ENSC 835: HIGH-PERFORMANCE NETWORKS
CMPT 885: SPECIAL TOPICS: HIGH-PERFORMANCE NETWORKS

Comparative Analysis of Wireless Routing Algorithms in ns-2

Spring 2006

Final Project Report

www.sfu.ca/~ekchen

Edward Chen (ekchen@sfu.ca)

Colin Ng (cnge@sfu.ca)

List of Acronyms

AP	Access Point
AODV	Ad-Hoc On-Demand Distance Vector
CSS	Center for Systems Science
CMU	Carnegie Mellon University
DSDV	Destination-Sequenced Distance Vector
DS	Distributed System
DSR	Dynamic Source Routing
GUI	Graphic User Interface
LAN	Local Area Network
RP	Routing Protocol
SFU	Simon Fraser University
UML	Universal Modeling Language
WAN	Wide Area Network
WDS	Wireless Distributed System

Table of Contents

List of Figures

List of Tables

Abstract

A Wireless Distributed System (WDS) appears to be a promising approach to resolving critical networking limitations prevalent in a centralized wireline topology . Information is routed through intermediary nodes to reach its destination. In a WDS there are many possible routes that a packet may traverse. An effective way to reduce congestion is to efficiently route packets through the network maze. However the choices grow exponentially due to the combinatorial nature of this topology. Routing protocols must be carefully designed to maximum network efficiency.

In this project, we analyzed and compared three routing protocols (AODV, DSDV and DSR) implemented in ns-2. The protocols are simulated in a WDS environment with routing protocols, number of nodes and mobility as varying parameters.

We found that DSR required the most nodal overhead due to the storage of all possible routes to all destinations in its cache. DSDV introduced most traffic into the network due to its periodic update of the entire network from every node.

When the network is small, we found that DSDV is most efficient in delivering data packets to destinations. When the network is large, the network should utilize AODV or DSR to avoid excessive forwarding of data packets.

This is also the case with total number of dropped packets. Data packets in a network utilizing DSDV are being dropped more frequently than AODV and DSDV, regardless of size of the network. This difference grows exponentially as the network grew larger.

However, on average, data packets routed using DSDV experienced least End-to-End delay, regardless of the size of the network. Therefore, we concluded that AODV and DSR are more reliable routing protocols while DSDV minimized End-to-End delay and maximized throughput.

1 Introduction

1.1 Overview and Shortcoming of Centralized System

With the explosive growth of networks, there exists a critical need to deliver information in a robust and efficient manner. Although applications such as the Internet were built on the vision of a completely decentralized network that allowed unlimited scalability [3], the reality is that most systems today are still built on the client-server concept.

In a centralized system, all functions and information are contained within a server with clients connecting directly to the server to send and receive information, as illustrated in Figure 1.

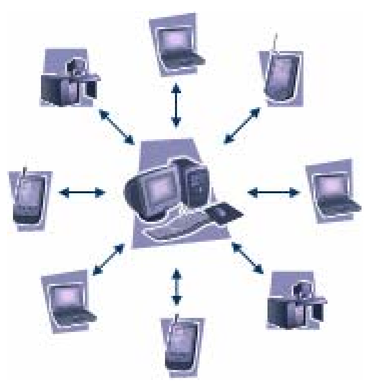


Figure 1 Traditional Client-Server Topology [6]

As the network continues to grow, this traditional topology is inadequate to meet the demand of its users. This heavy emphasis on a central servers places great burden on the network. As a centralized network expands, issues such as scalability, fault-tolerance, security and infrastructure cost will undoubtedly hinder its growth.

In recent years, the establishment of the IEEE 802.11 wireless protocol has allowed users to roam freely within a wireless Local Area Network (LAN) by communicating with access points in the LAN. However, this protocol still utilizes a centralized topology for communication.

In a typical wireless LAN environment, illustrated in Figure 2, clients utilize access points (AP) in networks to connect with other clients. Information is first sent from a sender to the AP and then forwarded to the receiver. This approach still retains deficiencies of the traditional Centralized systems, e.g., the failure of access points will have a catastrophic effect on the overall network.

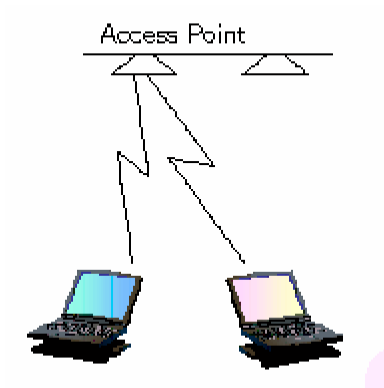


Figure 2 Wireless Local Area Network (LAN) [6]

Ideally, we would like to combine the IEEE 802.11 standard with the functionality of a distributed system environment. Issues such as scalability, connectivity and security in the wireline centralized approach can thus be much more improved than a centralized topology.

