

THE FLORIDA STATE UNIVERSITY  
COLLEGE OF ARTS AND SCIENCES

RELIABILITY AND SECURITY IN MOBILE AD HOC NETWORKS

By

DONALD J SCOTT JR

A Thesis submitted to the  
Department of Computer Science  
in partial fulfillment of the  
requirements for the degree of  
Master of Science

Degree Awarded:  
Spring Semester, 2006

The members of the Committee approve the Thesis of Donald Scott defended on April 10, 2006.

Alec Yasinsac  
Professor Directing Thesis

Mike Burmester  
Committee Member

Breno de Medeiros  
Committee Member

Approved:

Dr. David B. Whalley, Chair, Department of Computer Science

The Office of Graduate Studies has verified and approved the above named committee members.

I dedicate this work to the amazing people in my life who have helped me to become the person that I am today, from family and friends to the chance encounters who inspired me. I would especially like to thank my parents for being wonderful role models and my sister for all her support.

## ACKNOWLEDGEMENTS

I would like to acknowledge all the assistance and advice that I have received over the years from faculty, staff, and my fellow students at Florida State University. In particular I would like to thank the security professors in the Computer Science department, especially Dr. Yasinsac.

## TABLE OF CONTENTS

List of Tables .....	vi
List of Figures .....	vii
Abstract .....	viii
1. INTRODUCTION .....	1
2. BACKGROUND .....	4
2.1 MANETs .....	4
2.2 MANET security .....	4
2.3 The Broadcast Storm .....	5
2.4 Density and Distribution .....	7
3. DYNAMIC GOSSIP .....	11
3.1 Density in the Overlap .....	11
3.2 Outside the Overlap and Beyond the First Hop.....	13
3.3 Perimeter Retransmissions .....	14
4. DYNAMIC GOSSIP WITH NON-UNIFORM DISTRIBUTION .....	17
5. TESLA .....	20
6. CLOCK SYNCHRONIZATION .....	23
6.1 Direct Synchronization .....	23
6.2 Indirect Synchronization .....	24
7. ATTACKING TESLA .....	26
7.1 Mobile Ad hoc Networks .....	26
7.2 Sensor Networks .....	29
8. CONCLUSIONS .....	31
REFERENCES .....	33
BIOGRAPHICAL SKETCH .....	35

## LIST OF TABLES

Table 1: Density-Driven Dynamic Gossip Computations .....	Page 14
Table 2: For Near-rim Transmissions .....	Page 15

## LIST OF FIGURES

Figure 1: A Flooding-optimal Network .....	7
Figure 2: A Network With Redundancy Under Flooding .....	7
Figure 3: A Disconnected Network .....	8
Figure 4: Retransmission Coverage Area .....	9
Figure 5: Overlap .....	12
Figure 6: Skewed .....	17
Figure 7: Connectivity .....	18
Figure 8: Direct Synchronization .....	21
Figure 9: TESLA Key Generation and Disclosure .....	23
Figure 10: Messages in the TESLA Attack .....	27
Figure 11: TESLA Attack .....	27

## ABSTRACT

Routing in mobile ad hoc networks presents many challenging problems not faced when routing in static networks with infrastructure. While much research has been done in this area, there is still much progress left to make before routing can be considered reliable and secure. This paper presents both research providing for more reliable routing as well as research showing how the TESLA protocol for secure routing can be attacked.

# CHAPTER 1

## INTRODUCTION

The recent trend in technology toward portable wireless devices has fueled research in the area of mobile ad hoc networks (MANETs). MANETs are networks composed of mobile computing devices, called nodes, which communicate wirelessly without the use of infrastructure. In order to communicate with nodes farther than one hop away, intermediate nodes must act as routers. While MANETs have many advantages over traditional wired networks, they also have a unique set of challenges. Besides communicating without infrastructure, nodes in ad hoc networks face limitations on their own resources. They have limited computational power, bandwidth, and power supply.

Despite these challenges, MANETs offer improved solutions to a number of difficult situations. Since they do not rely on infrastructure to communicate, they can be used in environments such as battlefields and disaster recovery scenarios. They also allow networks to be built on the fly at conferences and in classrooms for easy communication.

In order for MANETs to be an effective solution in these environments, they must have reliable, secure routing protocols. Since intermediate, possibly untrusted nodes may be used for forwarding messages, achieving security and reliability is a challenge. Protocols developed for communication in ad hoc networks have to account for both the limitations in resources of the nodes as well as the dynamic topology of the network. Besides moving around, nodes are also free to leave and join the network at any time. These problems make creating a robust, secure routing protocol difficult.

Over the past two years we have addressed difficulties in security and reliability. The first challenge we addressed was the problem of message delivery in a dense

network. In order to guarantee delivery to every node in a network, a node may use flooding, broadcasting the message and having every node that receives the message rebroadcast it. In dense networks flooding creates a storm of redundant messages leading to contention issues as well as wasting resources. One solution is the use of probabilistic broadcast protocols. These protocols resemble flooding but nodes only rebroadcast the message with a probability less than one. While this solution helps mitigate the problems created by flooding in a dense network, it does not deal with node mobility and the varied topology of MANETs. We propose a solution [1] in which the probabilistic broadcast protocol chooses what probability to broadcast with on the fly.

While flooding and probabilistic broadcast deal with the topic of reliability, MANETs also face challenges in the area of secure communication. The resource constraints on nodes in ad hoc networks limit the cryptographic measures that can be taken to secure messages. One proposed solution is TESLA, used in routing protocols such as Ariadne. TESLA uses hash chains and delayed key disclosure as a secure authentication mechanism. While studying Ariadne and TESLA, we took note of the assumption that nodes in the network are able to keep their clocks loosely synchronized within a certain bound. Given the inexpensive hardware typically used in the creation of nodes as well as the varied circumstances under which they are deployed, we found that this is not a safe assumption. We present an attack on TESLA [2] requiring only a limited clock skew outside of the assumptions made by the TESLA.

This paper presents those two notions. Section 2 discusses the challenges in creating a reliable broadcast protocol without flooding. Sections 3 and 4 review the Dynamic Gossip protocol presented in [1]. The second half of this paper deals with secure authentication as presented in [2]: Section 5 introduces the TESLA protocol [3, 4, 5, 6]. Section 6 deals some clock synchronization scenarios and section 7 details the attack on TESLA first shown in [2]. Section 8 presents our conclusions.

## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 MANETs**

Mobile ad hoc networks are networks formed without fixed infrastructure such as routers and gateways [7]. Instead normal nodes in the network acts as routers, forwarding messages towards their destination. The nodes themselves are mobile and thus routes between nodes may be constantly changing.

Since the nodes in MANETs are mobile they have limited resources. Nodes are bandwidth and energy constrained, with limited computational power. Routing protocols for MANETs must be able to deal with both the dynamic topology of the network as well as the resource constraints of the nodes. Much research has been done to develop reliable and efficient routing protocols [8, 9, 10, 11].

