The Pennsylvania State University

The Graduate School

College of Engineering

# PERFORMANCE ANALYSIS OF END-TO-END SMALL SEQUENCE NUMBERS ROUTING PROTOCOL

A Thesis in

Computer Science and Engineering

by

Jang Young Kim

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2010

The thesis of Jangyoung Kim was reviewed and approved* by the following:

Kyusun Choi
Assistant Professor of Computer Science & Engineering
Thesis Advisor

John Metzner
Professor of Computer Science & Engineering
Co-Advisor

Mahmut Kandemir
Professor of Computer Science & Engineering
Graduate Program Chair

*Signatures are on file in the Graduate School

# ABSTRACT

In networking communication, there are several nodes and stations such as base station or destination nodes. For communication among each node, they send packets or data to each nodes and destination nodes. In this perspective, it is very important to determine the way or direction that packets or data are sent to each of the nodes or destination nodes. Thus the data or packet transmission efficiency is based on routing direction. One of the standard routing protocols is an ad-hoc routing protocol which determines the way the packets are sent to their destination.

In an ad-hoc routing protocol, there are several kinds of protocols such as DSR (Dynamic Source Routing) and AODV (Ad-hoc On-demand Distance Vector). In the proposed protocol based on small end-to-end sequence numbers, route direction can be changed properly with the assistance of helper nodes. Between each station, there are helper nodes which guide the proper routes efficiently.

In this thesis, we focus on the simulation analysis of the proposed protocol and comparison with other routing protocol models such as AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing). We simulated using parameters such as simulation time, number of nodes, number of connections and packet size. The performance criteria will be packet delivery fraction (reliability), routing load (routing overhead) and data throughput.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## ACKNOWLEDGEMENTS

I sincerely appreciate that Prof. Kyusun Choi and Prof. John Metzner have given me opportunity to write a Master thesis in the department of Computer Science & Engineering at the Pennsylvania State University at University Park. I was also permitted to design and implement software using electronic devices in Prof. Kyusun Choi's lab.

I also appreciate Prof. Mahmut Kandemir's kind attention to my master's thesis. Thanks to Mr. Kiran Kashalkar for an opportunity to extend his work. Additionally, special thanks to numerous authors of networking references and other participants in this thesis.

# Chapter 1

# INTRODUCTION

There are several lists of ad-hoc routing protocols in Mobile Ad-hoc Networks (MANET) [1]. Also there are two representative categories which are pro-active and reactive (on-demand) protocols. Pro-active protocols refresh routes periodically throughout the network. One example of pro-active protocols is DSDV (Destination Sequenced Distance Vector) based on Bellman Ford algorithm which computes shortest paths in a weighted graphs [4]. The DSDV (Destination Sequenced Distance Vector) protocol is suitable for small number of nodes but not proper for dynamic networks. The disadvantages of this pro-active protocol are large amount of data and slow response for maintenance. On the other hand, reactive (on-demand) protocol sends packets based on the packet requests. Thus the reactive protocol minimizes the extra consumptions and it is more efficient.

As we know, the re-active protocol is more efficient than pro-active protocol especially with large number of nodes and traffic network communication. The proposed protocol (end-to-end small sequence numbers) [5] is motivated by the disadvantages of the re-active protocol and pro-active protocol. The two representative protocol AODV (Ad-hoc On-demand Distance Vector) [2] and DSR (Dynamic Source Routing) [3] are based on route discovery and maintenance.

The above two representative protocols such as AODV (Ad-hoc On-demand Distance Vector) [2] and DSR (Dynamic Source Routing) [3] are trying to transmit data or packets with reliability. However, when the routes failed or other problems occurred on route maintenance then it will be inefficient to use those routing protocols. Furthermore, in the Mobile Ad-hoc Network (MANET), the route maintenance mechanism works dynamically and routes need to be changed often when a link failed. At this point, we can consider a more dynamic and efficient protocol which is based on end-to-end small sequence numbers [5]. Sequence numbers are ordered numbers arranged in segmented packets or data that transmitted. For example, we need to send 100 Bytes of data through 10 Byte Internet connections. In order to send the data, we need to partition into 10 segments and sent by 10 Bytes each 10 times. After transmission, we need to re-order those 10 segments by sequence numbers. Thus it might be more efficient when it is based on small sequence numbers in this case.

In addition, the small size of sequence numbers prevents overhead and extra unnecessary resources. From another perspective, this proposed protocol can be energy-efficient and have a good data reliability mechanism. There is another protocol called TCP (Transmission Control Protocol) [6]. TCP (Transmission Control Protocol) is not a routing protocol. TCP is one of the standard protocols of Internet protocol. TCP uses a bytes stream transfer mechanism with large 32 bit sequence numbers which also can prevent ambiguity. To reduce the overhead, there is a strong motivation to design an efficient protocol based on small size of sequence numbers.

The basic concept of the efficient proposed protocol is motivated by the draft paper by Dr. John Metzner [10]. This paper indicates an efficient wireless routing to a base station or multi-base network using window controls and acknowledgement protocols. It also shows an end-to-end acknowledgement path and multiple-path sequence number limit protocol. The proposed protocol description will be introduced in Chapter 3.

In the Chapter 2, we will describe background work including description of AODV (Ad-hoc On-demand Distance Vector) [2] and DSR (Dynamic Source Routing) [3]. The experimental results and performance & comparison analysis will be described in Chapter 5. Finally, a conclusion and discussion is given in Chapter 6.

# Chapter 2

# BACKGROUND

In the previous chapter, we described several lists of ad-hoc routing protocols to be continued with the proposed protocol based on small sequence numbers. We can use these routing protocols to send data or packet with reliability and efficiency. Thus it is very crucial to enhance efficiency and speed in wireless network communication. We will describe this protocol in detail in this chapter.

One of the popular pro-active protocols is DSDV (Destination Sequenced Distance Vector) [4]. DSDV is a table-driven routing mechanism for ad-hoc mobile networks based on the Bellman-Ford algorithm. However, the re-active protocol is more efficient for dynamic mobile ad-hoc routing network such as AODV (Ad-hoc On-demand Distance Vector) [2] and DSR (Dynamic Source Routing) [3]. In addition to that, we can think about the proposed protocol based on the small size of sequence numbers which is more appropriate for a dynamic mobile network [5]. Besides, we may refer to other protocols such as TORA (Temporally Ordered Routing Algorithm) [19] and B.A.T.M.A.N (Better Approach To Mobile Ad-hoc Networking) [18]. TORA is an adaptive situation-aware routing algorithm for mobile ad-hoc routing networks. B.A.T.M.A.N is also proper for dynamic mobile ad-hoc routing networks but it does not use end-to-end sequence numbers.

