21%	121	275
5	DSR	8	AODV	2	RW	1	2100	627	29.86%	702	997
6	DSR	8	AODV	2	RW	2	2107	891	42.29%	48	253
7	DSR	8	AODV	2	RW	3	2098	932	44.42%	118	1020
8	DSR	8	AODV	2	RW	4	2109	1142	54.15%	109	9
9	DSR	6	AODV	4	RW	1	2115	130	6.15%	18905	896
10	DSR	6	AODV	4	RW	2	2112	1210	57.29%	78	537
11	DSR	6	AODV	4	RW	3	2094	1354	64.66%	150	945
12	DSR	6	AODV	4	RW	4	2106	1786	84.81%	136	241
13	DSR	4	AODV	6	RW	1	2098	63	3.00%	18905	14
14	DSR	4	AODV	6	RW	2	2080	666	32.02%	195	509
15	DSR	4	AODV	6	RW	3	2098	771	36.75%	310	967
16	DSR	4	AODV	6	RW	4	2102	1049	49.90%	244	18
17	DSR	2	AODV	8	RW	1	2111	786	37.23%	455	284
18	DSR	2	AODV	8	RW	2	2094	1529	73.02%	75	337
19	DSR	2	AODV	8	RW	3	2113	1032	48.84%	97	389
20	DSR	2	AODV	8	RW	4	2118	1657	78.23%	74	10
21	DSR	0	AODV	10	RW	1	2093	1079	51.55%	1020	208
22	DSR	0	AODV	10	RW	2	2133	1795	84.15%	261	292
23	DSR	0	AODV	10	RW	3	2128	1482	69.64%	147	483
24	DSR	0	AODV	10	RW	4	2078	1999	96.20%	130	110

25	AODV	10	DSR	0	RW	1	2093	1079	51.55%	1020	208
26	AODV	10	DSR	0	RW	2	2133	1795	84.15%	261	292
27	AODV	10	DSR	0	RW	3	2128	1482	69.64%	147	483
28	AODV	10	DSR	0	RW	4	2078	1999	96.20%	130	110
29	AODV	8	DSR	2	RW	1	2116	604	28.54%	709	371
30	AODV	8	DSR	2	RW	2	2117	891	42.09%	47	45
31	AODV	8	DSR	2	RW	3	2118	876	41.36%	117	347
32	AODV	8	DSR	2	RW	4	2089	1116	53.42%	112	11
33	AODV	6	DSR	4	RW	1	2127	101	4.75%	18871	13
34	AODV	6	DSR	4	RW	2	2096	1180	56.30%	77	410
35	AODV	6	DSR	4	RW	3	2108	1277	60.58%	154	576
36	AODV	6	DSR	4	RW	4	2094	1675	79.99%	136	18
37	AODV	4	DSR	6	RW	1	2109	68	3.22%	15791	13
38	AODV	4	DSR	6	RW	2	2129	646	30.34%	190	684
39	AODV	4	DSR	6	RW	3	2127	735	34.56%	330	816
40	AODV	4	DSR	6	RW	4	1202	1052	87.52%	245	15
41	AODV	2	DSR	8	RW	1	2094	823	39.30%	429	1726
42	AODV	2	DSR	8	RW	2	2100	1647	78.43%	73	558
43	AODV	2	DSR	8	RW	3	2103	1083	51.50%	96	529
44	AODV	2	DSR	8	RW	4	2098	1661	79.17%	74	9
45	AODV	0	DSR	10	RW	1	2118	1182	55.81%	960	1001
46	AODV	0	DSR	10	RW	2	2106	1786	84.81%	279	617
47	AODV	0	DSR	10	RW	3	2100	1574	74.95%	145	1098
48	AODV	0	DSR	10	RW	4	2117	2079	98.21%	121	274

Appendix L

20 Node Series Data Collection

Run #	Target Protocol	# Target Protocol Nodes	Subject Protocol	# Subject Protocol Nodes	Mobility Model	Velocity	Total Packets Sent	Total Packets Received	Delivery Ratio	End-toEnd Delay (Latency)	Average E-2-E Delay
1	DSR	20	AODV	0	RW	1	4659	4655	99.91%	136	65
2	DSR	20	AODV	0	RW	2	4684	4683	99.98%	135	10
3	DSR	20	AODV	0	RW	3	4683	4544	97.03%	141	312
4	DSR	20	AODV	0	RW	4	4734	4724	99.79%	134	77
5	DSR	18	AODV	2	RW	1	4699	3916	83.34%	112	58
6	DSR	18	AODV	2	RW	2	4637	3873	83.52%	123	9
7	DSR	18	AODV	2	RW	3	4683	3712	79.27%	129	570
8	DSR	18	AODV	2	RW	4	4683	3838	81.96%	124	96
9	DSR	16	AODV	4	RW	1	4671	4171	89.30%	154	118
10	DSR	16	AODV	4	RW	2	4671	4595	98.37%	137	62
11	DSR	16	AODV	4	RW	3	4681	3850	82.25%	156	434
12	DSR	16	AODV	4	RW	4	4670	3989	85.42%	139	79
13	DSR	14	AODV	6	RW	1	4658	3389	72.76%	205	165
14	DSR	14	AODV	6	RW	2	4639	3866	83.34%	188	13
15	DSR	14	AODV	6	RW	3	4671	3429	73.41%	249	225
16	DSR	14	AODV	6	RW	4	4687	3228	68.87%	214	327
17	DSR	12	AODV	8	RW	1	4641	3338	71.92%	223	112
18	DSR	12	AODV	8	RW	2	4719	4279	90.68%	111	11
19	DSR	12	AODV	8	RW	3	4676	3094	66.17%	152	188
20	DSR	12	AODV	8	RW	4	4679	3574	76.38%	141	345
21	DSR	10	AODV	10	RW	1	4682	2700	57.67%	76	10
22	DSR	10	AODV	10	RW	2	4675	3747	80.15%	116	14
23	DSR	10	AODV	10	RW	3	4667	2710	58.07%	162	342
24	DSR	10	AODV	10	RW	4	4724	3396	71.89%	134	212
25	DSR	8	AODV	12	RW	1	4699	2651	56.42%	92	9
26	DSR	8	AODV	12	RW	2	4688	4091	87.27%	118	10
27	DSR	8	AODV	12	RW	3	4665	3226	69.15%	152	227
28	DSR	8	AODV	12	RW	4	4671	3556	76.13%	230	162
29	DSR	6	AODV	14	RW	1	4684	3238	69.13%	109	9