Sensor networks are similar to MANETs, although they lack mobility. Sensor networks are self organizing networks formed by small sensor nodes [12, 13]. Nodes are typically scattered in an environment to collect data, resulting in non-uniform distributions. These nodes are low cost and low power with the ability to sense the environment, process data, and communicate with other nodes. Nodes are designed to collect information from their environment and transmit it back through the network to a sink, or base station. Thus communication is routinely one way, from the nodes towards the base station. Sensor networks face many of the same routing challenges as MANETs, lacking in infrastructure and facing resource limitations.

#### **2.2 MANET Security**

The same characteristics of MANETs that make routing challenging also create

challenges for security. MANETs face both passive and active attacks [14]. The use of the wireless medium for transmitting messages allows adversaries to intercept transmissions. MANETs are also susceptible to active attacks, such as injecting false routing information, not routing packets, and flooding the network with traffic. Security solutions are needed that will scale well to large networks and will cooperate with the resource limitations of MANET nodes. Some routing protocols have been proposed to deal with these challenges [15, 16].

The mobility of nodes in ad hoc networks and the resulting route changes make detecting routing misbehavior difficult. In order to identify malicious routing behavior, a node must be able to tell the difference between malicious behavior and the normal routing changes characteristic of a mobile ad hoc network.

Secure authentication in traditional networks is done with public key cryptography and digital signatures. Nodes in MANETs lack the computational resources necessary to make traditional asymmetric cryptography an efficient security solution. Symmetric cryptography uses much more inexpensive cryptographic operations but ad hoc networks lack the infrastructure for distributing shared keys.

Key distribution and management is also a security issue in ad hoc networks. Nodes cannot rely on fixed points to be certificate authorities since nodes may leave the network at any time. Even with a node designated as the certificate authority, the other nodes in the network will not necessarily have a route to the certificate authority. In order to counter this problem, nodes in the network need a way to organize themselves without relying on a trusted authority or an initialization phase.

## **2.3 The Broadcast Storm**

Due to the lack of infrastructure in ad hoc networks, intermediate nodes are relied upon to act as routers. With mobile nodes acting as routers, paths are unreliable. A path that is used for one message may not be available for the next message. One

solution to this problem is for nodes to use a flooding algorithm [17]. Flooding involves the source node broadcasting the message and every node in the network broadcasting the message upon its receipt.

Although flooding is a simple, reliable protocol, guaranteeing that any node connected to the network will eventually receive the message, it has some disadvantages. Flooding creates many unnecessary messages, consuming valuable resources. Additionally, flooding also leads to contention, collision, and packet loss in dense networks. When the contention accelerates to the point of deadlock, this is referred to as the broadcast storm problem [18].

One counter to the broadcast storm problem is to only have nodes broadcast the original message with a chosen probability,  $p$ . This approach is referred to as probabilistic flooding [19, 20, 21]. Although probabilistic broadcast protocols (also termed gossip protocols) reduce the number of redundant messages, they do not inherently guarantee delivery. The challenge becomes deciding which messages are necessary and which can be eliminated.

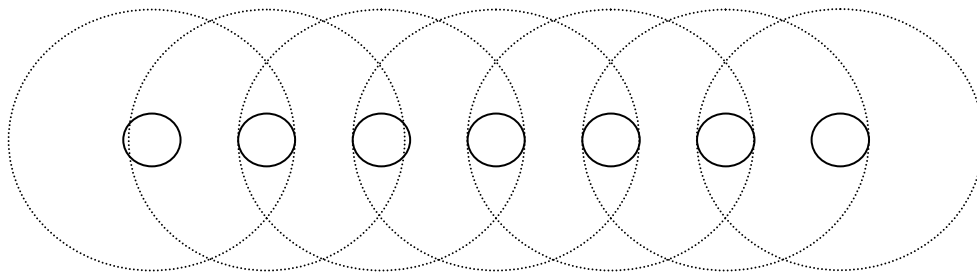
A simple gossip protocol starts with the source node broadcasting a message. When a node receives the message for the first time it broadcasts that message with probability  $p$ , between 0 and 1, reducing the overall number of messages transmitted. Although sending fewer messages reduces the problem of the broadcast storm, it also results in less reliability than using flooding. The choice of the  $p$  value is key in retaining reliability while minimizing the broadcast storm.

Haas et al. reason that an adequately large network will demonstrate bimodal behavior [19]. In this situation, there exists a threshold value for  $p$ . Choosing any value above  $p$  will result in nearly all nodes receiving the message whereas any value below  $p$  will result in almost none of the nodes receiving the message. A test network in [19] came up with an optimal value for  $p$  of .65. While running tests on different values of  $p$  to find the optimal value may work in mostly static network, that approach is not effective for mobile ad hoc networks.

The value for  $p$  must be selected large enough to propagate messages to the entire network, but small enough to avoid broadcast storms. Unfortunately, in mobile ad hoc networks, determining a single value of  $p$  for the entire network is unrealistic. Fixed networks with uniform distributions would be able to use a fixed  $p$  value obtained through trial and error. Mobile networks, though, may have dynamic topologies, with area being dense at times, sparse at others. This characteristic makes it unreasonable to use a static value for  $p$ . Thus we propose a dynamic protocol. The basis for the dynamic protocol stemmed in part from work analyzing broadcast zones of adjacent nodes [18]. The authors discussed the utility of retransmissions by nodes in the disjoint broadcast areas of the adjacent nodes. We leverage density information in overlap areas in order to find an effective retransmission probability.

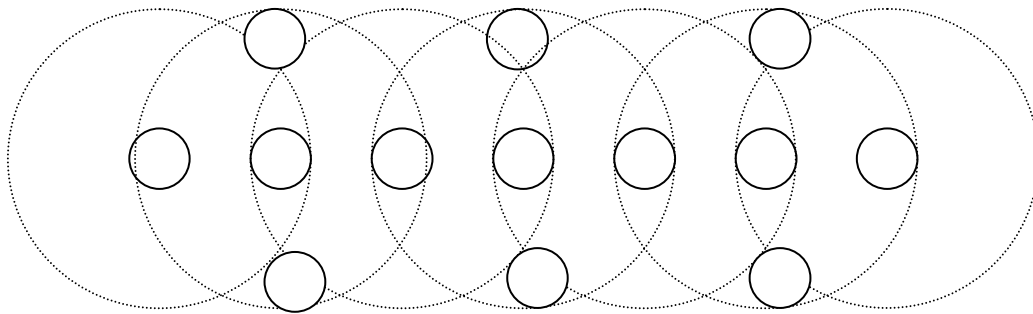
## 2.4 Density and Distribution

The primary factor influencing the broadcast storm is network density. The reason behind the contention, collision, and packet loss in the broadcast storm is too many nodes in the same broadcast area sending messages at the same time. In a less dense area, though, flooding could be the optimal transmission protocol. An example of this type of network is shown below in Figure 1. The only redundant transmissions in this example come from the end nodes.