DSR (Dynamic Source Routing) [3] is the first representative routing protocol of a re-active protocol. It is similar to AODV (Ad-hoc On-demand Distance Vector) [2] in the sense that it also based on packet requests. However, DSR uses source routing instead of routing table. DSR mechanism accumulates the address of each device between source and destination during route discovery. The path information is cached by node processing. The paths are used to route packets. In order to do source routing, the packets contain the address of each device the packet will traverse. This might result in larger overhead or large addresses. To avoid this problem, DSR forwards packets on a hop basis optionally. In DSR (Dynamic Source Routing) protocol, a route is established only when it is required and hence the need to find routes to all other nodes in the network as required.

The next popular re-active protocol is AODV (Ad-hoc On-demand Distance Vector) [2]. Generally, AODV is motivated by disadvantages of DSR but DSR has better performance of packet transmission in some sense. AODV is a routing protocol proper for mobile ad-hoc networks (MANETs) and other wireless ad-hoc networks. It establishes a route to a destination only on demand. AODV (Ad-hoc On-demand Distance Vector) avoids the counting-to-infinity problem of other distance vector protocols by using sequence numbers on route updates. AODV uses sequence numbers but not end-to-end sequence numbers. The main difference from DSR (Dynamic Source Routing) is that a source-filled path is not required.

For our proposed protocol based on small sequence numbers, the use of end-to-end sequence numbers utilizing performance metric based on the resource availability of participating nodes has been conceptualized. In this protocol, routes change dynamically with feedback responses. The problem dealt with is communication to a base station or a network of multiple base stations, with the assistance of helper nodes. These helper nodes could be implemented to relay data to the base stations or other nodes. The route can change based on window information and sender desires. There will be no route establishment and no end-to-end acknowledgement re-send needed. If a helper nodes or nodes are used, there is additional hop-by-hop acknowledgement. The return end-to-end acknowledgement can be relayed back on the same route. Whenever a node joins as a helper node, it can distribute resources dynamically.

In this thesis, we use the protocol so that it could be analyzed for its performance and compared against the other protocols. The way to implement this protocol is to apply it in a simulator that already had implementation of other wireless ad-hoc protocols so that it could be compared with already present test cases. We use NS-2 (Network Simulator) tool for generating results of the three protocols which are AODV (Ad-hoc On-demand Distance Vector), DSR (Dynamic Source Routing) and End-to-End Small Sequence Numbers routing protocol.

The performance metrics will be packet delivery fraction (reliability), data throughput and routing load. Thus we can generate those performance results of each protocol and do a comparison analysis. The next chapter will describe the end-to-end small sequence number protocol in detail. The descriptions of the protocol will be introduced in the next chapter.

# Chapter 3

# PROTOCOL DESCRIPTION

## 3.1 Protocol

This section will introduce and describe the wireless efficient protocol based on the small end-to-end sequence numbers [5]. The previous chapters provided a background about other protocols such as AODV (Ad-hoc On-demand Distance Vector) [2], DSR (Dynamic Source Routing) [3], DSDV (Destination Sequenced Distance Vector) [4] or TORA (Temporally Ordered Routing Algorithm) [19].

When compared to the above ad-hoc routing protocols, this section shows an efficient wireless routing protocol with end-to-end small sequence numbers. The sequence numbers are the most important factor in order to re-order the transmitted packets. It also prevents useless re-transmissions and ambiguity.

In this section, we will focus on protocol description and assumptions. In the next two sections, we will look at the theorem and proof of the protocol [10]. Thus we can get the background of the end-to-end small sequence numbers protocol in this chapter before looking at the experimental results.

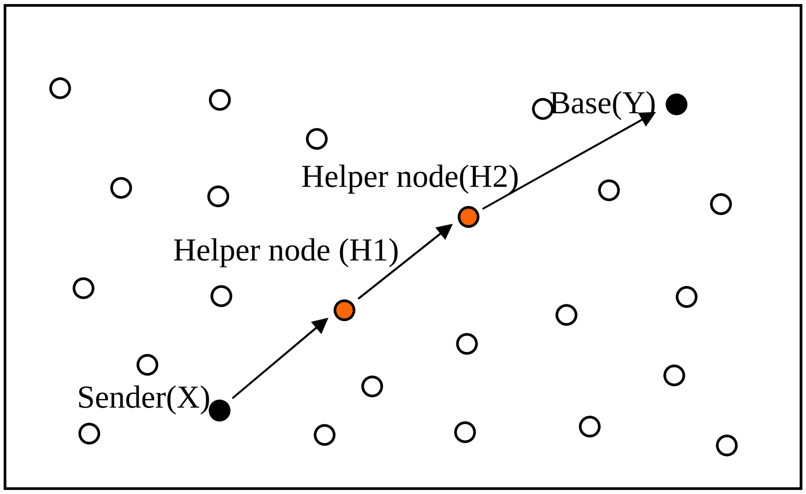

Figure 1: Network topology [5]

The protocol description is based on the previous works of Kashalkar and Metzner [5] [10]. In the above Figure 1, typical network topology shows two helper nodes, a sender station and a base station. The orange nodes are helper nodes (H1, H2) which relay and guide the packets or data to their proper destination. The black node indicates sender (X) which sends a data or packets. Another black node is a base station (Y) which receives data or packets. There might be additional helper nodes or other station nodes, but Figure 1 network topology shows a standard example of network topology. In this network protocol topology, those helper nodes have an important role for route discovery and packet transmission performance.

### 3.1.1 Assumptions

The specific assumptions of this protocol are based on the draft paper by Dr. John Metzner [10]. In the above Figure 1, Station X is willing to communicate with station Y. Station X has an address IDX and station Y has an address IDY. There are two helper nodes (H1, H2) between station X and station Y in the above Figure 1. Let H# denote a helper node, where # means a number of hops. Routes can be changed dynamically and no pre-establishments needed. Ranges are short and collisions between nodes can be minimized. Also in some cases, limited collision avoidance can be employed.

We can assume that the network is of the wideband, low utilization type. It is a common situation with low energy, short range transmissions. If the packet does not reach the base station directly, it still might reach a relay station. A relay station that hears X without error may ACK (acknowledgement), include ID, information, a window size (at least one packet), distance, number of hops and amount of power. The window size will be capacity or energy availability.