1.2 A Better Network: Distributed Systems

A Distributed System (DS) can be defined as a network topology that decentralizes the system so that no node has a greater central role than any other node[6]. This topology fulfills the need for a robust, open-ended and infinitely scalable system by eliminating the central server and efficiently utilizes network resources [4]. Network resources are allocated across the network to alleviate computational bottlenecks within a single node or network area.

A partial application of a Distributed System is the Internet. Initially the Internet was designed to be a robust system with unrestricted scalability [5]. In reality however, it is still dominantly reliant on localized web servers for database and file storage. If a major server or router fails, many clients will be unable to send or receive data. Issues related with a centralized topology are still prevalent with the current Internet.

In a fully distributed system, all nodes on the network are of equal significance, the failure of one node should not have a catastrophic effect on any other node on the network. A fully Distributed System has the potential to enhance system security, scalability and connectivity [3].

1.2.1 Distributed System Security

Unlike a centralized system, a distributed system lacks a central server for storage of critical information. The information is spread among nodes and is retrieved only at the demand of the requesting node. When the security of any node is compromised, the breach is localized and thus has no detrimental effect [6].

1.2.2 Distributed System Scalability

A distributed system does not utilize a central server to process incoming data from all clients. Individual nodes in a fully distributed system communicate directly among themselves.

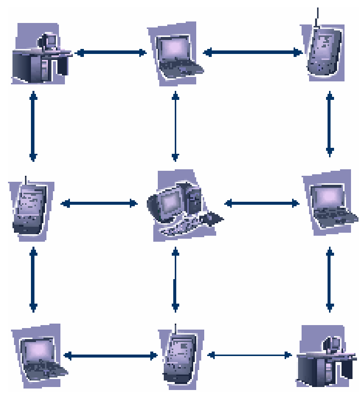


Figure 3 Distributed System Topology [6]

Requests for information and the actual transfer of information are performed locally. This eliminates the need for a powerful server and thus provides enhanced scalability. Additional nodes are able to freely join the network without incurring additional strain on the system [6]. Each additional node is also an additional resource for the network to utilize to ensure that the overall network remains efficient and robust.

1.2.3 Distributed System Connectivity

Unlike a centralized system, a distributed system does not have a single point of failure. When a node fails in a distributed environment, information is simply routed around the failed node and continues its path to the receiver node. The distributed system will maintain its functionalities as long as there is an alternate path available.

1.3 Project Overview

There are two objectives for this project – to provide comparative analysis between the three different routing protocols and to provide individual analysis of each routing protocols

To illustrate the path taken to realise our objectives, this report is organized as follows:

Section 2 provides background information of the three routing protocols used in this project.

Section 3 presents an overview of the different applications used to generate and analyse the trace files outputted by ns-2.

Section 4 provides an overview of simulation setup and parameters used in this project.

Section 5 discusses the results obtained and provides explanations of the results.

Section 6 concludes this project and discusses possible future works to be done.

Additional Matlab plots, code listings for both ns-2 and Java are provided in the *Appendix*.

2 Routing Protocols (RP) Overview

An individual packets needs to be routed to its destination node through one of many available paths in the network. Nodes themselves can behave as routers and forward packets onto the next node. When a new node joins the network, additional possible routes are created and thus add to the complexity of route discovery and selection.

The routing process is further complicated by the mobility of wireless nodes. In a wireless distributed system, nodes communicate wirelessly and thus have the ability to roam freely as long as it is within the signal proximity of at least one other node in the network.

These issues create tremendous challenges to RP designers. A RP must be designed so that it could route packets to their destinations efficiently according to the different dynamical network characteristics (e.g. congestion, node failure) and user specifications (e.g. packet priority, minimum delay).

In this project, we will compare three different routing protocols implemented in ns-2 in a wireless distributed environment. The first two routing protocols discussed, AODV and DSR, are known as reactive routing protocols. This means that when a path does not exist from a source node to a destination node, a path is queried. The last routing protocol discussed is a proactive routing protocol known as DSDV. A proactive protocol is a protocol that attempts to discover a path before a message is sent.

2.1 Ad-Hoc On-Demand Distance Vector Routing (AODV)

The Ad-Hoc On-Demand Distance Vector Routing is a routing protocol especially designed for ad-hoc mobile networks [7]. The protocol uses a routing table to determine the path that a packet will travel to arrive at its destination. However, a route is only kept active for a finite period of time. A route is deemed active if there are packets travelling along it within a set periodic time frame. If this time expires with no packets traversing, the path is removed from the routing table. In AODV, the routing table contains the next hop of the path, and this next hop is the destination that the packet will travel to.