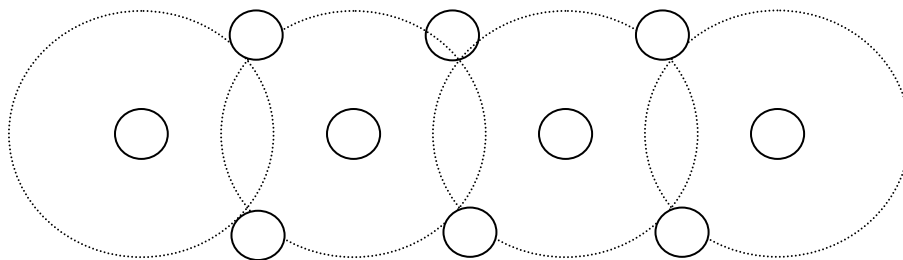
**Figure 1. A Flooding-optimal Network**

Adding nodes within the flooding optimal network shown in Figure 1 would result in redundant messages. Figure 2 shows such a network. If the additional nodes participate in the flooding algorithm they only add redundant messages in the network.



**Figure 2. A Network With Redundancy Under Flooding**

If nodes are removed from the flooding optimal network in Figure 1, the result is a disconnected network. Figure 3 shows a network that has more nodes than the network illustrated in Figure 1 but is still disconnected. Node density is not the only important concern in flooding; proximity of nodes also plays a role. While flooding guarantees message delivery in a connected network, it is not effective in a disconnected network.

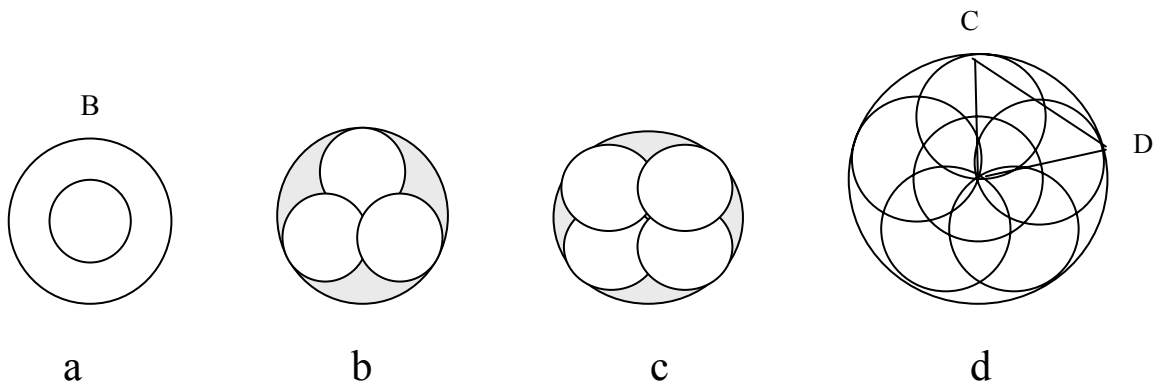


**Figure 3. A Disconnected Network**

While the above networks are important for pointing out properties of networks, we are interested in determining the retransmission probability for any arbitrary network. In order to determine a way to find  $p$  for an arbitrary network, we use analytical reasoning as opposed to computer simulations.

Before beginning, we make two assumptions about retransmission properties in the networks we are looking at. First, we only look at original and first hop retransmissions. Second, all links in our networks are bidirectional (every node has the same transmission range).

To begin we look at the maximum coverage area of one hop retransmissions. This is shown in Figure 4a. Circle A represents the broadcast range of a single node, circle B shows the maximum one hop retransmission range. If the radius of A is  $r$  then the radius of B is  $2r$ .



**Figure 4. Retransmission Coverage Area**

If we want to achieve maximum coverage with as few nodes as possible, what is the optimum placement for these nodes? The optimum placement would be equidistant apart on the circumference of the origin node's broadcast area, circle A. We illustrate this in Figures 4.b-d, with the areas not covered by any transmission a darker shade.

From this observation we can determine the optimal number of retransmissions that should result from the original broadcast. If no nodes rebroadcast, the area left uncovered is the difference between circles A and B,  $3\pi r^2$ . Instead of using a complex formula to determine the uncovered area when rebroadcasts are present, we approximate by creating an upper bound on the uncovered area.

We use the upper bound of the area between the arc and line ((C, D) in Figure 4.d) as the uncovered area. This area is the difference between the cone with apexes as C and D and the triangle with apexes at C and D and is greater than the area actually left

uncovered by the two overlapping retransmission areas. Seeing as there is one area for each retransmission, we bound the uncovered area (UA) with optimal node placement by the formula given in Equation 1. This equation is only dependent on the number of nodes,  $n$ , and the radius of the transmission range,  $r$ .

$$UA = 4r^2 * (\pi/n - \sin(\pi/n) * \cos(\pi/n) ) \quad (1)$$

By the above equation, ten optimally placed nodes will cover more than 93% of the rebroadcast area, with fifteen covering over 97% and 20 covering over 98%. In a relatively static, uniformly distributed network we would be able to use these numbers to determine the number of retransmissions needed to guarantee message delivery to all two hop neighbors with a high probability. In the next section we leverage this information to describe a dynamic gossip protocol.

## CHAPTER 3

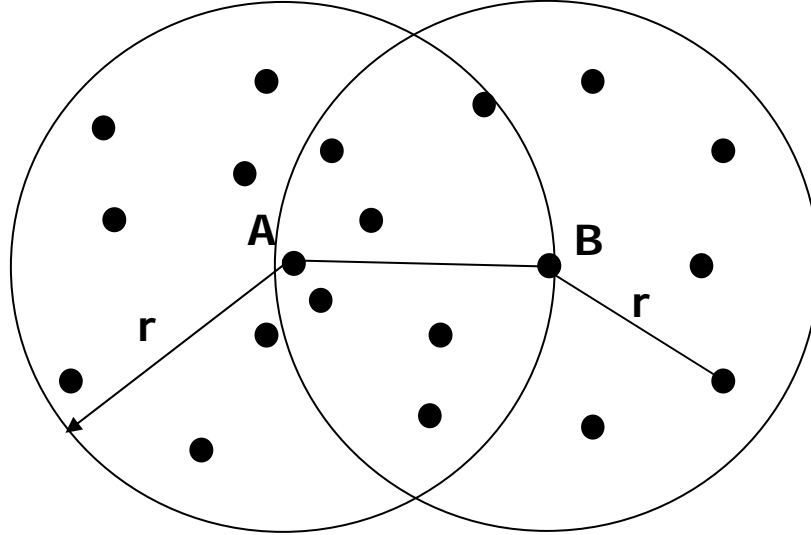
### Dynamic Gossip

Static probabilistic retransmission schemes are an effective way to reduce the broadcast storm problem in mostly static networks. In dynamic networks, though, static mechanisms are less effective. We propose a modification to static gossip that uses local node density information to choose retransmission rates and reduce redundant broadcasts. Upon broadcasting, each node will determine the retransmission probability of its' neighbors based on the number of local neighbors. This allows nodes to choose different broadcast probabilities based on the current layout of the network instead of relying on a single probability.