End-to-end acknowledgement is necessary for reliable data communication. End-to-end acknowledgement can be cumulative. If the link is set up directly to the base station, end-to-end is the same as hop-by-hop. If a helper node or nodes used, there is the additional option of hop-by-hop acknowledgement.

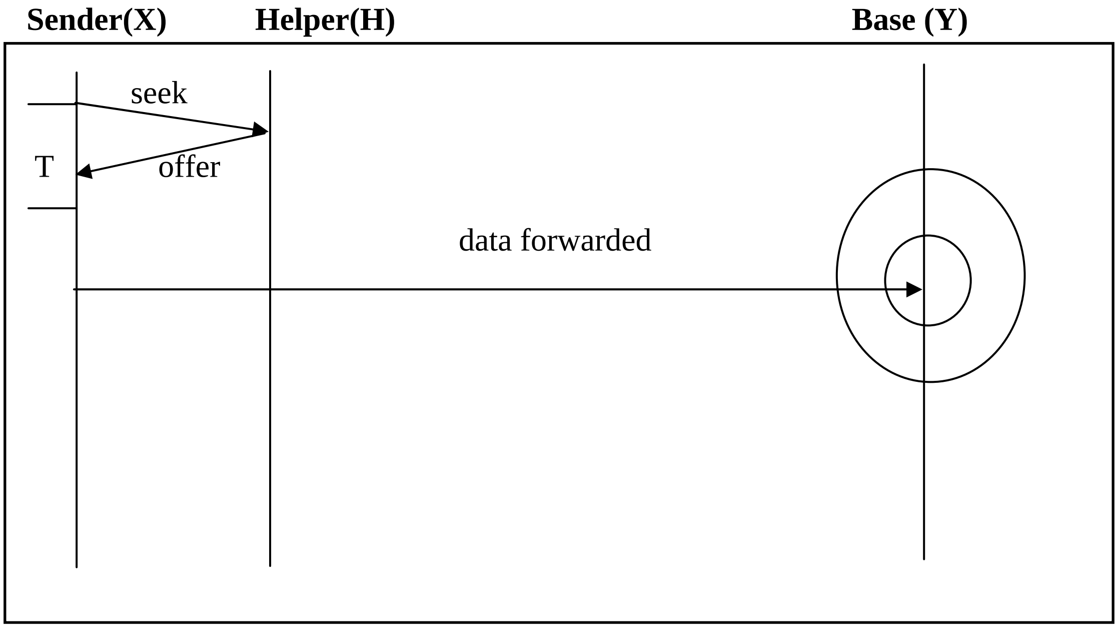

Figure 2: Message diagram [5]

In the above Figure 2, Sender X wishes to send a “seek” packet to Y with the address of X which is IDX to Base station Y (IDY). Sender X and Base station Y also can communicate directly.

In the case Sender X and Base station Y can not communicate directly, Sender X needs to send a “seek” packet to helper nodes. After Helper node H receives a “seek” packet, Helper node H sends “offer” packet to Sender X. After the T time expires, the data can be forwarded from Sender X to Base station Y. Figure 2 shows a message diagram mechanism among Sender X, Base station Y and Helper nodes H.

As we mentioned above, if the Sender X and Base station Y can not communicate directly, the “seek” and “offer” packet can be transmitted between Sender X and Helper nodes H. The “offer” packet consists of other information that may help the seeker to make a decision so as to which helper node to forward data to the destination. This information includes number of hops to the destination and available bandwidth.

On the receiving side, it chooses the most favorable helper node and sends its packets with a certain helper node’s ID. We can consider a broad-cast or multi-cast mechanism to do this. However, broad-cast or multi-cast mechanisms are somewhat limited to static network.

An alternative way can be to add the addresses and send the data. The chosen helper node knows both addresses and can subtract the same from the packet before sending. Sender X can send a new “seek” packet to Base station Y at anytime during the transmission.

Helper nodes H information in the “offer” packet will be a “window length” that indicates the number of packets that the helper nodes can control in time T. T is a system constant. The window length is a variable and can change dynamically with the helper nodes. It also varies with number of hops to the destination nodes. Apart from data itself, data packets include the sender ID, destination ID, relay ID (helper node) and a sequence number.

### 3.1.2 Description

There are three main parts categorized in the protocol description. First, there is a small number based on window size of W packets in T seconds. Second, the intermediate station can not send any packets more than T seconds after received, which means if a packet is older than T seconds then it will be discarded. Third one is the limitation on the number of hops.

The source has maximum window size of W packets and transmitted time T. One objective is to reduce number of bits in the sequence number to prevent overhead. This is also the one of the main goals that proposes an efficient protocol based on small sequence numbers.

Sometimes, the source or sender stations change routes or send the same copies to different nodes or destinations. In this case, ambiguity problems also may occur. This is also the reason why we need to design a protocol based on end-to-end (station-to-station) small sequence number protocol. In addition, to maintain our previous assumptions, we exclude the situation related to out-of-order transmissions. This protocol is implemented for small number of hops and short range which is appropriate for wireless communication networks.

Let’s assume that H hops is the maximum number of hops. Furthermore, the maximum size of packet is W and T seconds time. By the formula, we can use the maximum packet rate W/T packets per seconds. In particular, we use 8 packets per second in the simulation analysis.

All the packets can be received by the intermediate nodes at a maximum of T seconds time. If the acknowledgement is not received, the packet might be re-transmitted which also could be an ambiguity. The time-to-live feature can prevent ambiguity problems despite small sequence numbers. The variable T is a standard parameter which can be changed.

For the simulation time set up later, we defined T as 100 seconds time. The sender station can use a smaller size of window W, lower packet rate and lesser ambiguity problems because small number of packets can arrive within T seconds time. However, if we wish to send many packets per rate then we can change to a faster rate where as many as W packets actually can be delivered in T seconds.

The source or sender station knows that intermediate node h hops from source or sender station can not be holding a copy of the packet for retransmission at hT seconds after the source injected. T might be related to the number of packets that can be sent in time T at maximum rate. The source or sender station will not inject more than W different packets into a route in any T second interval. The receiver or destination station is allowed to accept any new packet in Wr positions starting from first not received. The number of bits in the sequence number is at least $\log_2[HW+Wr]$.

### 3.2 Theorem and Proof

The propagation time (packet route time) can be ignored compared to the packet time. The number of bits required in the sequence number is at least $\log_2[HW+Wr]$.

Let’s consider first H=2 (helper nodes), refer to the Figure 3. Below Figure 3 shows the receiver viewpoint, knowing the sender’s window limit is W.