If a source-destination path does not exist in the routing table, a network configuration packet will be sent to request for routes to the destination node. Figure 5, based on Universal Modelling Language (UML), summarises the logic flow involved with this algorithm, while Figure 6 shows how a node handles a network configuration packet.

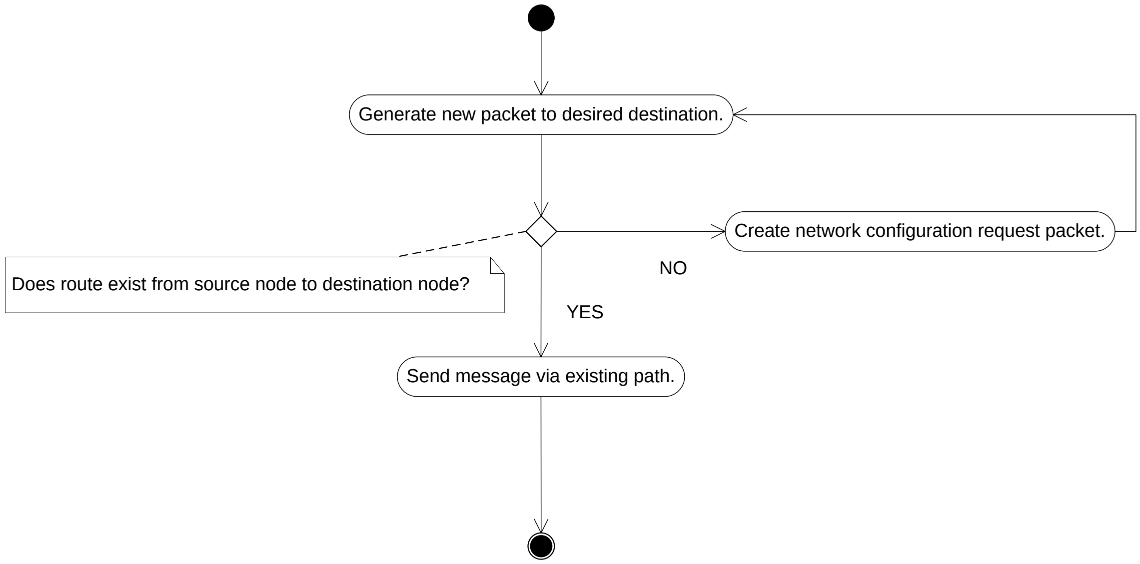


Figure 4 Activity Diagram of AODV Routing Protocol

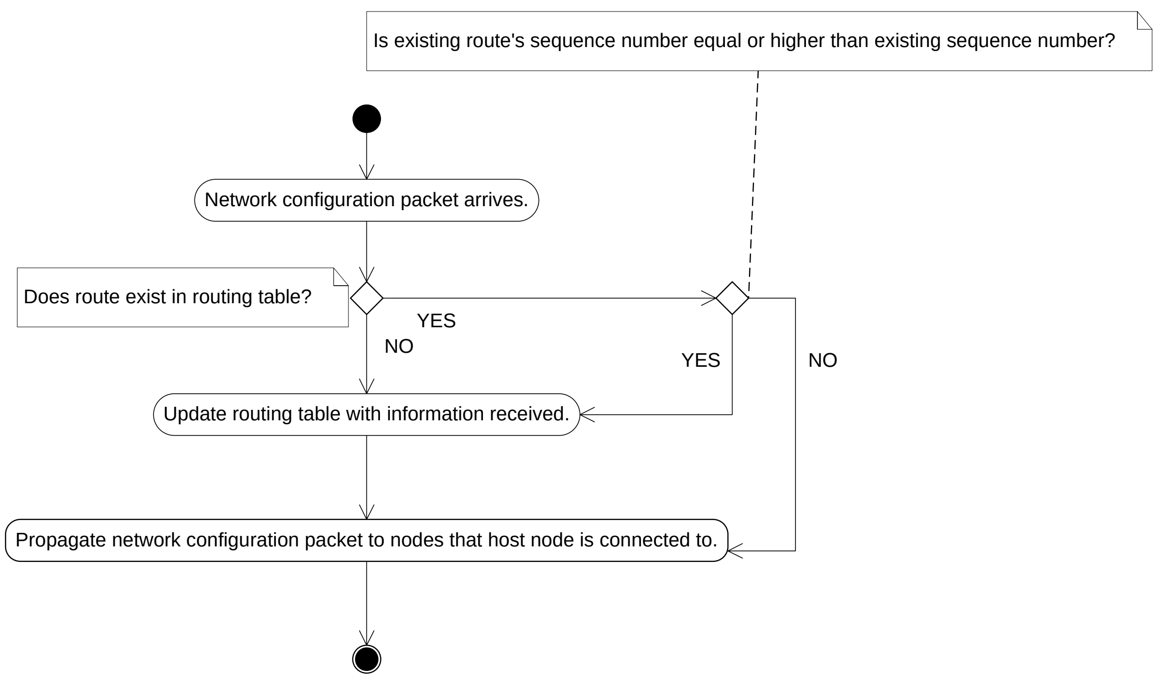


Figure 5 Activity Diagram of AODV network configuration packet

It is also important to note that AODV is able utilising both multicast and unicast. When propagating a network configuration packet, the packet can be multicasted to the other members of the multicast group. If this is the case, there are certain considerations, such as the overhead involved with sending such a heavy load, that have to be taken into account. This will be analysed later during comparisons between routing protocols.

In addition, AODV also has a mechanism that constantly sends out messages to nodes that are within one hop of the source node. This mechanism is used to keep links between the source node and these nodes alive.

2.2 Dynamic Source Routing