#### 3.1 Density in the Overlap

Although we are considering mobile networks, we begin our reasoning assuming fairly uniform distribution in the network. Under uniform distributions, we can know how many nodes occupy a certain area. This is important because we will be looking at the broadcast overlap area of two communicating nodes. By looking at the nodes that are in the broadcast area of two communicating nodes, we can determine an optimal rebroadcast probability.

Figure 5 presents an example network to help illustrate this discussion. Nodes A and B are at the limit of each other's transmission range. Since there are seven nodes in the overlap area, node A would choose a retransmission probability of one in seven, or .143, if it wanted one node in that area to rebroadcast the message.



**Figure 5. Overlap**

While this ratio can be deduced by inspecting the figure, we could also use the observations in the last section and our assumption of uniform distribution to arrive at .143 mathematically. By taking into account the total broadcast range, the overlap area between the nodes A and B, and the local density we can find the proper rebroadcast probability. This only works with the assumption that mostly uniform node distribution also means proportional node density. In other words, we can be sure that seven nodes in seven units of area generally means one node per unit of area.

Noting our assumptions, the proportional ratio (PR) of the overlap area to the total broadcast area can be computed. Our assumptions stated that all nodes in the network have bi-directional links, and thus equal broadcast areas. With equal, circular broadcast areas we can use the integral given in Equation 2 to find the overlap of two nodes at the limit of each others broadcast range. That ratio comes out to be  $.41\pi r^2$  [18].

$$PR = 4 \int_{r/2}^r \sqrt{r^2 - x^2} dx \quad (2)$$

Along with the proportional ratio, nodes also need to be aware of their local node density (LND). Nodes can find their broadcast area by measuring their transmission range (or having it pre-programmed). A nodes one hop neighbors can be found in many ways. A node may use a “hello” process, such as a short message (ping) at periodic intervals to determine one hop neighbors. LND is simply the number of one hop neighbors divided by the broadcast area.

The number of nodes in the overlap area (NNOA) is the local node density multiplied by the proportional relationship, so  $NNOA = LND * PR$ . From equation 2 we therefore have  $NNOA = LND * .41$  for our example. A node can now use NNOA to determine a retransmission probability that will have nodes in the overlap area retransmit the message without excess redundant messages. The source node chooses a desired number of rebroadcasts, DR, and divides this by the NNOA to find the retransmission probability. Thus  $p = DR / NNOA$ .

### **3.2 Outside the Overlap and Beyond the First Hop**

Now that we have found a way to control the number of retransmissions made in the overlap area, what happens in the rest of the broadcast area? Due to the assumed uniform distribution, the message will propagate the same amount in every direction. This can be seen by imagining that node B is on every side of the broadcast area. The overlap area will be the same size and, due to the assumption of uniform distribution, there will be an equal number of nodes in each overlap area. Thus, the message will propagate the same amount in every direction.

One of the added benefits of Dynamic Gossip is the ability to control the behavior of the message beyond the first hop. This allows the message originator to choose whether they are interested in reducing redundancy or enhancing reliability. A higher retransmission probability results in greater reliability while a lower probability results in lessening the effects of the broadcast storm. Every node that rebroadcasts the message has the ability to change the rebroadcast probability, effectively controlling the

message growth at each step. This feature does not require nodes to retain any additional state than normal flooding.

Computations in Table 1 illustrate our point. The value of  $p$  required for a desired number of retransmissions is dependant on the density of the network. With four nodes in the overlap area a probability of .25 is required to have a single retransmission whereas in a network with 39 nodes in the overlap area a probability of only .0256 is required for a single retransmission.

<b>Table 1 Density-Driven Dynamic Gossip Computations</b>				
LND	NNOA	p = Gossip Retransmission Probability		
		DR = 1	DR = 5	DR = 10
4	2	0.5000	1	1
10	4	0.2500	1	1
30	12	0.0833	0.4167	0.8333
100	39	0.0256	0.1282	0.2564
500	195	0.0051	0.0256	0.0513
1000	390	0.00256	0.01282	0.02564
10000	3900	0.00026	0.00128	0.00256

### 3.3 Perimeter Retransmissions

In ad hoc networks it cannot be presumed that the rebroadcasting nodes will be placed optimally. Using geometry, the added information can be discovered dealing with node proximity. Since we are focusing on generally uniformly distributed networks and known network density we can choose an area on the perimeter of the broadcast area instead of the overlap area used previously.

Although we have some control through the chosen probability of how many nodes rebroadcast, we cannot choose whether or not those nodes are near or far from the source. In order to overcome this we choose an area around the edge of the circle of width epsilon. Now we use the proportion of the area of the rim compared to the overall broadcast area to determine the retransmission probability. We use Equation 3 to

determine  $p$ , with  $r$  as the broadcast range,  $n$  as the number of desired retransmissions, and the rim width as  $r*(1-\epsilon)$ .

$$p = n / (1 - \epsilon^2)*LND, 0 < \epsilon < 1 \quad (3)$$

Table 2 provides examples using Equation 3. As before, the larger the number of retransmissions desired the larger the value of  $p$  necessary. As the values of  $\epsilon$  increase, decreasing the width of the rim, the values of  $p$  rise to compensate. Of note is the third row of Table 2, where the value of  $p$  is larger than one for the desired number of retransmissions. A single broadcast by the source would be unable to achieve the desired number of retransmissions.

<b>Table 2. P For Near-rim Transmissions</b>				
$p$	$n$	LND	$\epsilon^2$	$\epsilon$
0.263158	5	100	0.81	0.9
0.526316	10	100	0.81	0.9
1.052632	20	100	0.81	0.9
0.026316	5	1000	0.81	0.9
0.052632	10	1000	0.81	0.9
0.105263	20	1000	0.81	0.9
0.098039	5	100	0.49	0.7
0.196078	10	100	0.49	0.7
0.392157	20	100	0.49	0.7
0.009804	5	1000	0.49	0.7
0.019608	10	1000	0.49	0.7
0.039216	20	1000	0.49	0.7

These observations show that a source node has the ability to control numerous characteristics of the one hop retransmissions, such as the reach of one hop retransmissions.