30	DSR	6	AODV	14	RW	2	4676	4253	90.95%	135	12
31	DSR	6	AODV	14	RW	3	4652	3807	81.84%	173	264
32	DSR	6	AODV	14	RW	4	4676	4024	86.06%	217	155
33	DSR	4	AODV	16	RW	1	4675	4097	87.64%	112	13
34	DSR	4	AODV	16	RW	2	4662	4253	91.23%	111	29
35	DSR	4	AODV	16	RW	3	4664	4088	87.65%	112	197
36	DSR	4	AODV	16	RW	4	4684	4203	89.73%	112	18
37	DSR	2	AODV	18	RW	1	4688	4406	93.98%	145	115
38	DSR	2	AODV	18	RW	2	4686	4680	99.87%	135	40
39	DSR	2	AODV	18	RW	3	4676	4482	95.85%	143	119
40	DSR	2	AODV	18	RW	4	4691	4564	97.29%	140	62
41	DSR	0	AODV	20	RW	1	4676	4664	99.74%	135	39
42	DSR	0	AODV	20	RW	2	4655	4653	99.96%	136	11
43	DSR	0	AODV	20	RW	3	4675	4486	95.96%	141	148
44	DSR	0	AODV	20	RW	4	4683	4649	99.27%	136	41
1	AODV	20	DSR	0	RW	1	4676	4664	99.74%	135	40
2	AODV	20	DSR	0	RW	2	4655	4653	99.96%	136	11
3	AODV	20	DSR	0	RW	3	4675	4486	95.96%	62	148
4	AODV	20	DSR	0	RW	4	4683	4649	99.27%	136	41
5	AODV	18	DSR	2	RW	1	4657	3804	81.68%	126	96
6	AODV	18	DSR	2	RW	2	4647	3878	83.45%	123	10
7	AODV	18	DSR	2	RW	3	4663	3602	77.25%	123	127
8	AODV	18	DSR	2	RW	4	4687	3767	80.37%	125	63
9	AODV	16	DSR	4	RW	1	4674	4225	90.39%	152	85
10	AODV	16	DSR	4	RW	2	4702	4658	99.06%	135	54
11	AODV	16	DSR	4	RW	3	4673	3745	80.14%	99	173
12	AODV	16	DSR	4	RW	4	4673	3924	83.97%	159	68
13	AODV	14	DSR	6	RW	1	4694	3449	73.48%	204	62
14	AODV	14	DSR	6	RW	2	4671	3888	83.24%	188	9
15	AODV	14	DSR	6	RW	3	4650	3319	71.38%	99	184
16	AODV	14	DSR	6	RW	4	4702	3245	69.01%	210	388
17	AODV	12	DSR	8	RW	1	4658	3518	75.53%	192	118
18	AODV	12	DSR	8	RW	2	4682	4244	90.65%	111	9
19	AODV	12	DSR	8	RW	3	4718	3108	65.88%	103	174
20	AODV	12	DSR	8	RW	4	4656	3657	78.54%	137	315
21	AODV	10	DSR	10	RW	1	4705	2724	57.90%	75	7
22	AODV	10	DSR	10	RW	2	4669	3735	80.00%	117	10
23	AODV	10	DSR	10	RW	3	4668	2703	57.90%	118	337
24	AODV	10	DSR	10	RW	4	4667	3266	69.98%	140	197
25	AODV	8	DSR	12	RW	1	4688	2638	56.27%	94	9
26	AODV	8	DSR	12	RW	2	4660	4068	87.30%	118	8
27	AODV	8	DSR	12	RW	3	4680	3290	70.30%	99	298
28	AODV	8	DSR	12	RW	4	4689	3639	77.61%	173	186
29	AODV	6	DSR	14	RW	1	4670	3220	68.95%	110	9
30	AODV	6	DSR	14	RW	2	4699	4698	99.98%	135	9
31	AODV	6	DSR	14	RW	3	4682	3933	84.00%	101	442

32	AODV	6	DSR	14	RW	4	4691	4071	86.78%	217	253
33	AODV	4	DSR	16	RW	1	4675	4097	87.64%	113	13
34	AODV	4	DSR	16	RW	2	4668	4267	91.41%	111	10
35	AODV	4	DSR	16	RW	3	4685	4268	91.10%	96	272
36	AODV	4	DSR	16	RW	4	4709	4216	89.53%	113	74
37	AODV	2	DSR	18	RW	1	4700	4515	96.06%	137	14
38	AODV	2	DSR	18	RW	2	4706	4706	100.00%	134	9
39	AODV	2	DSR	18	RW	3	4676	4504	96.32%	79	304
40	AODV	2	DSR	18	RW	4	4659	4630	99.38%	136	141
41	AODV	0	DSR	20	RW	1	4659	4655	99.91%	136	65
42	AODV	0	DSR	20	RW	2	4684	4683	99.98%	135	10
43	AODV	0	DSR	20	RW	3	4683	4544	97.03%	54	312
44	AODV	0	DSR	20	RW	4	4734	4724	99.79%	134	77

Appendix M

10 Node Series Results

Table M1. 10 node series average throughput percentage baseline statistics

Velocity	Mean	SD	SEM	High Limit	Low Limit
1 fps	0.5368008	0.0300844	0.041694	0.578495	0.495107
2 pfs	0.8426587	0.0014365	0.001991	0.842513	0.838531
3 fps	0.7229761	0.0375440	0.052032	0.775009	0.670944
4 fps	0.97201637	0.0141898	0.019666	0.991682	0.952351

Table M2. 10 node series throughput percentage results

	Velocity			
	1 fps	2 fps	3 fps	4 fps
High Limit	0.578495	0.842513	0.775009	0.991682
Low Limit	0.495107	0.838531	0.670944	0.952351
Protocol Mix	Average Throughput Percentages			
A2-D8	0.345799(T)	0.603581(T)	0.479605(T)	0.666597(T)
A4-D6	0.046854(T)	0.438173(T)	0.496083(T)	0.861631(T)
A6-D4	0.038757(T)	0.441585(T)	0.486640(T)	0.649477(T)
A8-D2	0.328890(T)	0.575530(T)	0.451001(T)	0.658284(T)

Table M3. 10 node end-to-end delay statistical analysis

Velocity	Mean	SD	SEM	High Limit	Low Limit
1 fps	310.2	20.14	27.92	338.08	282.24
2 fps	336.62	3.606	4.997	341.61	331.62
3 fps	373.89	15.95	22.10	395.99	351.79
4 fps	294.26	8.251	11.44	305.69	282.82

Table M4. 10 node end-to-end delay results analysis

	Velocity 1 fps	2 fps	3 fps	4 fps
High Limit	338.08	341.61	395.99	305.69
Low Limit	282.24	331.62	351.79	282.82
Protocol Mix				
A2 - D8	219.65(T)	233.55(T)	384.63(F)	274.41(T)
A4 - D6	1287.0(T)	432.49(T)	572.76(T)	400.99(T)
A6 - D4	1480.8(T)	427.84(T)	567.15(T)	411.54(T)
A8 - D2	228.87(T)	243.12(T)	406.72(T)	278.17(T)

Appendix N

20 Node Series Results

Table N1. 20 node average throughput percentage statistical analysis

Velocity	Mean	SD	SEM	High Limit	Low Limit
1 fps	0.9997	0.00041	0.00056	1.000	0.9991
2 fps	0.9986	0.00203	0.00282	1.000	0.9957
3 fps	0.9852	0.01687	0.02337	1.000	0.9618
4 fps	0.9951	0.00282	0.00390	0.9990	0.9912