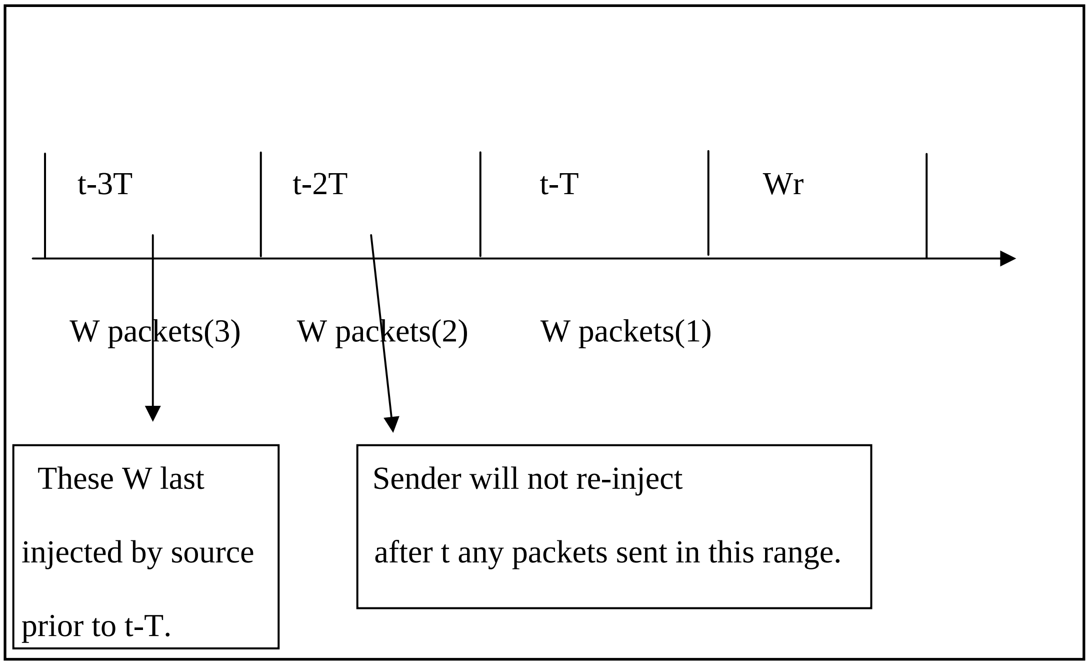
t-3T
t-2T
t-T
Wr
W packets(3)
W packets(2)
W packets(1)
These W last injected by source prior to t-T.
Sender will not re-inject after t any packets sent in this range.

Figure 3: Receiver viewpoint (H=2) [10]

Please refer to the above Figure 3. For the above H=2 example, after time t, the only previous packets that can be received at the destination station from any path are the 2W most recently sent packets. This is in addition to Wr possible new packets.

In case of H=3, the source will not re-inject after t-T any packet in interval 3, or after t-2T any packet prior to interval 3. The first intermediate node can receive an interval 3 packet as late as t-T, which could arrive at the next intermediate node as late as t, and the destination station as late as t+T, so interval 3 should be included in the possible received range. However, packets prior to interval 3 can not be injected by the source after t-2T, so they can not arrive at the destination node later than t. Thus the widest range of reception is 3W+Wr, and in general HW+Wr. This range can not exceed $2^b$, where b is the sequence number bit size. Thus it is proved.

Wr is smaller than W, so a bound for any amount of range is $2^b \geq (H+1)W$. That is, doubling H adds at most 1 bit to the required sequence number size. In this case, if a large number of hops are not efficient, it might use lower W and lower bit rate for routes. If we apply the method to communication to a base station or a network of cooperating base stations, this allows for a very efficient communication network.

In the following chapters, we will describe and introduce the experimental setup and performance analysis. In this thesis, the main works are performance & comparison analysis with simulation results of protocols. Therefore in the following section we introduce what we needed to understand the experimental setup before presenting the experimental results.

# Chapter 4

# EXPERIMENTAL SETUP

In the experimental setup, we need to set the simulation parameters and define the performance metrics with three protocol models. There will be three types of performance metrics for comparison analysis with three protocols. The three protocol types will be DSR (Dynamic Source Routing) [3], AODV (Ad-hoc On-demand Distance Vector) [2], E2ESeqNumAgent (End-to-end Sequence number agent) [5]. The three performance metrics will be packet delivery fraction (reliability), routing load and data throughput.

Table 1: Simulation parameters

| | |
|---|---|
| **Protocols** | **DSR, AODV, E2ESeqNumAgent** |
| **Number of nodes** | **25 or 50 nodes** |
| **Map size** | **500m * 500m** |
| **Traffic type** | **CBR (Constant Bit Rate)** |
| **Number of connections** | **5, 10, 15, 20** |
| **Connection rate** | **8 packets/second** |
| **Packet size** | **512 Bytes** |
| **Simulation time** | **100 seconds** |
| **Pause time** | **0, 10, 20, 40, 100 seconds** |

In the above Table 1, we set the eight parameters that can be adjusted. Basically, this simulation is based on 500m by 500m map size and CBR (Constant Bit Rate) traffic type. CBR (Constant Bit Rate) is also useful for streaming multi-media content and packets or data transmission in the wireless communication networks.

Other parameters are number of nodes and packet size. The packet size is fixed to 512 bytes which is same as 512*8 bits. The number of nodes is one of the important parameters in this experimental simulation and also can be changed. We set the simulation parameter (number of nodes) 25 nodes and 50 nodes in this experiment. Thus we can compare the performance analysis among the three protocols with the simulation results of 25 nodes and 50 nodes.

The remaining two parameters are connection rate and number of connections (CBR load). The connection rate is also fixed to 8 packets per second. On the other hand, the number of connections is 5, 10, 15, 20 and it can be changed in the simulation. Also the number of connections is called CBR load. Hence if the number of connections is large, it will be heavy traffic on the wireless communication networks.

The last remaining parameters are simulation time and pause time. The total simulation time in this experiment is 100 seconds. Also there will be a pause time which is 0, 10, 20, 40, 100 seconds. In this experiment, we try to focus on the 100 seconds simulation time.

We can simulate based on 5, 10, 15 and 20 connections. In the following results, we show mainly based on 5 and 20 connections with 3 trials to get the average results. We also assume that the start of a run can be the start of route change in the simulation experiments.

Even with a multi-connected network, not all connections are started at the same time. Or, in the 20 connection case, we assume that the other 19 or 20 connections already exist, and we check the performance of the new connection. In both cases, with heavy traffic exists there is a possibility that a new connection cannot be made, or can be made only by restarting all connections and changing many routes.