### **3.4 State Retention**

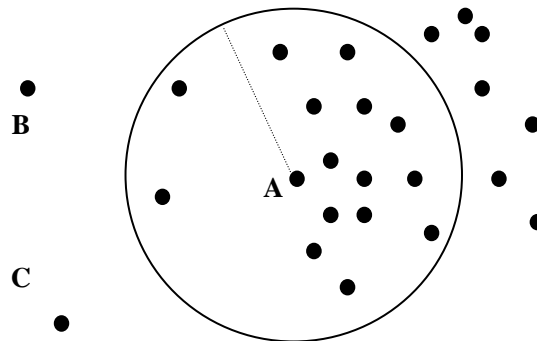
In flooding algorithms nodes are required to keep state information about previously received messages, whether through message sequence numbers or some other means. Since Dynamic Gossip requires all nodes to discard every copy of messages received after the first, nodes are required to retain state information. We leave the choice of which algorithm to use up to the implementers of the protocol.

Aside from the state information required to recognize previously received messages, Dynamic Gossip does not require the retention of any other state information. Nodes must be programmed with their transmission range in order to calculate the retransmission probability. The local node density is also required for this computation but can be obtained on demand.

## CHAPTER 4

### DYNAMIC GOSSIP WITH NON-UNIFORM DISTRIBUTION

Up to this point we have assumed a uniform distribution in networks with Dynamic Gossip. While this assumption is useful for describing the basics of Dynamic Gossip and showing its' properties, it is often an unrealistic assumption. Given the dynamic nature of mobile ad hoc networks, they are unlikely to keep uniform distribution for very long, if at all. One of the main difficulties with using a gossip protocol in ad hoc networks is the varying topologies. Static gossip protocols do not deal well with changes from dense node coverage to sparse. To deal with these properties of ad hoc networks we must make some modifications to the Dynamic Gossip approach.

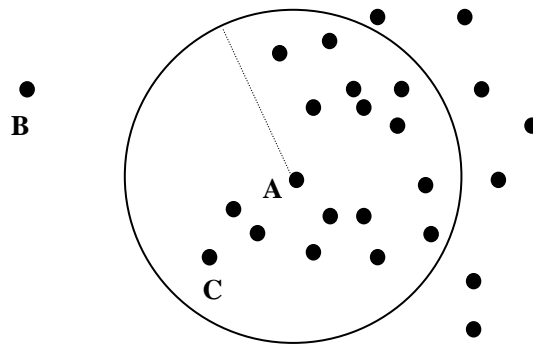


**Figure 6. Skewed**

Figure 6 illustrates a network with a skewed distribution. Node A has a LND of 15, although 13 of those nodes are in one half of the broadcast area. Let's assume that node A wants to make slight reduction from flooding and only have ten nodes retransmit (66% of the nodes in broadcast range). Node A broadcasts the message and sets the one hop retransmission rate to .667. With two neighbors in range Node B will receive the message 89% of the time but Node C will only receive the message 67% of the

time.

In applications requiring a high level of deliverability, a 67% reception rate could be unacceptable. This scenario shows the problem with using a gossip protocol in ad hoc networks. Skewed networks can result in inconsistent, unreliable delivery. In fact, it is well known that it is very difficult to express quantitatively the probability that each node will receive a message under any probabilistic protocol. Simulations can provide some insights about particular networks, but cannot provide consistent figures for arbitrary networks.



**Figure 7. Connectivity**

Mobility and network skew create another problem for message distribution. What happens if a node becomes temporarily disconnected from the network? Figure 7 above provides an example. This Figure shows Figure 6 after nodes have moved. Now, node C is more likely to receive messages because she has more neighbors, but node B is out of range of every node in the network. Even with flooding a message would not reach node B. In order to deal with this problem, nodes in the network would need to add a store and forward mechanism to their broadcast algorithm. In this algorithm, nodes would hold on to messages for a given amount of time after broadcasting. Under certain conditions they would rebroadcast the message in an attempt to reach nodes that had been disconnected from the network. The details of

such an algorithm are beyond the scope of this work as we are focusing on dense networks.

In sparse networks the purpose of probabilistic protocols works against them. Reducing the total number of messages sent results in messages not reaching their targets instead of reducing the effects of the broadcast storm problem. This problem will be addressed in the future, as the purpose of this paper was simply to create a protocol that dealt with the issues of the broadcast storm. Again, we have laid the theoretical foundations for delivering messages with a probabilistic scheme in a dense network, handling messages in a sparse network has been left for future research.

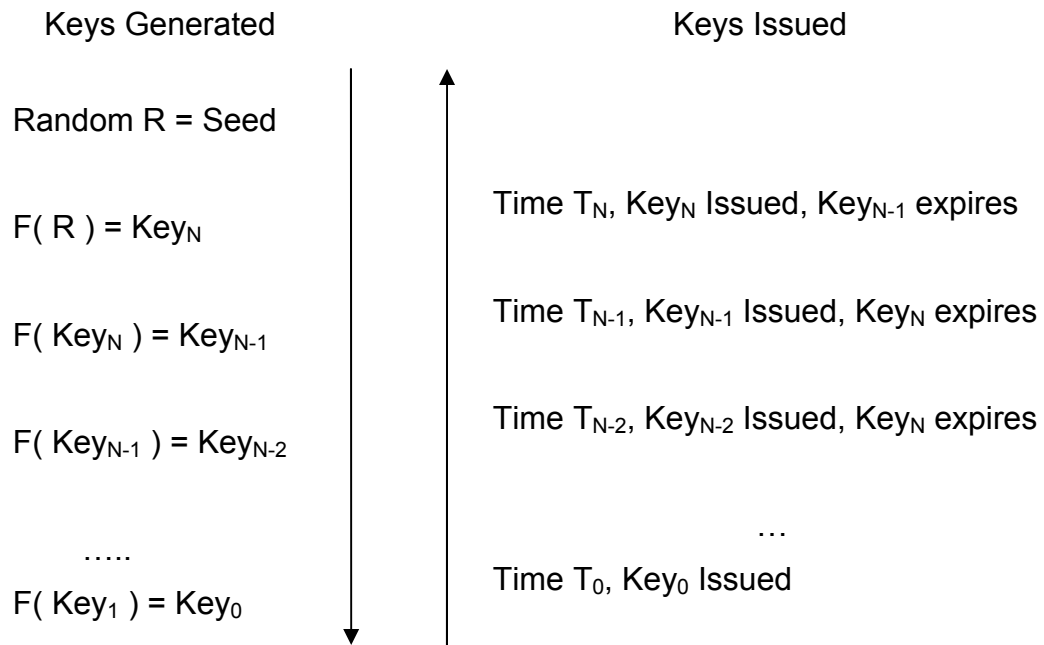
## CHAPTER 5

### TESLA

One proposed solution to the problem of secure authentication in mobile ad hoc networks is the Ariadne protocol [3]. This protocol makes use of TESLA, an authentication protocol for ad hoc networks created by Perrig, Canetti, Tygar, and Song [3, 4, 5, 6]. TESLA aims to provide authentication in ad hoc networks without adding expensive computations or message overhead. A single message authentication code (MAC) computation is required for both the message sender and receiver, adding only about 20 bytes per packet [4]. The computational burden on nodes is reduced by using symmetric encryption and delayed key disclosure.