Table N2. 20 node average throughput percentage results analysis

	Velocity			
	1 fps	2 fps	3 fps	4 fps
High Limit	1.000	1.000	1.000	0.9990
Low Limit	0.9991	0.9957	0.9618	0.9912
Protocol Mix				
A2-D18	0.8547(T)	0.8172(T)	0.7917(T)	0.8173(T)
A4-D16	0.7372(T)	0.6583(T)	0.6124(T)	0.6617(T)
A6-D14	0.6615(T)	0.5741(T)	0.5387(T)	0.5735(T)
A8-D12	0.7352(T)	0.6745(T)	0.4550(T)	0.6220(T)
A10-D10	0.7799(T)	0.7455(T)	0.5334(T)	0.7658(T)
A12-D8	0.7353(T)	0.6835(T)	0.4657(T)	0.6187(T)
A14-D6	0.6663(T)	0.5732(T)	0.5463(T)	0.5540(T)
A16-D4	0.7365(T)	0.6623(T)	0.5827(T)	0.6613(T)
A18-D2	0.8547(T)	0.8124(T)	0.7610(T)	0.8108(T)

Table N3. 20 node average end-to-end delay statistical analysis

Velocity	Mean	SD	SEM	High Limit	Low Limit
1 fps	286.28	1.8878	2.6164	288.90	283.67
2 fps	288.10	0.9683	1.3420	288.90	286.76
3 fps	291.64	5.1018	7.0706	298.71	284.57
4 fps	288.33	0.7011	0.9717	289.30	297.36

Table N4. 20 node average end-to-end delay results analysis

	Velocity 1 fps	2 fps	3 fps	4 fps
High Limit	288.90	288.90	298.71	289.30
Low Limit	283.67	286.76	284.57	297.36
Protocol Mix				
A2-D18	303.54(T)	323.48(T)	329.86(T)	341.71(T)
A4-D16	312.68(T)	360.77(T)	367.20(T)	377.18(T)
A6-D14	304.05(T)	348.31(T)	364.83(T)	351.16(T)
A8-D12	271.24(T)	294.62(T)	306.29(T)	354.98(T)
A10-D10	294.76(T)	308.46(T)	313.98(T)	299.55(T)
A12-D8	270.18(T)	293.96(T)	299.94(T)	325.17(T)
A14-D6	303.26(T)	349.29(T)	361.21(T)	360.68(T)
A16-D4	311.77(T)	357.62(T)	385.26(T)	377.61(T)
A18-D2	306.23(T)	326.76(T)	340.71(T)	342.98(T)

Appendix O

30 Node Series Results

Table O1. 30 node average throughput percentage statistical analysis

Velocity	Mean	SD	SEM	High Limit	Low Limit
1 fps	0.9760	0.02088	0.02894	1.0000	0.9471
2 fps	0.9988	0.00057	0.00079	0.9997	0.9981
3 fps	0.9965	0.00241	0.00334	0.9999	0.9932
4 fps	0.9699	0.01508	0.02090	0.9907	0.9489

Table O2. 30 node average throughput percentage results analysis

	Velocity			
	1 fps	2 fps	3 fps	4 fps
High Limit	1.0000	0.9997	0.9999	0.9907
Low Limit	0.9471	0.9981	0.9932	0.9489
Protocol Mix				
A2-D28	0.9449(T)	0.9533(T)	0.9533(T)	0.9402(T)
A4-D26	0.9415(T)	0.9998(T)	0.9836(T)	0.9499(F)
A6-D24	0.9406(T)	0.9055(T)	0.9559(T)	0.9306(T)
A8-D22	0.8280(T)	0.8847(T)	0.8608(T)	0.8612(T)
A10-D20	0.8355(T)	0.9224(T)	0.8753(T)	0.9015(T)
A12-D18	0.7451(T)	0.9222(T)	0.8897(T)	0.9055(T)
A14-D16	0.7527(T)	0.8979(T)	0.8353(T)	0.8763(T)
A16-D14	0.7353(T)	0.8981(T)	0.8325(T)	0.8751(T)
A18-D12	0.7415(T)	0.9197(T)	0.8868(T)	0.9022(T)
A20-D10	0.8227(T)	0.9211(T)	0.8704(T)	0.8982(T)
A22-D8	0.8069(T)	0.8827(T)	0.8563(T)	0.8509(T)
A24-D6	0.9205(T)	0.9565(T)	0.9501(T)	0.9285(T)
A26-D4	0.9237(T)	0.9988(T)	0.9810(T)	0.9565(F)
A28-D2	0.9207(T)	0.9470(T)	0.9495(T)	0.9371(T)

Table O3. 30 node average end-to-end delay statistical analysis

Velocity	Mean	SD	SEM	High Limit	Low Limit
1 fps	143.1	2.9924	4.1471	147.29	139.00
2 fps	140.4	0.3591	0.4977	140.87	139.87
3 fps	140.6	0.9135	1.2660	141.89	139.37
4 fps	143.6	1.6631	2.3046	145.90	141.29

Table O4. 30 node average end-to-end delay results analysis

	Velocity			
	1 fps	2 fps	3 fps	4 fps
High Limit	147.29	140.87	141.87	145.90
Low Limit	139.00	139.87	139.37	141.29
Protocol Mix				
A2-D18	135.99(T)	134.84(T)	134.37(T)	136.09(T)
A4-D16	137.05(T)	140.09(F)	142.09(T)	145.32(F)
A6-D14	136.39(T)	147.25(T)	146.73(T)	147.94(T)
A8-D12	157.92(T)	160.85(T)	166.91(T)	161.25(T)
A10-D20	156.28(T)	139.06(T)	147.57(T)	141.23(T)
A12-D18	172.76(T)	138.40(T)	145.28(T)	140.96(T)
A14-D16	154.42(T)	128.98(T)	139.48(F)	132.46(T)
A16-D14	157.88(T)	129.19(T)	140.56(F)	131.59(T)
A18-D12	174.15(T)	139.44(T)	145.46(T)	141.65(F)
A20-D10	158.22(T)	139.04(T)	148.27(T)	142.35(F)
A22-D8	162.32(T)	161.17(T)	167.19(T)	163.20(T)
A24-D6	139.34(F)	146.34(T)	147.76(T)	142.97(F)
A26-D4	139.73(F)	140.23(F)	143.12(T)	145.20(F)
A28-D2	139.07(F)	133.80(T)	134.60(T)	136.29(T)

References

Abolhasan, M., Lipman, J., & Chicharo, J. (2004). A routing strategy for heterogeneous mobile ad hoc networks. *Proceedings of the IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communications, Shanghai, China, 1,* 13-16.

Abolhasan, M., & Wysocki, T. (2003). A performance study of dynamic zone topology routing protocol. *Proceedings of the 7th International Symposium on DSP for Communications Systems, Coolangatta, Gold Coast*, 111-117.

Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2003, February). LPAR: An adaptive routing strategy for MANETs. *Journal of Telecommunication and Information Technology, 2*, 28-37.