In the experimental setup, we simulated by three different protocols with three performance metrics. We set 25 nodes and 50 nodes with connection numbers from 5 to 20. In the following results sections, we showed three figures of each performance metric based on the number of nodes and connections.

# Chapter 5

# PERFORMANCE ANALYSIS

## 5.1 Performance Metrics

In this section 5.1, we will introduce and define the formula for each performance metric to simulate our experiment. There will be three performance metrics which are packet delivery fraction (PDF), routing load and data throughput.

### 5.1.1 Packet delivery fraction (reliability)

Packet delivery fraction is the portion rate of how many packets are sent and received. In other words, the packet delivery fraction is the ratio of the number of packets received to the number of packets sent. If the packet delivery fraction ratio is high, then a large number of packets are delivered to their destination with high data or packet reliability.

**Packet delivery fraction (%)**

**= (Number of packets received / Number of packets sent)*100**

Figure 4: Packet delivery fraction

The formula of packet delivery fraction ratio is defined above in Figure 4. The packet delivery fraction (PDF) is proportional to the number of packets received. Thus if a large number of the packets are delivered to the destination with respect to the number of packets sent, it will be close to 100 % PDF.

### 5.1.2 Routing load

In this section 5.1.2, we will define the term “routing load” which is related to the number of routing packets and the number of packets received. With a constant number of packets received, if the number of routing packets is large, the routing load will tend to be higher.

**Routing load = Number of routing packets / Number of packets received**

Figure 5: Routing load (overhead)

As seen in Figure 5 above, we define the formula of routing load number by the calculation of the number of routing packets in proportion to the number of packets received. A routing packet is a packet that traverses in the route of networks. Therefore, if a large number of routing packet exists, the routing load will be increased which is not efficient.

In the previous performance metrics, the packet delivery fraction (PDF) is proportional to the number of packets received while the routing load is inversely proportional to the number of packets received.

### 5.1.3 Data throughput

In this section 5.1.3, we will define the formula of data throughput and estimate it by the formula. The data throughput is one of critical performance metrics in this simulation experiments.

**Data throughput**

**= (Packet size (bytes) * 8 * Number of packets received) / Simulation time**

Figure 6: Data throughput

In Figure 6 above, the formula for the estimation of data throughput is defined. Basically, the third performance metric (data throughput) is related to three parameters as shown in Figure 6 which are simulation time, packet size and the number of packets received. In this performance metric, the data throughput is proportional to the number of packets received.

The data throughput is the data that is received per second and is notated as bits per second. In our simulation experiment, the packet size is 512 bytes so we multiplied 8 by the packet size to convert into bits. Lastly, the simulation time will be 100 seconds.

### 5.2 Simulation Results

As you see in the previous section (performance metrics), the number of packets received is the most important factor in the simulation experiments. Also that is why we compare the three different routing protocols (AODV, DSR, E2ESeqNumAgent) by those three different performance metrics.

We will show the five figures of the each performance metric based on 25 nodes and 50 nodes. In each figure, there will be three bars (representing AODV, DSR, E2ESeqNumAgent) of each performance metric based on the two different number of nodes.

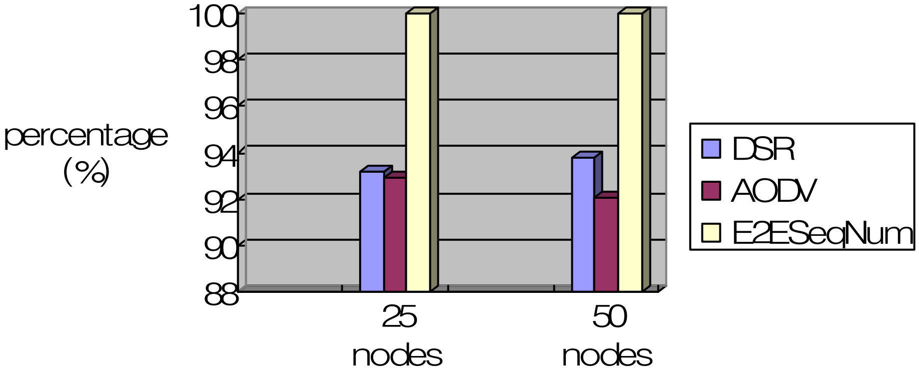

Figure 7: Packet delivery fraction result

In the above Figure 7, you can see the comparison experimental results of three different protocols. The x-axis indicates the number of nodes and the y-axis shows the percentage (%) of the packet delivery fraction ratio. The above results are based on the 512 bytes packet size and with a connection number of 20 (CBR load).

In Figure 7, which is calculated for 25 nodes and 50 nodes, the End-to-end sequence number agent protocol indicates the highest percentage of the packet delivery fractions (reliability). The DSR (Dynamic Source Routing) is the second highest percentage of the packet delivery fraction. The lowest percentage of the packet delivery fraction ratio is AODV (Ad-hoc On-demand Distance Vector). Although the AODV (Ad-hoc On-demand Distance Vector) routing protocol is developed to avoid the disadvantages of DSR (Dynamic Source Routing), the DSR (Dynamic Source Routing) has better performance of packet delivery fraction than AODV (Ad-hoc On-demand Distance Vector).

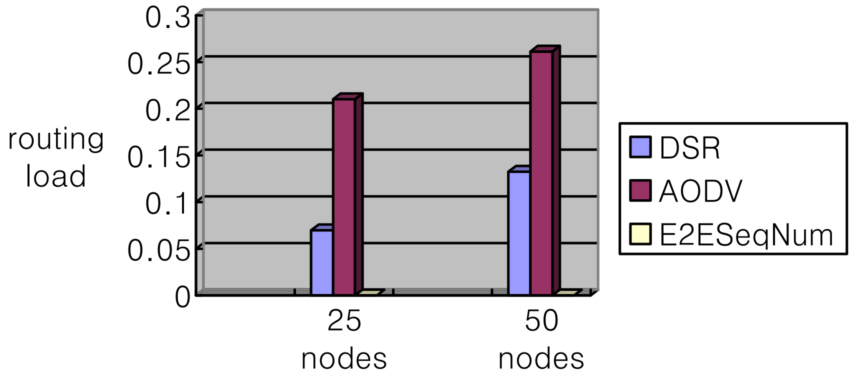

Figure 8: Routing load (overhead) result

Figure 8 above shows the routing load results. Again, the End-to-end sequence number agent routing protocol has the best performance on the routing load which is close to zero. The DSR (Dynamic Source Routing) has the second highest performance with the routing load performance metric lower than the AODV (Ad-hoc On-demand Distance Vector). This performance metric is based on 20 connections.