In TESLA, each node transmitting messages begins by choosing the final key,  $K_N$ , of their hash chain. The node then uses a random function  $F$  and  $K_N$  to create a one-way hash chain. The hash chain is computed such that  $K_{N-1} = F[K_N]$ ,  $K_{N-2} = F[K_{N-1}]$ , and so on. In general,  $K_i = F^{N-i}(K_N)$ . Each value of  $K$  becomes a key, and every node in the network can verify a new key if they have the function  $F$  and an earlier key. Since  $F$  is one way nodes are not able to efficiently find a key which has not been released yet from an earlier key. Nodes must have a way to obtain an initial value of the key chain which they can verify is bound to the sender.

Figure 8 provides an illustration of key generation and disclosure in TESLA. Nodes begin by choosing a random number as a seed and can quickly compute a one-way key chain using function  $F$ . Messages sent during the first time interval will be authenticated with a MAC computed using  $Key_0$ . At the end of the first time interval  $Key_0$  will be published and MACs will now be computed using  $Key_1$ . Since  $Key_0$  has been released, any messages with a MAC computed using  $Key_0$  and received after  $T_0$  cannot be guaranteed to be authentic.



**Figure 8. TESLA Key Generation and Disclosure**

Since TESLA uses delayed key disclosure, nodes must be aware of the time intervals for which each key is intended. If a node sends out messages during interval  $T_i$  using key  $K_i$ , each node receiving that message must know of that correlation. Keys are only valid for a single time period. Receivers are able to verify messages sent during a time period if they can be sure that the key associated with the message has not yet been released by the source. A packet received during time period  $T_c$  with a MAC computed with key  $K_a$  would not be accepted, assuming time interval  $T_a$  came before  $T_c$ . It is possible that a malicious node has already received key  $K_a$ , created a new message, and computed a MAC using key  $K_a$ .

In order for the time interval scheme to work, nodes in the network must have loosely synchronized clocks. Without synchronized clocks it would not be possible for the receiving node to determine the current time interval and thus whether or not the key has been released yet. Nodes are not required to have fully synchronized clocks but they must be able to determine the maximum clock difference between any two nodes

in the network,  $\Delta$ . Nodes must also be aware of a pessimistic upper bound on maximum network latency,  $\tau$ . Nodes can use these two values to determine the allowable time interval during which a receiver will receive a message. Thus senders are able to choose a key to use which the receiver will know has not been published yet. The authors of [3] suggest using a time interval at least  $\tau + 2\Delta$  in the future, ensuring that when the packet reaches the destination node there is still at least  $\Delta$  before the sender releases the key.

Upon receiving a packet, the destination node first checks that the key used to create the MAC on the packet has not been published yet. If the time interval the message was keyed for has passed, the node drops the packet. Otherwise, the destination node puts the packet into a buffer until the key for that message is disclosed. When the node receives the key it first validates the key using the function  $F$  and a previous hash chain value from the sending node. If verification succeeds, the node uses that key to verify the MAC on any messages in the buffer associated with that key.

## CHAPTER 6

### Clock Synchronization

One of the foundations of the TESLA protocol is the assumption of loose clock synchronization. Clock synchronization is necessary in TESLA due to the use of delayed key disclosure. While clock synchronization in wired networks is a well studied area [22, 23, 24], research is still being done on clock synchronization algorithms for ad hoc networks [25, 26, 27]. Ad hoc network characteristics, such as possible partitioning of the network, create additional challenges for clock synchronization. For clarity we now review the two clock synchronization schemes suggested in [3], discussing the application of each in the TESLA protocol.

#### 6.1 Direct Synchronization

In direct synchronization nodes send messages directly between one another in order to synchronize their clocks. A two message protocol to accomplish direct clock synchronization is detailed in [4] and shown in Figure 9. The receiver, node R, sends a synchronization request to the sender, node S, with a nonce. Node R also records the local sending time,  $t_R$ . The sender sends back the local sender time,  $t_S$ , and the nonce, signed.



**Figure 9. Direct Synchronization**

When node R receives the response from node S, it can compute the maximum time difference between the nodes as  $\Delta = t_S - t_R$ . The network delay in sending the

messages make the value of  $\Delta$  greater than the actual difference between the two clocks.

The messages must be sent in a secure way, such as using digital signatures. They would be small, though, so the computations would not be overly expensive. The only problem would be determining how often these messages had to be sent. If they are required fairly often the main benefits of using TESLA, namely low computational cost and overhead, are negated.

## 6.2 Indirect Synchronization

Instead of using messages sent directly between nodes, a third party could be involved in clock synchronization. In ad hoc networks this third party could be an agreed upon node in the network or nodes could be equipped with GPS devices.

The problem with using a single node as a point of reference for other nodes in the network is resiliency. The common node becomes a single point of failure as well as a bottleneck and secure authentication may become unavailable if the reference node was unavailable, for instance due to a DoS attack. Also, authentication in the entire network could become compromised if the node itself was compromised. In addition, the issues of choosing the reference node and choosing a new reference node if the current node leaves the network adds unnecessary complexity to the protocol.

GPS offers a viable solution [28]. Nodes could synchronize themselves with the GPS and periodically send out signed packets containing their time synchronization with the time reference. One disadvantage though is that GPS synchronization adds hardware requirements to nodes wishing to participate in secure communication in the network. GPS also suffers from line of sight and atmospheric condition problems. Differences in atmospheric conditions between nodes could lead to a GPS signal taking time  $x$  to reach node A while taking time  $y$  to reach node B. This would not be much of a worry in a small network where the atmospheric conditions will be the same for all nodes, but as networks become increasingly large this would be a concern. Line of sight would be a

problem for nodes in big cities, buildings, or areas in which the local geography prevents satellite communication. Lack of line of sight could leave certain nodes locked out of the network just the same as a DoS attack on a reference node. These factors would need to be considered when determining how nodes in the network would loosely synchronize their clocks. In certain scenarios these concerns would be negligible while in other scenarios they could be determining factors.

# CHAPTER 7

## Attacking TESLA

### 7.1 Mobile Ad hoc Networks

The creators of the TESLA protocol make several questionable assumptions in [3] about the networks and nodes utilizing TESLA. They disregard attacks on the physical layer, assume bidirectional links, assume that the network may drop, corrupt, reorder, or duplicate packets, and assume that nodes are able to determine the maximum network delay,  $\tau$ . A key assumption they make, though, is that nodes have loosely synchronized clocks differing by at most  $\Delta$ . No assumptions are made as to how the nodes will accomplish clock synchronization or about any specific hardware running on the nodes.