Acharya, A., Misra, A., & Bansal, S. (2002). A label-switching packet forwarding architecture for multi-hop wireless LANs. *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia, USA,* 33-40.

Adler, M., & Scheidler, C. (1998). Efficient communication strategies for ad hoc wireless networks. *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures, Puerto Vallerta, Mexico*, 259-268.

Ahmed, M., Vanitchannant, N., & Dao, S. (2002). *A collaborative multicast routing (CMR) protocol for hybrid wireless networks.* Retrieved April 12, 2004, from http://www.ee.ucla.edu/~ameesh/papers/2232-342.pdf

Al-Shurman, M., Yoo, S-M., & Park, S. (2004). A performance simulation for route maintenance in wireless ad hoc networks. *Proceedings of the 42nd annual ACM Southeast Conference, USA,* 25-30.

Anastasi, G., & Lenzini, L. (2000). QoS provided by the IEEE 802.11 wireless LAN to advanced applications: a simulation analysis. *Wireless Networks, 6*, 99-108.

Arabshahi, P., Gray, A., Kassabalidis, I., Das, A., Narayanan, S., El-Sharkawi, M., & Marks II, R. (2001). Adaptive routing in wireless communication networks using swarm intelligence. *Proceedings of the 9th AIAA International Communication Satellite Systems Conference, Toulouse, France,* 256-265.

Bae, S., Lee, S., & Gerla, M. (2001). Multicast protocol implementation and validation in an ad hoc network testbed. *Proceedings of the Third International Workshop on Implicit Computational Complexity, Aarhus, Denmark*, 3196-3200.

Bai, F., Sadagopan, N., & Helmy, A. (2003). IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks. *Proceeding of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, USA,* 825-835.

Baran, B., & Sosa, R. (2001). AntNet routing algorithm for data networks based on mobile agents. *Revista Iberoamericana de Inteligencia Artificial, 12*, 75-84.

Beech, W., Nielsen, W., & Taylor, J. (1998). *AX.25 link access protocol for amateur packet radio.* Retrieved July 06, 2005 from http://www.tapr.org/pdf/AX25.2.2.pdf

Bhargava, B., Wu, X., Lu, Y., & Wang, W. (2004). Integrating heterogeneous wireless technologies: a cellular aided mobile ad hoc network (CAMA). *Mobile Networks and Applications, 9*(4), 393-408.

Boleng, J., Navidi, W., & Camp, T. (2002). Metrics to enable adaptive protocols for mobile ad hoc networks. *Proceedings of the International Conference on Wireless Networks, USA,* 293-298.

Bolinth, E., Lappeteläinen, A., Ojala, J., Pauli, M., Kreämling, A., Journé, T., Lacroix, D., Böhnke, R., Wegmann, B., & Schultz, E. (2001). QoS enhancements for HIPERLAN/2. *Proceedings of the IST Mobile Communications Summit, Barcelona, Spain,* 171-181.

Boukerche, A. (2004). Performance evaluation of routing protocols for ad hoc wireless networks. *Mobile Networks and Applications, 9*(4), 333-342.

Breslau, L., et al. (2000, May). Advances in network simulation. *IEEE Computer, 33*(5), 59-67.

Briesemeister, L. (2001). *Group membership and communications in highly mobile ad hoc networks.* Unpublished doctoral dissertation, Technischen Universitat, Berlin, Germany.

Bruno, R., Conti, M., & Gregori, E. (2001). WAN technologies for mobile ad hoc networks. *Proceedings of the 34th Hawaii International Conference on System Sciences, USA,* 9003-9014.

Buszko, D., Lee, W., & Helal, A. (2001). Decentralized ad-hoc groupware API and framework for mobile collaboration. *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, USA,* 5-14.

Buttyán, L., & Hubaux, J-P. (2003, October). Stimulating cooperation in self-organizing mobile ad hoc networks [Electronic version]. *Mobile Networks and Applications, 8*(5), 579-592.

Caire, G., Lhuillier, N., & Rimassa, G. (2002). A communications protocol for agents on handheld devices. *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems, Bologna, Italy*, 201-215.

Calafate, C., Garcia, R., & Manzoni, P. (2003). Optimizing the implementation of a MANET routing protocol in a heterogeneous environment. *Proceedings of the IEEE Eighth International Symposium on Computers and Communication, Kemer-Antalya, Turkey*, 217-225.

Camara, D., & Loureiro, A. (2000). A novel algorithm for ad hoc networks. *Proceedings of the 33rd Hawaii International Conference on System Science, USA, 8,* 8022-8030.

Camp, T., Boleng, J., Williams, B., Wilcox, L., & Navidi, W. (2002). Performance comparison of two location based routing protocols for ad hoc networks. *Proceedings of the twenty-first annual joint conference of the IEEE Computer and Communications Societies, USA, 3,* 1678-1687.

Cavin, D., Sasson, Y., & Schiper, A. (2002). On the accuracy of MANET simulators. *ACM Workshop on the Principles of Mobile Computing, Toulouse, France,* 38-43.

cygwin (2005). Software to emulate UNIX [Computer software and manual]. Retrieved February 16, 2005 from http://cygwin.com/

Chacón, D., Bell, B., & McCormick, J. (2000). Using autonomous agents to improve efficiency and robustness in slow, unreliable networks. *Proceedings of the American Association for Artificial Intelligence Spring Symposium, USA,* 34-36.

Chandramouli, V. (2002). *A detailed study on wireless LAN technologies*. Unpublished manuscript, University of Texas at Arlington.

Chen, T., & Gerla, M. (1998). Global state routing: A new routing scheme for ad-hoc wireless networks. *Proceedings of the IEEE International Conference on Communications, USA*, 171-175.

Choudhury, R., Paul, K., & Bandyopadhyay, S. (2002). An agent-based connection management protocol for ad-hoc wireless networks. *Journal of Network and Systems Management, 10*(4), 483-504.

Choudhury, R., Paul, K., & Bandyopadhyay, S. (2004). MARP: a multi-agent routing protocol for mobile wireless ad hoc networks. *Autonomous Agents and Multi-Agent Systems, 8*, 47-68.

Clausen, T., & Jacquet, P. (2003). *Optimized link state routing protocol*. Retrieved November 12, 2003 from http://www.ietf.org/ietf/1id-abstracts.txt

Cordeiro, C., Abhyankar, S., Toshiwal, R., & Agrawal, D. (2004). Bluestar: enabling efficient integration between bluetooth WPANs and IEEE 802.11 WANs. *Mobile Networks and Applications, 9,* 409-422.

Christin, N. (2005). *ns-2* all-in-one [Computer software and manual]. Retrieved February 15, 2005 from http://www.sims.berkeley.edu/~christin/ns-cygwin.shtml

Das, B., Perkins, C., & Royer, E. (2000). Performance comparison of two on-demand routing protocols for ad hoc networks. *Proceedings of the IEEE Conference on Computer Communications, Tel-Aviv, Israel*, *1,* 1-13.