Perhaps the simplest of the three routing protocols, Dynamic Source Routing (DSR) incorporates two different mechanisms to keep track of available routes throughout the network. The two mechanisms are route-discovery and route-maintenance [8].

Furthermore, as the indicated by its name, this protocol is only initiated when there is a request to send a packet from a source node to a destination node. Hence, there is no additional overhead required to maintain a "current-view" of the network topology. This "current-view" is in fact, the actual path from the source node to the destination node. As a result, the routing table within DSR contains multiple paths to a destination nodes, as opposed to just keeping the next hop within a routing table, such as the case in AODV.

Figure 7 is a UML activity diagram that summarises the flow of the protocol.

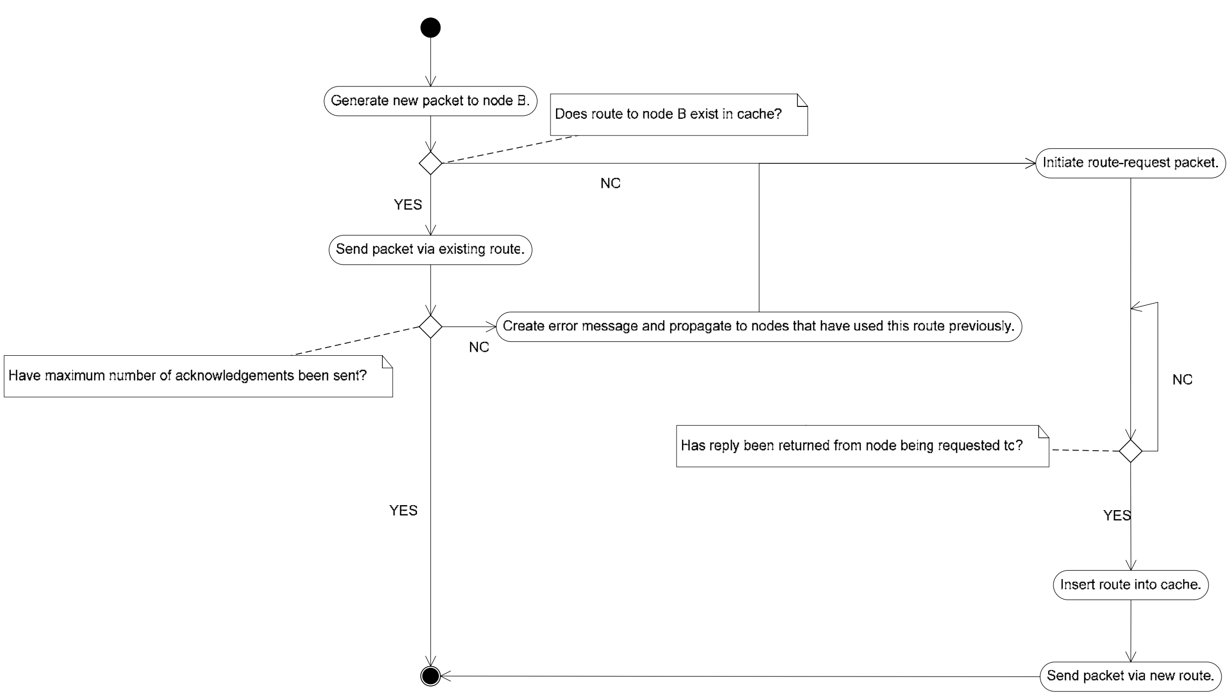


Figure 6 Activity Diagram of Logic Flow in DSR

When a packet is sent from node A to node B, a check is first made to see if a route exists. If the route already does not exist, the route-discovery mechanism is initiated and a network configuration packet is sent to discover the path to the destination. If the inquiry is replied with an acknowledgement, the route is placed into the source's cache for future references. The packet is then sent via this newly-discovered path.