Most notably, Figure 8 shows that the highest routing load value is for AODV (Ad-hoc On-demand Distance Vector) based on the 50 nodes is above 0.25. The 50 nodes metric has heavier traffic load than the 25 nodes metric. In Figure 8, DSR (Dynamic Source Routing) shows low routing load (0.05 ~ 0.1) based on 25 nodes.

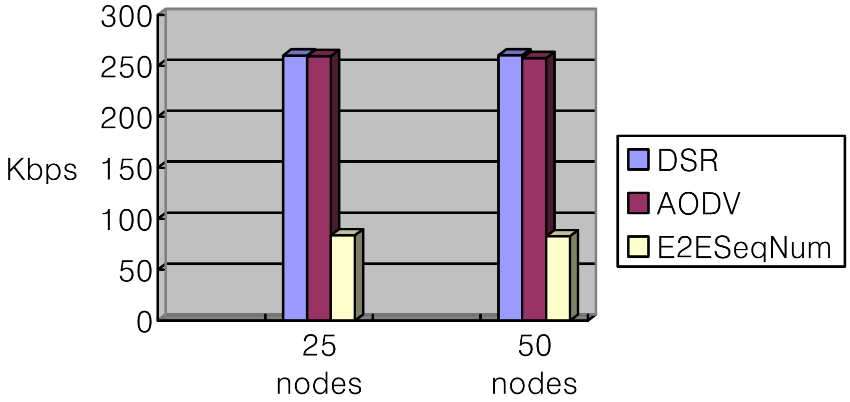

Figure 9: Data throughput result (20 connections)

In Figure 9 above, the data throughput results are presented based on a connection number of 20. A connection number of 20 has a heavier traffic load than a connection number of 5 or 10. The data throughput metric will be one of the disadvantages of the routing protocol (End-to-end sequence number agent) since the end-to-end sequence number agent routing protocol has lower performance on the metric data throughput. In the below Figure 10, we can look at the data throughput results with 10 connections. The data throughput of DSR and AODV is lower than the previous Figure 9.

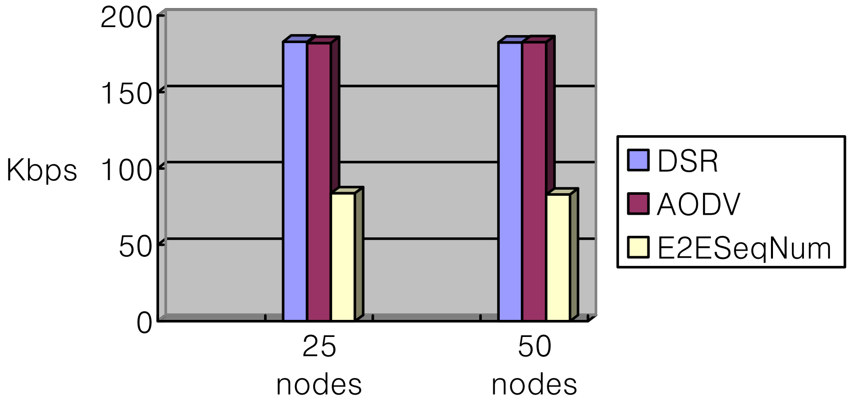

Figure 10: Data throughput result (10 connections)

The end-to-end sequence number agent routing protocol has the best performance for most of the metrics (packet delivery fraction, routing load, data throughput) because this protocol has delivered packets to the destination very well and there is not too much difference between the number of packets sent and the number of packets received.

However, the total number of packet received is smaller than other routing protocols such as AODV and DSR based on 20 connections.

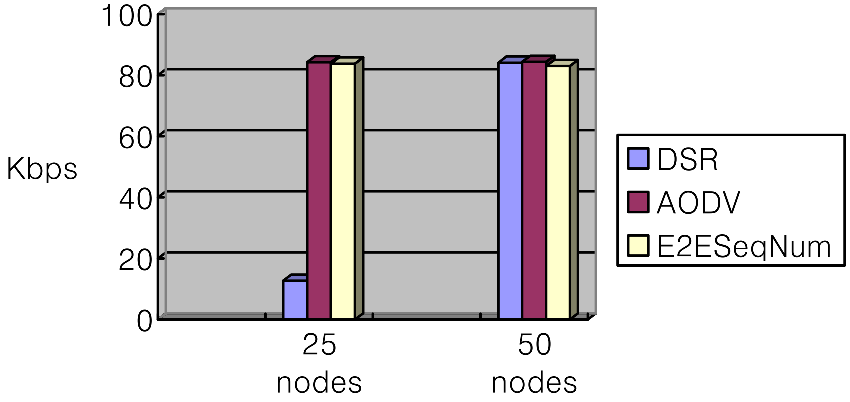

Figure 11: Data throughput result (5 connections)

However, Figure 11 shows a better performance of End-to-end sequence number agent protocol on the performance metric data throughput. Figure 11 is based on 5 connections which is a lighter traffic load. Figure 11 indicates the data throughput of the End-to-end sequence number agent routing protocol is the highest, and is almost the same as AODV. Thus the End-to-end sequence number agent routing protocol can send packets to the destination very well with helper nodes but the speed is not as fast as AODV and DSR.

We can find a big trend via the above experimental simulation results. Even though the data throughput of the end-to-end small sequence number protocol is somewhat low, it shows the highest performance on the packet delivery to the destination and the lowest routing load (overhead).

If we reduce the number of connections as a parameter, the packet delivery fractions will be almost the same among those three protocols. However, with the high traffic calculation (20 connections), the end-to-end small sequence number protocol shows an efficient and safe packet delivery performance which is the main advantage of a wireless routing protocol utilizing the end-to-end small sequence numbers protocol.

In addition, the end-to-end small sequence number protocol shows the lowest routing overhead with high traffic (20 connections) which can be also an energy-efficient and the most appropriate routing protocol for wireless mobile ad-hoc networks. In the following section, we will describe the comparison analysis among three protocols based on our experimental simulation results.

### 5.3 Comparison Analysis

In the previous section, we looked at the simulation results and experimental setup for comparison performance analysis of three different routing protocols (AODV, DSR, End-to-end sequence number agent). There is also another protocol DSDV (Destination Sequenced Distance Vector) [4] which uses TCP (Transmission Control Protocol) [6]. Even though the DSDV (Destination Sequenced Distance Vector) [4] was not used as a protocol for comparison analysis, this protocol had the second highest performance on the packet delivery fraction ratio. Also this protocol has the third best performance on the data throughput and routing load. Since DSDV (Destination Sequenced Distance Vector) uses TCP (Transmission Control Protocol), it is very appropriate for data or packet transmission.