Deb, A., Freburg, K., Surdu, J., Hall, A., & Maymi, F. (2002) Modeling the communications capabilities of the infantry soldier. *Communication Networks and Distributed Systems Modeling and Simulation Conference, USA*, 906-911.

Denko, M. (2003). The use of mobile agents for clustering in mobile ad hoc networks. *Proceedings of the South African Institute for Computer Scientists and Information Technologists*, *Pretoria, South Africa,* 241-247.

Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research, 9*, 317-365.

Di Caro, G., & Dorigo, M. (1998b). Ant colonies for adaptive routing in packet-switched communications networks. *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, *Amsterdam, Holland*, *1498*, 673-682.

Dougherty, D., & Robbins, A. (1997). *Sed & awk*. Sebastopol, CA: O'Reilly Media Inc.

Draves, R., Padhye, J., & Zill, B. (2004). Comparison of routing metrics for static multi-hop wireless networks. *Proceedings of the ACM conference on Applications, technologies, architectures, and protocols for computer communications, USA*, 133-144.

Dubendorf, V. (2003). *Wireless data technologies*. New York, NY: John Wiley & Sons.

Dunne, C.R. (2001). Using mobile agents for network resource discovery in peer-to-peer networks. *SIGecom exchanges, 2*(3), 1-9.

Flickenger, R. (2002). *Building wireless community networks*. Cambridge, MA: O'Reilly Press.

Garcia-Luna-Aceves, J., Mosko, M., & Perkins, C. (2003). A new approach to on-demand loop-free routing in ad hoc networks. *Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing, USA,* 53-62.

Gast, M. (2005). *802.11 Wireless Networks the Definitive Guide 2nd Edition*. Cambridge, MA: O'Reilly Press.

Ge, Y., Lamont, L., & Villasenor, L. (2005). Hierarchical OLSR – a scalable proactive routing protocol for heterogeneous ad hoc networks. *Proceedings of the Wireless and Mobile Computing Conference, Montreal, Canada*, 1-7.

Ghosh, A., Wolter, D., Andrews, J., & Chen, R. (2005). Broadband wireless access with WiMAX/802.16: Current performance benchmarks and future potential. *IEEE Communications Magazine, 43*(2), 129-136.

Gilbert, J., Choi, W-J., & Sun, Q. (2005). MIMO technology for advanced wireless local area networks. *Proceedings of the 42nd Annual Design Automation Conference, USA,* 413-415.

Goleniewski, L. (2002). *Telecommunications essentials*. Indianapolis, IN: Addison Wesley.

Golmie, N., Chevrollier, N., & Rebala, O. (2003). Bluetooth and WLAN coexistence: challenges and solutions. *IEEE Wireless Communications Magazine, 10*(6), 22-29.

Golmie, N., Van Dyck, R., Soltanian, A., Tonnerre, A., & Rebala, O. (2003). Interference evaluation of Bluetooth and IEEE 802.11b systems. *Wireless Networks, 9*, 201-211.

Günes, M., Sorges, U., & Bouazizi, I. (2002). ARA – the ant-colony based routing algorithm for MANETs. *Proceedings of the International Workshop on Ad Hoc Networking, British Columbia, Canada,* 79-85.

Haas, Z. (1997). A new routing protocol for the reconfigurable wireless networks. *Proceedings of the IEEE International Conference on Personal Communications, USA,* 562-566.

Haas, Z., Deng, J., Liang, B., Papadimitratos, P., & Sajama, S. (2002). Wireless ad hoc networks. *Encyclopedia of telecommunications* (John Proakis, Ed.). Ithaca, NY: John Wiley & Sons.

Haas, Z., & Pearlman, M. (2001). The performance of query control schemes for the zone routing protocol. *IEEE/ACM Transactions on Networking, 9*(4), 167-177.

Haas, Z., Pearlman, M., & Samar, P. (2003). *The zone routing protocol for ad hoc networks*. Retrieved November 05, 2004 from http://www.ietf.org/proceedings/

02nov/I-D/draft-ietf-manet-zone-zrp-04.txt

Hać, A. (2003). *Mobile telecommunications protocols for data networks*. Ithaca, NY: John Wiley & Sons.

Heissenbüttel, M., & Braun, T. (2003). Ants-based routing in large scale mobile ad-hoc networks. *Proceedings of the Kommunikation im verteilung Systemen, Leipzig, Germany,* 91-99.

Hong, X., Xu, K., & Gerla, M. (2002). Scalable routing protocols for mobile ad hoc networks. *IEEE Network, 6*(4), 11-21.

Hu, Y-C., Das, S., & Pucha, H. (2002). Exploiting the synergy between peer-to-peer and mobile ad hoc networks. *Proceedings of the Ninth Workshop on Hot Topics in Operating Systems, USA*, 1-6.

Hu, Y-C., & Johnson, D. (2000). Caching strategies in on-demand routing protocols for wireless ad hoc networks. *Proceedings of the ACM Conference on Mobile Ad Hoc Networking (MobiCom), USA*, 231-242.

Hu, Y-C., Perrig, A., & Johnson, D. (2002). Ariadne: A secure on-demand routing protocol for ad hoc networks. *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking, USA*, 12-23.

Huang, C-F., Lee, H-W., & Tseng, Y-C. (2004). A two-tier heterogeneous mobile ad hoc network architecture and its load-balance routing problem. *Mobile Networks and Applications, 9*, 379-391.

Hubaux, J-P., Gross, T., Boudec, J-Y., & Vetterli, M. (2001). Toward self-organized mobile ad hoc networks: The terminodes project. *IEEE Communications Magazine, 31*(1), 118-124.

IEEE (1999). *IEEE standard 802.11, 1999 edition*. New York, NY: Institute of Electrical and Electronics Engineers.

IEEE (1999b). *IEEE standard 802.11b-1999 supplement to ANSI/IEEE standard 802.11, 1999 edition*. New York, NY: Institute of Electrical and Electronics Engineers.

IEEE (2004). *IEEE 802.16-2004 Standard for Local and Metropolitan Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems.* New York, NY: Institute of Electrical and Electronics Engineers.

IEEE (2006). IEEE task group advances standards process for higher-speed 802.11™ wireless local area networks. Retrieved February 20, 2006 from http://standards.ieee.org/announcements/pr_tgp802.11n.html

Illmann, T., Krueger, T., Kargl, F., & Weber, M. (2001). Transparent migration of mobile agents using the JAVA platform debugger architecture. *Proceedings of the International Symposium on Mobile Agents, USA*, 198-212.

IrDA (1997, July 18). *Specifications for IrLAN.* Retrieved January 30, 2004 from http://www.irda.org/standards/pubs/IrLAN.PDF

Iwata, A., Chaing, C-C., Pei, G., Gerla, M., & Chen, T-W. (1999). Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications, 17*(8), 1369-1379.

Jardosh, A., Belding-Royer, M., Almeroth, K., & Suri, S. (2003). Towards realistic mobility models for mobile ad hoc networks. *Proceedings of the ACM Conference on Mobile Ad Hoc Networking, USA,* 217-229.