If a path already exists in the cache, the route-maintenance mechanism is initiated and the packet is sent. The source node waits for an acknowledgement from the destination node. This acknowledgement serves as an indication that the route still exists. A important parameter of this protocol is to send a specific number of acknowledgments. Hence, a successfully-sent packet would receive equal number of acknowledgements from the destination node. If this does not happen, then the route is treated as a broken route and the route-discovery mechanism is initiated for every node on the broken path.

2.3 Destination-Sequenced Distance Vector Routing (DSDV)

Destination-Sequenced Distance Vector Routing is an extension of the Bellman-Ford algorithm, which involves finding the shortest path between two points [9]. There is a high overhead because new network-configuration packets are constantly being broadcasted and existing network routes are constantly updated. This overhead is essential because the network-configuration packets are used to keep a "current view" of the network topology. This "current view" however, is only stored in the form of the next hop the packet should travel along a set path in the routing table.

When a network-configuration packet is received by a node, there are two actions that are taken. First, the route information in the network-configuration packet is compared node's cache. If a shorter path exists, the shortest path will be used to replace the existing path. If the path does not exist in the cache, then this path is copied into the routing table. This task, although it is simple, is vital to the functionality of the protocol because it ensures that every node has a copy of the shortest path from a source node to a destination node.

The second action performed is to forward network-configuration packet to all the other connected nodes. This is done to ensure that other nodes have access to the most recent network routes. Figure xxx below exemplifies how the packets are forwarded to different nodes, hence guaranteeing that all nodes have the shortest route from any given source node to any given destination node.

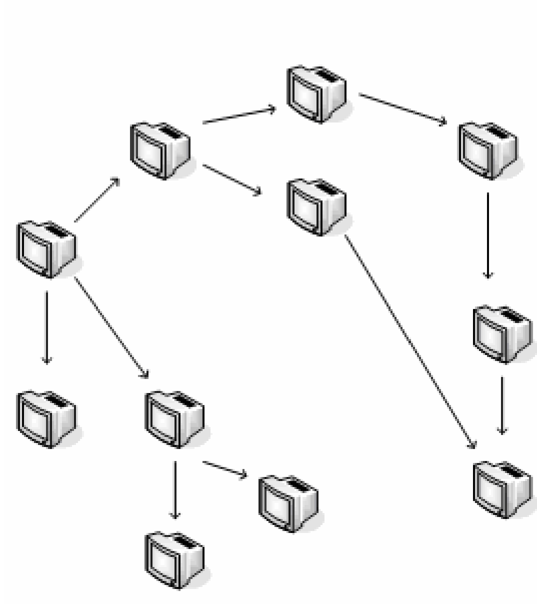


Figure 7 Network Configuration Packet Propagating through Network

It can be seen that the time interval between network configuration packets are sent is vital to keeping the efficiency of the routing protocol. If a network is highly-dynamic, network-configuration packets must be sent frequently. If a network is static, there is no need to update the routing table often and thus the time interval between network-configuration packets can be longer

3 Project Application Overview

Before any analysis can be performed, there are several steps that have to be taken to accurately generate the desired results. This section provides an overview of the various applications used.

Figure 8 illustrates these steps in the form of a UML state diagram.

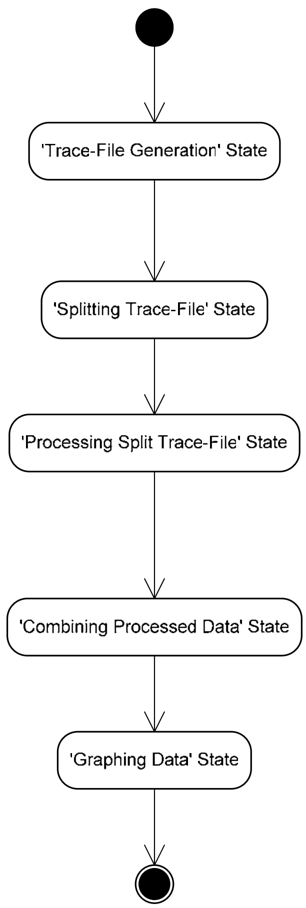


Figure 8 Steps undertaken before analysis

The purpose of each state is summarized in Table 1, and will be explained in more detail in the following few sections.