The use of loose time synchronization and delayed key disclosure leaves the TESLA protocol open to the attack first described in [2]. This attack assumes non-negligible clock drift. The amount of clock drift is not assumed to be of some objective magnitude, it is instead based on the choice of network variables  $\Delta$  and  $\tau$ .

We outline the attack by looking at the Ariadne protocol discussed in [3] and the algorithm for route discovery. Route discovery packets contain the time interval at which the packet is intended to reach the destination and, therefore, which key has been used to create the MAC on the message. As discussed earlier, this time interval is based on the value  $\tau$ , the pessimistic upper bound on network latency. The information in the packet is not encrypted, only signed, and thus a malicious node can read the entire packet, including the time interval. We give an attack on clock synchronization against TESLA that compromises authentication in Figure 10.

Message 1: S -> M: {  $M_j$  |  $MAC(K_i, M_j)$  }

M: intercept  $M_j$  wait.

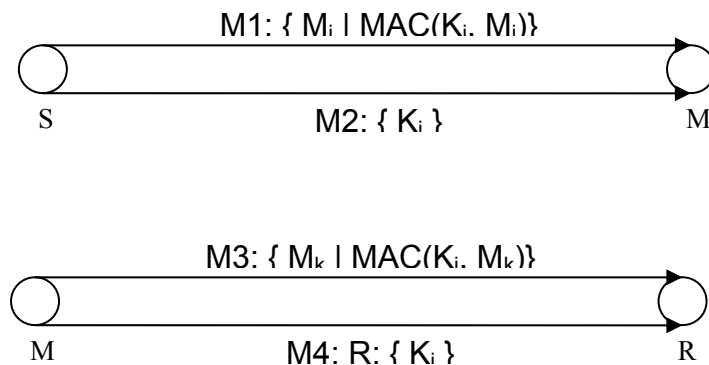
Message 2: S -> M: {  $K_i$  }

Message 3: M -> R: {  $M_k$  |  $MAC(K_i, M_k)$  }

Message 4: M -> R: {  $K_i$  }

**Figure 10. Messages in the TESLA attack**

In this example the malicious node is assumed to be the only intermediate node between the sender and receiver to make the illustration and explanation of the attack easier. The attack would work if there were other nodes in the path, before, after, or before and after the malicious node.



**Figure 11. TESLA Attack**

Figure 11 illustrates the messages passed in the attack. The source node, node S, begins by broadcasting an initial setup message to the receiver. This message contains the beginning of a time interval,  $T_j$ , the length of a time interval, and the delay before the key for time period  $T_j$  is released. The malicious node, node M, takes note of all this information and stores it for later use.

At some later time, the node M receives a message destined for the receiver, node R, during time interval  $T_x$ . The malicious node does not forward the message on to R.

Node M is aware of the time intervals as well as the time when node S will disclose the key,  $K_x$ . Node M waits until node S releases the key and then uses  $K_x$  to create a MAC on a new packet that node M created. Node M then forwards this packet onto node R, followed by the packet containing key  $K_x$ .

According to the TESLA protocol an attack of this type should fail. The receiving node should receive the packet from node M and determine that the time interval has passed and thus the packet is invalid. If node R receives all messages in what it believes to be correct time intervals, it will believe that the message created by node M originated from node S.

In order for this attack to work, the clock drift between nodes S and R must be more than negligible. According to the assumptions made by the creators of the TESLA protocol, this would not happen. The TESLA parameters would be set so that the key disclosure delay would be longer than the maximum time synchronization error and the pessimistic upper bound on network latency combined. Of note, the tighter the parameters are set the easier it is for the nodes' clocks to drift to greater than  $\tau + \Delta$ . The smaller these two values are set the less distance clocks would need to skew to go beyond those values.

In order for this attack to work certain conditions are necessary. The clock skew between the sending node and the receiving node must be large enough that it is greater than the maximum synchronization error plus the time necessary for the malicious node to receive the key for the old time interval, create a MAC on a message, and send the message to the receiving node. Under these conditions, when the receiving node checks the time interval of the message it will still appear to be the correct time interval and thus a valid message. In a network situation like the one in the above example, with the malicious node at the only node between the source and destination, the least clock skew would be necessary for the attack to succeed. The greater the distance between the malicious node and destination the longer a transmission will take and thus the greater the clock skew necessary for the attack to

succeed, though greater clock drift is also possible.

We emphasize that under the assumptions made in the [3, 4, 5, 6] this attack will fail, since clock drift would be negligible. The assumptions made about clock drift are weak, though, as research has shown that clock drift is not negligible [26, 27]. Also, if clock drift is non-negligible but clock re-synchronization happens often enough, the attack becomes less effective. Performing clock synchronization adds overhead, though, negating some of the benefit of using TESLA.

A key part is how much clocks will drift. One of the benefits of ad hoc network is the ability for nodes to be constantly moving in and out of the network. In order for the network to be as general as possible, no restrictions have been placed on hardware required for nodes to join the network. One of the benefits of mobile ad hoc networks is that they are not restricted to a single environment or a single type of hardware. Both the quality of hardware and environmental conditions affect clock drift. Environmental factors such as changes in pressure and temperature as well as hardware factors such as changes in voltage impact clock drift [28]. These type of factors make influence whether or not it is safe to assume that clock drift in an ad hoc network is negligible.

## **7.2 Sensor Networks**

A special type of ad hoc network is the sensor network. These networks consist of tiny sensor devices that use wireless connections to form themselves into a network, forwarding sensor data on to a base station, or sink. Sensor networks are self forming networks which are static except for the occasional death of nodes. Nodes only communicate one way, sending messages to the base station. Network topology is typically non-uniform and nodes have varying battery life. Nodes in sensor networks face even stricter resource constraints than normal ad hoc networks. In order to provide secure authentication in sensor networks, a new form of TESLA was created and called  $\mu$ TESLA [6].

Since the basic structure of  $\mu$ TESLA is the same as that of TESLA,  $\mu$ TESLA is still susceptible to the attack described above. Not only is  $\mu$ TESLA still vulnerable, the risk

of an attack against a sensor network is higher than that of an ad hoc network with more robust hardware. Research has been done on clock drift in sensor networks [26, 27]. This research has shown that clock drift in sensor network is non-negligible. Part of the reason for drift is the inexpensive hardware used in sensor networks, such as simple crystal oscillators [28]. While inexpensive, crystal oscillators are liable to larger drift than more expensive timing mechanisms. With hardware inclined to greater drift, the attack outlined above has a greater chance of succeeding in sensor networks, the attacker having time to compute the MAC and forward the message with the destination still thinking it is in the old time interval.