Ji, L., & Corson, S. (2001). Differential destination multicast – a MANET multicasting routing protocol for small groups. *Proceedings of the IEEE Conference on Computer Communications, USA*, 1192-2001.

Jiang, X., & Camp, T. (2002). *An agent-based location service for ad hoc networks.* Unpublished manuscript, Colorado School of Mines, Golden, CO.

Johnson, D. (1999). *Validation of wireless and mobile network models and simulation.* Unpublished manuscript, Carnegie Mellon University.

Johnson, D., Maltz, D., Hu, Y-C., & Jetcheva, J. (2002). *The dynamic source routing protocol for mobile ad hoc networks (Updated)*. Retrieved November 14, 2003 from http://www.ietf.org/ietf/1id-abstracts.txt

Jung, E-S., & Vaidya, N. (2002). A power control MAC protocol for ad hoc networks. *Proceedings of the 8th annual International Conference on Mobile Computing and Networking, USA,* 36-47.

Kain, C. (2003). *Commercial mobile radio services for public sector agencies*. Retrieved September 7, 2004 from http://www.itspublicsafety.net/docs/MitretekFinalCMRS_PublicSafety.pdf

Kannan, S., Mellor, J., & Kouvatsos, D. (2003). Investigation of distance vector routing for mobile ad hoc networks. *Proceedings of the Post Graduate Networking Conference, Liverpool, United Kingdom,* 1-4.

Kanter, T. (2003). Going wireless, enabling an adaptive and extensible environment. *Mobile Networks and Applications, 8*, 37-50.

Kargl, F., Nagler, J., & Schlott, S. (2002, July 23). *Building a framework for MANET routing protocols.* Paper presented at the 2002 Symposium on Mobile Ad-Hoc Netzwerk. Retrieved August 9, 2003 from http://medien.informatik.uni-ulm.de/~frank/research/manetframework.pdf

Kassabalidis, I., El-Sharkawi, M., Marks II, J., Arabshahi, P., & Gray, A. (2001). Swarm intelligence for routing in communications systems. *Proceedings of the IEEE Conference on Global Communications (Globecom), USA*, 3613-3617.

Kawaguchi, N., Toyama, K., & Inagakia, Y. (2000). MAGNET: ad hoc network system based on mobile agents. *Computer Communications, 23*(8), 761-768.

Kawaguchi, N., & Inagaki, Y. (2000). *Ad-hoc on-demand communications system based on mobile agents*. Retrieved January 07, 2003, from http://www.inagaki.nuie.nagoya-u.ac.jp/~kawaguti/research/wpmc2000.pdf

Kim, Y., Lee, J-J., & Helmy, A. (2004). Modeling and analyzing the impact of location inconsistencies on geographic routing in wireless networks. *Mobile Computing and Communications Review, 8*(1), 48-60.

Kotz, D., Gray, R., & Rus, D. (2002). *Future directions for mobile agent research* (Tech. Rep. TR2002-415). Hanover, NH: Dartmouth College.

Kuri, J., & Kasera, S. (2001). Reliable multicast in multi-access wireless LANs. *Wireless Networks, 7*, 359-369.

Kwon, T., & Gerla, M. (2002). Efficient flooding with passive clustering (PC) in ad hoc networks. *ACM SIGCOMM Computer Communications Review*, *32*(1), 44-56.

Lee, S., Han, B., & Shin, M. (2002). Robust routing in wireless ad hoc networks. *Proceedings of the IEEE International Conference on Parallel Processing Workshops (ICPPW), Vancouver, B.C., Canada,* 73-79.

Lee, S-J., Hsu, W., Gerla, M., & Bagrodia, R. (2000). A performance comparison study of ad hoc wireless multicast protocols. *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Tel Aviv, Israel*, 565-574.

Li, Q., Aslam, J., & Rus, D. (2001). Online power-aware routing in wireless ad-hoc networks. *ACM SIGMOBILE, Rome, Italy,* 97-107.

Li, Q., & Rus, D. (2000). Sending messages to mobile users in disconnected ad-hoc wireless networks. *Proceedings of ACM Mobile Computing and Networking Conference, USA,* 44-55.

Li, Q., & Rus, D. (2003). Communication in disconnected ad hoc networks using message relay. *Journal of Parallel Distributed Computing, 63*, 75-86.

Li, X., & Cuthbert, L. (2004). Node-disjointness-based multipath routing for mobile ad hoc networks. *Proceedings of the First ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks, Venezia, Italy,* 23-29.

Lin, G., Noubir, G., & Rajaraman, R. (2004). Mobility models for ad hoc network simulation. *Proceedings of the IEEE INFOCOM, Hong Kong, China,* 454-463.

Littman, M. (2002). *Building broadband networks*. Boca Raton, Florida: CRC Press.

Liu, W., & Song, H. (2002). Research and implementation of mobile ad hoc network emulation system. *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW), Vienna, Austria*, 749-756.

Lo, S-C., Liu, J-C., & Chen, W-T. (2004). Heterogeneous routing protocol for group communications in wireless ad hoc networks. *Proceedings of the 2nd International Conference on Information Technology for Application, Harbin, China,* 5-10.

Lu, Y., Wang, W., Zhong, Y., & Bhargava, B. (2003). Study of distance vector routing protocols for mobile ad hoc networks. *Proceedings of the IEEE International Conference on Pervasive Computing and Communications, USA*, 187-194.

Lundberg, D. (2004). *Ad hoc protocol evaluation and experiences of real world ad hoc networking.* Unpublished master's thesis, Uppsala University, Uppsala, Sweden.

Lundgren, H., Ramachandran, K., Belding-Royer, E., Almeroth, K., Benny, M., Hewatt, A., Touma, A., & Jardosh, A. (2006). Experiences from the design, deployment, and usage of the UCSB meshnet testbed. *IEEE Wireless Communications, 13*(2), 18-29.

Mangold, S., Choi, S., May, P., Klein, O., Heirtz, G., & Stibor, L. (2002). IEEE 802.11e wireless LAN for quality of service. *Proceedings of the European Wireless Conference*, Florence, Italy, February 2002.

Marina, M., & Das, S. (2001). On-demand multipath distance vector routing in ad hoc networks. *Proceedings of the Ninth International Conference on Network Protocols (ICNP'01)*, Riverside CA.

Marwaha, S., Tham, C., & Srinivasan, D. (2002). Mobile agent based routing protocol for mobile ad hoc networks. *Proceedings of the IEEE Symposium on Ad Hoc Wireless Networks, Taipei, Taiwan, R.O.C.,* 1-5.

Mauve, M., Widmer, J., & Hartenstein, H. (2001, November). A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine, 15*(6), 30-39.

Microsoft (2002). *Microsoft computer dictionary* (Fifth ed.). Redmond, WA: Microsoft Press.

Migas, N., Buchanan, W., & McArtney, K. (2003). Mobile agents for routing, topology discovery, and automatic network reconfiguration in ad hoc networks. *Proceedings of the 10th International Conference and Workshop on the Engineering of Computer Based Systems, USA,* 200-207.