State	Purpose of State
Trace-file generation	To generate trace-file
Splitting trace-file	To divide the trace-file into smaller pieces
Processing split trace-file	To process each individual trace-file piece
Combining processed data	To combine the processed data of each trace-file piece
Graphing data	To graph processed data

Table 1 Purpose of Each State

3.1 *'Trace-File Generation' State*

The trace files are generated by executing a .tcl file with all the simulations parameters. For detailed simulation parameters, please refer to Section 4 of this report. For the detailed coding, please refer to the Appendix.

3.2 *'Splitting Trace-File' State*

When the trace-file-generation is complete, the data within the file would normally be processed directly. However, the time required to process a complete trace-file is extremely long (>24 hours per file). This is due to the logging of different data during the data-processing. Furthermore, the use of linked-lists, as opposed to arrays, also slows down the processing time.

As result of this slow processing time, it was decided that the more efficient path would be to divide the trace-file into smaller pieces, as show in Figure 10.

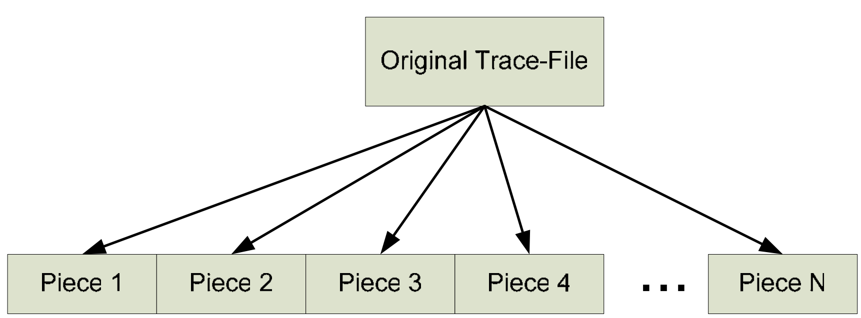


Figure 9 Splitting of trace-file into smaller pieces

The program written to perform the splitting of the trace-file is also written in Java. The UML activity diagram in Figure 11 shows the simple algorithm used.

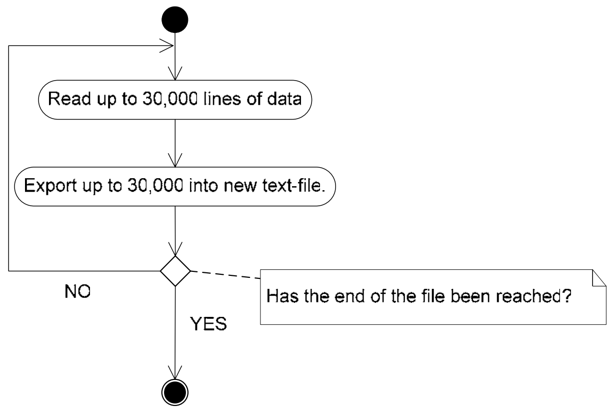


Figure 10 Splitting of trace-file into smaller pieces

3.3 'Processing Split Trace-File' State

The activity diagram for the program designed to do this is shown in Figure 12.