There is also another possible routing protocol named TORA (Temporally Ordered Routing Algorithm) [19]. This protocol is a situation-aware routing protocol and we predict it would perform well on our performance metric even if it is not selected for our simulation experiments. Thus in our performance comparison analysis, we basically compared the performance of the reactive (on-demand) protocols. There are also lesser known reactive routing protocols such as ACOR (Admission Control enabled On demand Routing), Backup Source Routing or CHAMP (CacHing And MultiPath routing) protocol.

From Figure 7 to Figure 11, we look at the experimental simulation results of the three protocols based on the three performance metrics. The End-to-end sequence number agent protocol has the best performance on the performance metric except for data throughput. The End-to-end sequence number agent routing protocol is the most appropriate one for packet delivery to the destination but not at a fast speed. This is because the End-to-end sequence number agent routing protocol has the helper nodes which assist the packet sent to the proper destination. Obviously, most packets sent are delivered to the proper destination but it is somewhat slow with heavy traffic. If we can suggest another fast routing protocol or accelerate it then it should be another great efficient routing protocol in wireless communication networks.

We also can consider other possible performance metrics such as security, energy efficiency or complexity. The security area has a little bit different perspective from the networking area in some sense but security is still a hot issue. For complexity, we can consider an algorithm perspective such as time or space domain. In the case of our performance analysis, we have focused on fast speed packet transmission rather security or complexity.

However, energy efficiency is also a very important performance metric on wireless packets or data transmission networks. Energy efficiency is also related to the number of nodes. We simulated using both 25 nodes and 50 nodes in this experiment.

Based on the number of nodes, we can see the different performance results between 25 nodes and 50 nodes. Clearly, energy consumption is lower on 25 nodes than 50 nodes. In some sense, the value of each performance metric is higher on 50 nodes than 25 nodes. The 50 nodes performance metric needs more energy consumption. To save energy consumption, if we are able to show the performance from 50 nodes using only 25 nodes, it also will be another great mechanism.

The End-to-end sequence number agent routing protocol definitely has better performance than the AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing) protocols. If we add the fast speed and energy efficient algorithm from the End-to-end sequence number agent routing protocol, it will be a great routing protocol for wireless packet or data transmission mechanism in the networks.

In this section, we described our comparison and performance analysis of the three different protocols which are DSR (Dynamic Source Routing), AODV (Ad-hoc On-demand Distance Vector) and End-to-end Sequence Number Agent routing. In addition, the three different performance metrics are the packet delivery fraction ratio (reliability), routing load (routing overhead) and the data throughput (bits per second).

Finally, we can indicate the advantages and disadvantages of each protocol’s performance based on our previous simulation experimental results. In the next chapter, we will conclude and discuss future works.

# Chapter 6

## CONCLUSIONS & DISCUSSIONS

Looking back at the previous chapters, we introduced and described the background of the protocols. There are pro-active protocols and re-active protocols in the ad-hoc routing protocol. The pro-active protocols refresh and maintain the route periodically but re-active protocols are based on the on-demand mechanism with packet requests. In this perspective, the re-active ad-hoc routing protocol is more energy-efficient and appropriate for wireless data communication networks.

In the list of re-active routing protocols, we chose the two popular routing protocols which are DSR (Dynamic Source Routing) [3] and AODV (Ad-hoc On-demand Distance Vector) [2]. These ad-hoc routing protocols are the most commonly used reactive protocols in wireless data communication networks.

In comparison to the above two re-active protocols, we proposed an efficient wireless routing protocol based on small end-to-end sequence numbers [5]. As shown in the previous experimental simulation results, the end-to-end small sequence number protocol performs better than AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing) ad-hoc routing protocol based on the performance metrics.

In addition to DSR (Dynamic Source Routing) [3] and the AODV (Ad-hoc On-demand Distance Vector) [2] routing protocol, we briefly introduced the DSDV (Destination Sequenced Distance Vector) [4] and TORA (Temporally Ordered Routing Algorithm) [19] routing protocol.

The DSDV (Destination Sequenced Distance Vector) uses TCP (Transmission Control Protocol) [6] and it is also proper for wireless data transmission networks. However, the DSDV (Destination Sequenced Distance Vector) is a pro-active routing protocol with slow reactions which means this protocol is not an efficient one.

The TORA (Temporally Ordered Routing Algorithm) routing protocol might be more efficient than DSDV (Destination Sequenced Distance Vector) or other ad-hoc routing protocols. The TORA (Temporally Ordered Routing Algorithm) ad-hoc routing protocol is an adaptive (situation aware) routing protocol which combines the advantages of pro-active and re-active routing protocols. However, the disadvantage of TORA (Temporally Ordered Routing Algorithm) routing protocol is that it is only efficient on the activated nodes. Thus we simulated and did performance comparison of the three protocols (AODV, DSR, E2ESeqNumAgent) in our experiments based on the three different performance metrics which are packet delivery fraction ratio, routing load and data throughput.

The DSR (Dynamic Source Routing) ad-hoc routing protocol [3] is a commonly used re-active routing protocol. DSR (Dynamic Source Routing) is an on-demand routing protocol implemented to the bandwidth consumed by control packets in the wireless communication networks without using a table-driven approach. Also, the DSR (Dynamic Source Routing) protocol uses flooding if there is no source route. The DSR (Dynamic Source Routing) protocol is based on the source route, otherwise it uses flooding which send packet to all nodes. However, the DSR (Dynamic Source Routing) protocol shows a better performance with high speed packets transmission than AODV (Ad-hoc On-demand Distance Vector) protocol.

The AODV (Ad-hoc On-demand Distance Vector) [2] is also commonly used in ad-hoc routing protocol and was designed to overcome the disadvantages of the DSR (Dynamic Source Routing) protocol. Even the DSR (Dynamic Source Routing) protocol shows a better performance than the AODV (Ad-hoc On-demand Distance Vector) protocol in our experiment, the AODV (Ad-hoc On-demand Distance Vector) routing protocol is still a good protocol for wireless mobile ad-hoc networks with the advantage of no extra traffic required. Both AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing) use sequence numbers so we can compare the performance results with the end-to-end small sequence number protocol. One of the basic approaches of the AODV (Ad-hoc On-demand Distance Vector) routing protocol is its use of an intermediate node for path discovery.