Mochocki, B., & Madey, G. (2005). H-MAS: Heterogeneous, mobile, ad-hoc sensor-network simulation environment. *Proceedings of the Agent Directed Simulation Conference, USA,* 1-7.

Nam (2003). Nam: Network Animator [Computer software and manual]. Retrieved January 20, 2004 from http://www.isi.edu/nsnam/nam/

Naor, Z., & Levy, H. (2001). A centralized dynamic access probability protocol for next generation wireless networks. *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, USA*, 767-775.

Navid, N., Houda, L., & Bonnet, C. (2000). DDR: Distributed dynamic routing algorithm for mobile ad hoc networks. *First Annual Workshop on Mobile Ad Hoc Networking and Computing, USA,* 19-27.

Navidi, W., Bauer, N., & Camp, T. (2003). *Improving the accuracy of random waypoint simulations through steady-state initialization* (Tech. Rep. No. MCS-03-08). The Colorado School of Mines.

Navidi, W., & Camp, T. (2004). Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, *3*(1), 99-108.

Newton, H. (2006). *Newton's telecom dictionary* (22th ed.). New York, NY: CMP Books.

Nikaein, N., Bonnet, C., & Nikaein, N. (2001). Hybrid ad hoc routing protocol – HARP. *Proceeding of International Symposium on Telecommunications, Tehran, Iran*, 1-7.

Nikaein, N., Labiod, H., & Bonnet, C. (2000). DDR-distributed dynamic routing algorithm for mobile ad hoc networks. *Proceedings of the 1st International Symposium on Mobile Ad Hoc Networking & Computing, USA,* 19-27.

ns-2 (2005). Network simulator software (version 2.28) [Computer software and manual]. Retrieved February 18, 2005 from http://www.isi.edu/nsnam/ns/

Obraczka, K., Viswanath, K., & Tsudik, G. (2001). Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks, 7*(6), 627-634.

Ohrtman, F., & Roeder, K. (2003). *Wi-Fi handbook: Building 802.11b wireless networks*. New York, NY: McGraw-Hill.

Papadimitratos, P., & Haas, Z. (2003). Secure link state routing for mobile ad hoc networks. *Proceedings of the IEEE Workshop on Security and Assurance in Ad Hoc Networks, USA*, 379-383.

Papapetrou, E., & Pavlidou, F. (2003, October). A novel approach to source routing for multi-hop ad hoc networks. *IEEE Communications Letters, 7*(9) 472-474.

Park, S., & Park, D. (2004). Adaptive core multicast protocol. *Wireless Networks, 10*, 53-60.

Park, V., & Corson, S. (2001). *Temporally-ordered routing algorithm (TORA) version 1 functional specification*. Retrieved 12-1 2003 from http://www.ietf.org/ietf/1id-abstracts.txt

Patil, B., Saifullah, Y., Faccin, S., Sreemanthual, S., Aravamudhan, L. Sharma, S., & Mononen, R. (2003). *IP in wireless networks*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference.

Pei, G., & Gerla, M. (2001). Mobility management for hierarchical wireless networks. *Mobile Networks and Applications 6*, 331-337.

Pei, G., Gerla, M., Hong, X. (2000). LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. *First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc), USA*, 11-18.

Pei, G., Gerla, M., Hong, X., & Chaing, C-C. (1999). Wireless hierarchical routing protocol with group mobility (WHIRL). *Proceedings of the IEEE Wireless Communications and Networking Conference, USA,* 1-18.

Pei, G., Gerla, M., & Chen, T. (2000). Fisheye state routing: A routing scheme for ad hoc wireless networks, *Proceedings of the IEEE International Conference on Communication, USA*, 70-74.

Perkins, C., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. *1994 Conference on Communication Architectures, Protocols and Applications, USA,* 234-244.

Peterson, B., Baldwin, R., & Raines, R. (2003). Inquiry packet interference in bluetooth scatternets. *Mobile Computing and Communications Review, 7*(2), 66-75.

Pirzada, A., & McDonald, C. (2004). Establishing trust in pure ad-hoc networks. *Proceedings of the 27th Australasian Computer Science Conference, Dunedin, New Zealand,* 47-54.

Poon, E., & Li, B. (2003). SmartNode: Achieving 802.11 MAC interoperability in power-efficient ad hoc networks with dynamic range adjustments. *Proceedings of the 23rd International Conference on Distributed Computing Systems, USA,* 650-657.

Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P., & Zipl, A. (2001). CRUMPET: creation of user-friendly mobile services personalized for tourism. *Proceedings of the IEEE Second International Conference on 3G Mobile Communications Technologies, London, United Kingdom,* 28-32.

Prakash, R. (2001). A routing algorithm for wireless ad hoc networks with unidirectional links. *Wireless Networks, 7*, 617-625.

Puliafito, A., & Tomarchio, O. (2000). Using mobile agents to implement flexible network management strategies. *Computer Communications, 23*, 708-719.

Raju, J., & Garcia-Luna-Aceves, J. (2000). A comparison of on-demand and table driven routing for ad-hoc wireless networks. *Proceedings of the 2nd Annual Conference on Implicit Computational Complexity, USA*, 1702-1706.

Ramasubramanian, V., Haas, Z., & Sirer, E. (2003). SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks. *Proceedings of the ACM Conference on Mobile Ad Hoc Networks, USA,* 303-314.

Rangarajan, H., & Garcia-Luna-Aceves, J. (2004).Achieving loop-free incremental routing in ad hoc networks. *Proceedings of the IEEE International Symposium on Computers and Communications, Alexandria, Egypt, 1,* 182-187.

Ross, J. (2003). *The book of Wi-Fi – install, configure, and use 802.11b wireless networking*. San Francisco, CA: No Starch Press.

Roy, S., & Garcia-Luna-Aceves, J. (2002). Node-centric hybrid routing for ad hoc networks. *Proceedings of the IEEE International Mobility and Wireless Access Workshop, USA,* 63-72.

Royer, E., Sun, Y., & Perkins, C. (2001). *Global connectivity for IPv4 mobile ad hoc networks, 2001*. Retrieved December 1, 2003 from http://www.ietf.org/ietf/1id-abstracts.txt

Samar, P., Pearlman, M., & Haas, Z. (2004). Independent zone routing: An adaptive hybrid routing framework for ad hoc wireless networks. *IEEE Transactions on Networking, 12*(4), 595-608.

Santamaria, A., & Lopez-Hernindez, F. (2001). *Wireless LAN standards and applications*. Norwood, MA: Artech House.

Scherpe, C., & Wolf, J. (2002). Real-time simulation of multi-hop ad hoc networks. *Proceedings of the 6th CaberNet Radical Workshop, Madeira Island, Portugal*, 1-7.

Shen, C-C., & Jaikaeo, C. (2003). *Ad hoc multicast routing algorithm with swarm intelligence* (Tech. Rep. No. 2003-08). Newark, DE: University of Delaware.