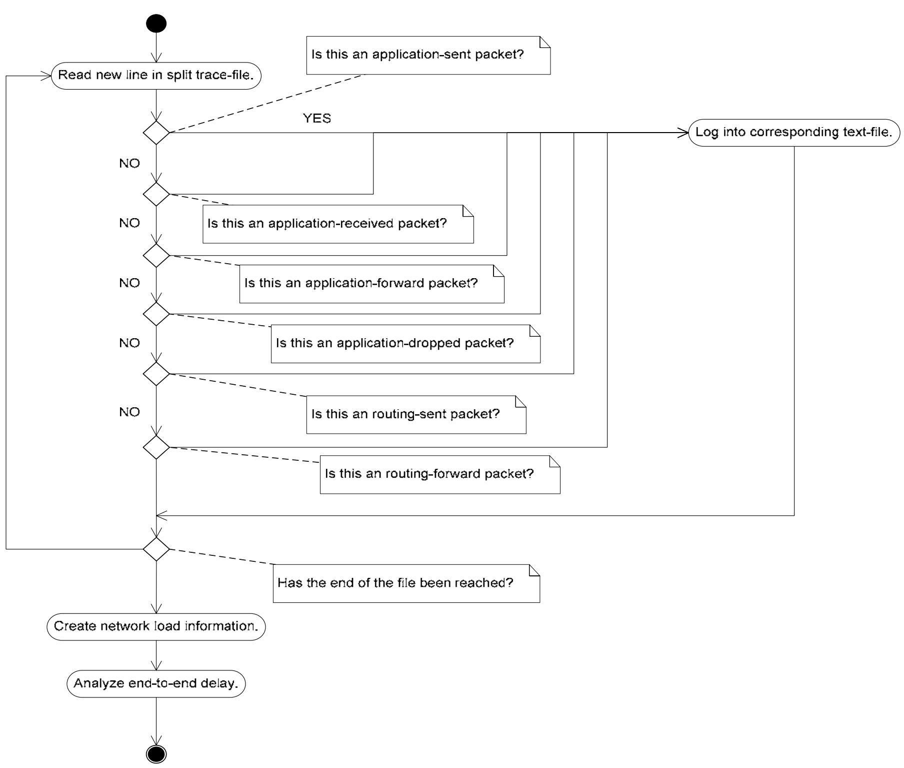


Figure 11 Processing of split trace-files

From the figure, it can be seen that three phases exist during the analysis – the data sorting phase, the network analysis phase, and the end-to-end delay analysis phase. In the first phase, the figure illustrates that six different types of packet are logged. The different checks that are performed during this phase split the data into two categories – routing-related data and application-related data. The routing-related information is used to update the routing tables, whereas the application-related data is the actual sending and forwarding of the application messages.

However, before any categorizing can be performed, it is important to first understand the logic flow of how the data is separated. There are two layers that the program will be tracking – the application layer and the routing layer. Distinguishing the difference between the two categories of data is done by analyzing the 18^{th} field in the new wireless trace-file formats [10]. If the field value is "AGT", then it means that the layer being accessed is the application layer. If the field value is "RTR", then it means that the layer being accessed is the routing layer.

Next, it is important to understand that when a message is being sent and when a message is being received, it is the application layer that is being accessed, whereas messages that are forwarded and dropped are being in reality accessing the routing layer. The reason that the forwarded and dropped are accessing the routing layer, not the application layer, is because it is the router that performs the routing of application messages based on the routing table. Shown below are examples of application-related messages (where it can be seen that the 1^{st} field defines the event such that sent messages are characterized by "s", that received messages are characterized by "r", that forwarded messages are characterized by "f", and that dropped messages are characterized by "d"):

1. Example of sent message that is application-related

s -t 103.841413918 -Hs 27 -Hd -2 -Ni 27 -Nx 923.53 -Ny 437.20 -Nz 0.00 -Ne -1.000000 -Nl **AGT** -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 27.1 -Id 29.0 -It cbr -Il 512 -If 0 -Ii 7213 -Iv 32 -Pn cbr -Pi 311 -Pf 0 -Po 2

2. Example of received message that is application-related

r -t 103.838498889 -Hs 42 -Hd 42 -Ni 42 -Nx 143.79 -Ny 356.32 -Nz 0.00 -Ne -1.000000 -Nl **AGT** -Nw --- -Ma a2 -Md 2a -Ms 0 -Mt 800 -Is 41.1 -Id 42.1 -It cbr -Il 532 -If 0 -Ii 7079 -Iv 27 -Pn cbr -Pi 474 -Pf 4 -Po 3

3. Example of forwarded message that is application-related

f -t 103.831373999 -Hs 0 -Hd 42 -Ni 0 -Nx 266.77 -Ny 395.94 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma a2 -Md 0 -Ms 33 -Mt 800 -Is 41.1 -Id 42.1 -It cbr -Il 532 -If 0 -Ii 7148 -Iv 27 -Pn cbr -Pi 477 -Pf 3 -Po 3

4. Example of dropped message that is application-related

d -t 103.831373999 -Hs 0 -Hd 42 -Ni 0 -Nx 266.77 -Ny 395.94 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma a2 -Md 0 -Ms 33 -Mt 800 -Is 41.1 -Id 42.1 -It cbr -Il 532 -If 0 -Ii 7148 -Iv 27 -Pn cbr -Pi 477 -Pf 3 -Po 3

Routing-related messages are slightly different in that in addition to the 18th field still being checked to ensure that routing layer is accessed and to the 1st field still being checked for the event, the 34th field also has to be checked. If the routing protocol is AODV, this field will have a value of "AODV". If the routing protocol is DSR, this field will have a value of "DSR". DSDV however, poses a slightly different check and instead will have the value "message". For routing-related messages, this project only inspects sent and forwarded messages. The following is an example of a sent and a forwarded message for each of the three routing protocols:

1. AODV sent routing-related message

s -t 107.175544419 -Hs 59 -Hd -2 -Ni 59 -Nx 667.34 -Ny 298.73 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma 0 -Md ffffffff -Ms 27 -Mt 800 -Is 59.255 -Id -1.255 -It **AODV** -Il 52 -If 0 -Ii 0 -Iv 24 -P aodv -Pt 0x2 -Ph 6 -Pb 40 -Pd 20 -Pds 42 -Ps 18 -Pss 69 -Pc REQUEST