## CHAPTER 8

### Conclusions

Mobile ad hoc networks have virtually limitless applications. They offer the ability to setup networks on the fly in harsh environments where it would be infeasible to deploy a traditional network infrastructure. While ad hoc networks have vast potential, there are many challenges left to overcome. MANETs lack a fixed infrastructure for routing and routes constantly change due to node mobility. Nodes must also overcome constraints on their resources such as computational power, bandwidth, and power supply. Routing protocols must become more reliable and message transmissions must be secure.

We present an approach for ensuring reliable transmission in mobile ad hoc networks. While message delivery can be guaranteed to nodes in a connected network by using a flooding algorithm, flooding has disadvantages. Dense networks are subject to the effects of the broadcast storm when flooding is used. The Dynamic Gossip protocol alleviates the problems caused by the broadcast storm by reducing the total number of messages sent. It also addresses some of the problems presented by networks with skewed node distributions by allowing each transmitting node to dynamically adjust the retransmission probability.

Due to the MANET constrained environment properties, there are many approaches to solving MANET deliverability and security issues. One such approach, the TESLA protocol, aims to facilitate secure authentication in ad hoc networks while retaining low computation and bandwidth requirements. To achieve these benefits, TESLA uses loose time synchronization and delayed key disclosure. We show how an inherent weakness of loose time synchronization can lead to an attack that leverages clock skew. While the TESLA protocol is secure against this attack under the assumptions stated in [3, 4, 5, 6], research has shown that clock skew is more than negligible, especially in sensor networks [26, 27], making those assumptions weak. The presence

of non-negligible clock skew leaves a network using the TESLA protocol open to attack.

We have presented a new message delivery protocol that transmits messages to the destination with a high rate of success while reducing the excessive redundant messages present in the flooding algorithm. We have also shown the dangers in relying on time synchronization between nodes in an ad hoc network. Due to environmental conditions and node hardware, using even loose time synchronization may leave the network vulnerable. We have detected and proposed these solutions through collaborative peer-reviewed research [1, 2].

## REFERENCES

- [1] Donald Scott and Alec Yasinsac, "Dynamic Probabilistic Retransmission in Ad Hoc Networks", Proceedings of the International Conference on Wireless Networks, June 21-4, 2004
- [2] Donald Scott "Relying On Time Synchronization for Security in Ad-hoc Networks." Proceedings of the 43rd ACM Southeast Conference (ACMSE 2006). Kennesaw State University, Kennesaw, GA. 18-20 March 2005.
- [3] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In Proc. MobiCom'02, Atlanta, GA, Sept. 2002.
- [4] Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. Efficient and Secure Source Authentication for Multicast. In Network and Distributed System Security Symposium, NDSS '01, pages 35–46, February 2001.
- [5] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In IEEE Symposium on Security and Privacy, pages 56–73, May 2000.
- [6] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MobiCom 2001), pages 189–199, July 2001.
- [7] MANET IETF working group. [www.ietf.org/html.charters/manet-charter.html](http://www.ietf.org/html.charters/manet-charter.html).
- [8] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, April 1999, pp. 46-55.
- [9] Charles E. Perkins and Elizabeth M. Royer. "[Ad hoc On-Demand Distance Vector Routing.](#)" *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.
- [10] D.B. Johnson and D.A. Maltz, Dynamic Source Routing in Ad-Hoc Wireless Networks, *Mobile Computing*, ed. T. Imielinski and H. Korth, Kluwer Academic Publisher, pp. 152-181, 1996.
- [11] Y.B. Ko and N.H. Vaidya, Location-Aided Routing in Mobile Ad Hoc Networks, *Proceedings of ACM/IEEE MOBICOM '98*, 1998.
- [12] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar. [SPINS: Security Protocols for Sensor Networks.](#) *in Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM 2001*, July 2001.
- [13] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod Varshney. [A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks.](#) *In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, Washington DC, October 27-31, 2003.
- [14] Mike Burmester, Tri van Le and Alec Yasinsac. Weathering the storm: managing redundancy and security in ad hoc networks. *Proceedings of the 3rd International*

*Conference on AD-HOC Networks & Wireless*, Vancouver, British Columbia, pp. 96--107, July 22-24, 2004.

- [15] Panagiotis Papadimitratos and Zygmunt J. Haas [Secure Routing for Mobile Ad hoc Networks](#) SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002.
- [16] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru and Herbert Rubens [An On-Demand Secure Routing Protocol Resilient to Byzantine Failures](#) In *ACM Workshop on Wireless Security (WiSe)*, Atlanta, Georgia, September 28 2002
- [17] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. IP Flooding in ad hoc networks. Internet draft (draft-ietf-manet-bcast-00.txt), Nov 2001. Work in progress.
- [18] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 151-162, Aug 1999
- [19] Zygmunt J. Haas, Josep Y. Halpern, and Li Li. Gossip-based ad hoc routing. In IEEE INFOCOM, Jun 2002.
- [20] Yoav Sasson, David Cavin, and Adnre Schiper. Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks. In Proc. of IEEE WCNC 2003.
- [21] J. Cartigny and D. Simplot, "Border Node Retransmission Based Probabilistic Broadcast Protocols in Ad-Hoc Networks." In Proc. 36<sup>th</sup> International Hawaii International Conference on System Sciences (HICSS'03), Hawaii, USA. 2003.
- [22] D. L. Mills. Improved algorithms for synchronizing computer network clocks. In Conference on Communication Architectures (ACM SIGCOMM'94), London, UK, August 1994. ACM.
- [23] B. Simons, J. Welch, and N. Lynch. An overview of clock synchronization. Technical Report RJ 6505, IBM Almaden Research Center, 1988
- [24] Network Time Synchronization Bibliography. [www.eecis.udel.edu/~mills/bib.htm](http://www.eecis.udel.edu/~mills/bib.htm)
- [25] J. Elson and D. Estrin. Time Synchronization for Wireless Sensor Networks. In 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, USA, April 2001.
- [26] Qun Li and Daniela Rus. Global clock synchronization in sensor networks. In InfoCom 2004, Hong Kong, March 2004. IEEE.
- [27] Saurabh Ganeriwal, Ram Kumar, Mani B. Srivastava, Timing-sync protocol for sensor networks, Proceedings of the first international conference on Embedded networked sensor systems, November 05-07, 2003, Los Angeles, California, USA
- [28] Trimble Navigation Limited. Data Sheet and Specifications for Trimble Thunderbolt GPS Disciplined Clock. Sunnyvale, California. Available at <http://www.trimble.com/thunderbolt.html>.

## **BIOGRAPHICAL SKETCH**

Donald Scott was born in Philadelphia, Pennsylvania on February 22, 1982. He graduated with a bachelor's degree in Computer Science from Florida State University in 2004. After graduation he began working on his master's degree in Computer Science with a focus on Information Security.