Sheu, S-T., & Chen, J. (2002). Mr^2RP: The multi-rate and multi-range routing protocol for ad hoc wireless networks. *Proceedings of the Seventh International Symposium on Computers and Communications, Taormina, Italy,* 435-440.

Shurman, M., Yoo, S-M., & Park, S. (2004). A performance simulation for route maintenance in wireless ad hoc networks. *Proceedings of the ACM Southeast Conference, USA,* 25-30.

Sinha, S., Krishnamurthy, S., & Dao, S. (2000). Scalable unidirectional routing with zone routing protocol (ZRP) extensions for mobile ad hoc networks. *Proceedings of the IEEE Wireless Communications and Networking Conference, USA,* 1329-1339.

Sivakumar, R., Sinha, P., & Bharghavan, V. (1999). CEDAR: A core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks, 17*(8), 1454-1465.

Soltanian, A., & Van Dyck, R. (2001). Performance of the Bluetooth system in fading dispersive channels and interference. *Proceedings of the IEEE Global Telecommunications Conference, USA,* 3499-3503.

Spohn, M., & Garcia-Luna-Aceves, J. (2001). Neighborhood aware source routing. *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing, USA,* 11-21.

Solis, I., & Obraczka, K. (2004). FLIP: A flexible interconnection protocol for heterogeneous internetworking. *ACM/Kluwer Mobile Networks and Applications, 9*(4), 347-361.

Stoica, I. (2000). Ns-2 software and simulation results. Retrieved December 12, 2003, from University of California at Berkeley, Web site: http://www.cs.berkeley.edu/~istoica/csfq/software.html

Stojmenovic, I., & Lin, X. (2001). Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Computing, 12*(10), 1023-1032.

Sudame, P., & Badrinath, B. (2001). On providing support for protocol adaptation in mobile wireless networks. *Mobile Networks and Applications 6*(1), 43-55.

Sugar, R., & Imre, S. (2001). Adaptive clustering using mobile agents in wireless ad-hoc networks. *Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems, Lancaster, UK,* 199-204.

Suvak, D. (2000). *IrDA and Bluetooth: A complementary comparison*. Retrieved January 30, 2004 from http://www.techonline.com/community/tech_topic/bluetooth/tech_paper/20990

Tcl Developers Exchange (2005). Tcl and Tk software [Computer software and manual]. Retrieved January 4, 2005 from http://www.tcl.tk/software/tcltk/

The CMU Monarch Project (1999). The CMU wireless and mobility extensions to ns. Retrieved on March 16, 2005 from http://www.monarch.cs.cmu.edu/

Tian, J., Hahner, J., Becker, C., Stepanov, I., & Rothermel, K. (2002). Graph-based mobility model for mobile ad hoc network simulation. *Proceedings of the 35th Annual IEEE Simulation Symposium, USA,* 337-344.

Tseng, Y., Ni, S., Chen, Y., & Sheu, J. (2002). The broadcast storm problem in a mobile ad hoc network. *Wireless Networks, 8*(2/3), 153-167.

Valera, A., Seah, W., & Rao, SV. (2003). Cooperative packet caching and shortest multipath routing in mobile ad hoc networks. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, USA,* 260-269.

Vasilakopoulou, E., Karastergios, G., & Papadopoulos, G. (2003). Design and implementation of the hiperlan/2 protocol. *Mobile Computing and Communications Review, 7*(2), 20-32.

Viswanath, K., & Obraczka, K. (2002). An adaptive approach to group communications in multi-hop ad hoc networks. *Proceedings of the seventh International Symposium on Computers and Communications, Taormina, Italy,* 559-567.

Viswanath, K., Obraczka, K., & Tsudik, G. (2004). *Exploring mesh- and tree based multicast routing protocols for MANETS.* Retrieved July 11, 2005, from University of California at Santa Cruz, Inter-networking Research Group Web Site: http://inrg.cse.ucsc.edu/publications.html

Vrana, G. (2001, May 24). Wireless ethernet: Serving the public. *EDN*. Retrieved November 6, 2004 from http://www.edn.com/article/CA159700.html

Walsh, K., & Sirer, E. (2003). Staged simulation: A general technique for improving simulation scale and performance. *ACM Transactions on Modeling and Computer Simulation, 14*(2), 170-195.

Wang, J., Khokhar, A., & Garg, V. (2002). Video communication with QoS guarantees over HIPERLAN/2. *Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering, USA*, 312-319.

Wedde, H., Farooq, M., Pannenbaecker, T., Vogel, B., Mueller, C., Meth, J., & Jeruschkat, R. (2005). BeeAdHoc: An energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. *Proceedings of the Genetic and Evolutionary Computation Conference, USA,* 153-160.

Wheat, J., Hiser, R., Tucker, J., Neely, A., & McCullough, A. (2001). *Designing a wireless network*. Rockland, MA: Syngress Publishing.

White, B., Lepreau, J., & Guruprasad, S. (2003). Lowering the barrier to wireless and mobile experimentation. *ACM SIGCOMM Computer Communications Review, 33*(1), 47-52.

White, T., Pagurek, B., & Duego, D. (2002). Biologically-inspired agents for priority routing in networks. *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference, USA,* 282-287.

Wieselthier, J., Nguyen, G., & Ephremides, A. (2001). Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *Mobile Networks and Applications, 6*(3), 251-263.

Williams, B., & Camp, T. (2002). Comparison of broadcasting techniques for mobile ad hoc networks. *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland,* 194-205.

Woo, S-C., & Singh, S. (2001). Scalable routing protocol for ad hoc networks. *Wireless Networks, 7*(5), 513-529.

Wooldridge, M. (2002). *Introduction to multiagent systems*. Chichester, UK: John Wiley & Sons.

Xfree86 (2005). Xwindows software [Computer software and manual]. Retrieved January 5, 2005 from http://www.xfree86.org/

Xu, K., & Gerla, M. (2002). A heterogeneous routing protocol based on a new stable clustering scheme. *Proceedings of the Military Communications Conference, 2, USA,* 838-843.

Yang, Y., Zincir-Heywood, A., Heywood, M., & Srinivas, S. (2002). Agent-based routing algorithms on a LAN. *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering, Winnipeg, Manitoba, Canada*, 1442-1447.

Ye, Z., Krishnamurthy, S., & Tripathi, S. (2003). A framework for reliable routing in mobile ad hoc networks. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, USA,* 270-280.

Yoon, J., Liu, M., & Noble, B. (2003). Random waypoint considered harmful. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, USA,* 1312-1321.

Zhang, X., & Jacob, L. (2003). Adapting zone routing protocol for heterogeneous scenarios in ad hoc networks. *Proceedings of the 2003 International Conference on Parallel Processing, Kaohsiung, Taiwan,* 341-348.

Zou, X., Ramamurthy, B., & Magliveras, S. (2002). Routing techniques in wireless ad hoc networks – classification and comparison. *Proceedings of the 6th World Multi-conference on Systemics, Cybernetics & Informatics, IV, USA,* 1-6.