The proposed efficient wireless routing protocol is based on the small sequence numbers [5]. When we segmented packets or data, we label the sequence number on each packet to maintain a proper order. Also the sequence number prevents packet transmission ambiguity. Since each packet has their unique number, it prevents multiple sending or inappropriate re-transmissions. Thus in this end-to-end small sequence numbers protocol, our focus is to propose an efficient protocol based on the small sequence numbers. If there is a large number of sequence numbers, there will be high traffic overhead and a more complicated order.

The basic concept of the end-to-end small sequence number protocol is based on the previous work by Dr. John Metzner [10]. In the previous chapter of this thesis, this protocol was described with assumptions and theory proof in detail. In this protocol, routes change dynamically with feedback responses with the assistance of the helper nodes to the destination. These helper nodes relay the data to the base stations or other possible nodes. There is no extra need for route set up and packet re-transmissions. In addition, the end-to-end transmission and its acknowledgement are required for reliable data communication. End-to-end connection is similar to hop-by-hop basis. Also this end-to-end small sequence numbers protocol assumes that the network is wideband and low utilization with a low energy, short range transmission.

In conclusion, the end-to-end small sequence number protocol performed better than AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing) protocols based on the performance metrics (packet delivery fraction, data throughput, routing load). Also, the end-to-end small sequence number protocol has a somewhat lower performance on the metric data throughput than other protocols. If we consider the packet transmission speed and energy-efficiency of end-to-end small sequence number protocol, this might be another good opportunity for future work.

For other possible future work or extension, we can consider security and energy-efficiency. Basically, this protocol and experimental setup is designed by NS-2 (Network Simulator second version) [7] with Tcl scripts file. Also Tcl scripts file can generate other possible scenario or traffic scripts. With this view, we can consider generating an energy-efficient routing scenario through the Tcl scripts. For another possible future work, we can think about segmentation. If we segment 512 bytes into 26 bytes 20 times, then they can be transmitted quickly with low energy consumption.

Another area of future study is to consider of the effect of packet errors and retransmission. The small sequence number method is designed to reduce retransmissions greatly for a noisy channel, compared to using TCP (Transmission Control Protocol) [6], for example. As we described TCP in the prior chapters, TCP is one of the standard Internet protocols using large sequence numbers. With TCP, there is a great deal of overhead in a rather large packet. Communication that breaks the packet into small pieces with small sequence numbers, only pieces with errors may need to be retransmitted, while with depending only on TCP a whole packet would have to be retransmitted even if only one piece was in error.

Furthermore, security might be another future focus for the protocol based on the end-to-end small sequence numbers. This protocol has helper nodes so that the packet can be sent to destination safely and properly. If the security implementation is added, it can prevent some possible security attacks such as DOS attack (Denial Of Service).

So in conclusion, the study and implementation of the NS-2 (Network Simulator second version) with Tcl scripts files are the most important and difficult work in this thesis. In order to design and simulate an efficient routing protocol, we have studied the many kinds of previous routing protocols to give us background.

## REFERENCES

[1] "List of ad-hoc Routing Protocols", Wikipedia http://en.wikipedia.org/wiki/Ad_hoc_routing_protocol_list.

[2] C. Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing", July 2003.

[3] D. Johnson, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4", February 2007.

[4] "Destination Sequenced Distance Vector (DSDV)", Wikipedia http://en.wikipedia.org/wiki/Destination-Sequenced_Distance_Vector_routing

[5] Kiran A. Kashalkar, "Implementation of an Efficient Wireless Routing Protocol Based on Small End-to-End Sequence Numbers", Pennsylvania State University at University Park (Master's Thesis), August 2009.

[6] Defense Advanced Research Projects Agency, RFC 793, "Transmission Control Protocol", September 1981.

[7] The Network Simulator (NS-2), http://www.isi.edu/nsnam/ns/.

[8] Trace Graph - Network Simulator NS-2 trace files analyzer, http://www.angelfire.com/al4/esorkor/

[9] Nam: Network Animator, http://www.isi.edu/nsnam/nam/.

[10] J.Metzner, “New acknowledgement protocols for dynamic wireless routing and multiple path transmission”, Draft 2008.

[11] OMNet++ Discrete Event Simulation System, http://www.omnetpp.org/.

[12] H. Zimmermann, “OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection”, IEEE Transactions and Communications, April 1980.

[13] The ns Manual, The Network Simulator ns-2: Documentation, http://www.isi.edu/nsnam/ns/ns-documentation.html.

[14] NS-3 (Network Simulator 3), http://www.nsnam.org.

[15] NS-3 Tutorials, http://www.nsnam.org/docs/tutorial.html.

[16] Ad-hoc On-demand Distance Vector (AODV), Wikipedia http://en.wikipedia.org/wiki/Ad_hoc_On-Demand_Distance_Vector_Routing.

[17] Y. Hu, “Implicit Source Routes for On-Demand Ad Hoc Network Routing”, Rice University, October 2001.

[18] B.A.T.M.A.N. Concept, “B.A.T.M.A.N. Overview” http://www.openmesh.net/wiki/BATMANConcept.

[19] TORA (Temporally Ordered Routing Algorithm) – Wikipedia

http://en.wikipedia.org/wiki/Temporally-ordered_routing_algorithm

[20] Dynamic Source Routing (DSR), Wikipedia
http://en.wikipedia.org/wiki/Dynamic_Source_Routing

[21] Francisco J. "Implementing a New Manet Unicast Routing Protocol in NS2", December 2004.

[22] Network Simulator tutorial (2002)
http://www.isi.edu/nsnam/ns/ns-tutorial/tutorial-02/index.html

[23] AODV (Ad-hoc On-demand Distance Vector) example
http://www.soi.wide.ad.jp/class/20060035/slides/04/51.html

[24] DSR (Dynamic Source Routing) example
http://www.cse.wustl.edu/~jain/cse574-06/ftp/j_fadh/sld016.htm

[25] Naga.V.Yedida, Rajesh Reddy Challa, "Performance comparison of AODV, DSR and OLSR routing protocols in static scenarios", March 2010.

[26] Nor Surayati Mohamad Usop, Azizol Abdullah, Ahmad Faisal Amri Abidin, " Performance evaluation of AODV, DSDV & DSR routing protocol in grid environment", IJCSNS 2009

[27] Yinfei Pan, “Design routing protocol performance comparison in NS2: AODV comparing to DSR as example”, SUNY-Binghamton