2. AODV forwarded routing-related message

f -t 107.175736691 -Hs 2 -Hd 18 -Ni 2 -Nx 496.24 -Ny 746.39 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma a2 -Md 2 -Ms 22 -Mt 800 -Is 23.255 -Id 18.255 -It **AODV** -Il 44 -If 0 -Ii 0 -Iv 26 -P aodv -Pt 0x4 -Ph 8 -Pd 20 -Pds 43 -Pl 57 -Pc REPLY

3. DSR sent routing-related message

s -t 16.414914946 -Hs 12 -Hd 42 -Ni 12 -Nx 348.87 -Ny 218.47 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 12.255 -Id 32.255 -It **DSR** -Il 84 -If 0 -Ii 300 -Iv 255 -P dsr -Ph 4 -Pq 0 -Ps 6 -Pp 1 -Pn 6 -Pl 12 -Pe 32->34 -Pw 0 -Pm 0 -Pc 0 -Pb 0->0

4. DSR forwarded routing-related message

f -t 16.404069831 -Hs 13 -Hd -1 -Ni 13 -Nx 128.61 -Ny 406.71 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma 0 -Md ffffffff -Ms 33 -Mt 800 -Is 32.255 -Id 34.255 -It **DSR** -Il 68 -If 0 -Ii 296 -Iv 32 -P dsr -Ph 3 -Pq 1 -Ps 6 -Pp 0 -Pn 6 -Pl 0 -Pe 0->0 -Pw 1 -Pm 1 -Pc 0 -Pb 46->34

5. DSDV sent routing-related message

s -t 124.120999159 -Hs 49 -Hd -1 -Ni 49 -Nx 190.53 -Ny 827.07 -Nz 0.00 -Ne -1.000000 -Nl **RTR** -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 49.255 -Id -1.255 -It **message** -Il 320 -If 0 -Ii 12695 -Iv 32

6. DSDV forwarded routing-related message

No forwarded routing-related messages are tracked because the DSDV implementation in ns-2 does not perform this. Instead, the implementation receives the message and then resends the message. So, the forwarded message is actually treated as a new sent message.

All these categories are created because these are the different types of data that the project will be using as the basis for evaluating the performances of the different routing protocols. In the second phase, an evaluation of the logged data is done. This will create several new text-files, where in them exists different network characteristics. The final phase completes the analysis by evaluating the end-to-end delay of the network.

3.4 'Combining Processed Data' State

When the evaluations have been completed, the result is a set of text-files containing the network characteristics for each split piece of the original trace file. This set however, is not useful to the user yet because the information placed into each individual piece is only a portion of the total network information. So, a Java program was written to combine the individual results. Figure 13 illustrates the logic flow behind the combination.

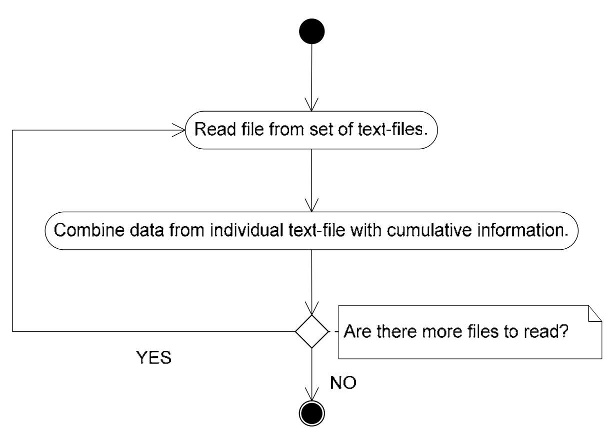


Figure 12 Combining processed pieces

3.5 ‘Graphing Data’ State

Graphs used for analysis were produced using Matlab based on the data points that were extracted from the complete processed data. Matlab was used because Microsoft Excel would not allow more than 32,000 data points to be plotted; this project produced graphs where very often there were more than 32,000 data points that had to be plotted.

4 Simulation Setup, Parameters, Performance Metrics and Previous Work

4.1 Simulation Setup

Project simulation is carried out using ns-2 as our simulation tool. We utilize a tool developed by the Monarch research group at Carnegie-Mellon University (CMU) that supports the simulation of multi-hop wireless networks [10]. The tools comes complete with physical, data link, and medium access control (MAC) layer models on ns-2. The Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer protocol. To transmit data packets, the tool uses unslotted Carrier Sense Multiple Access (CSMA) technique with Collision Avoidance (CSMA/CA) [11]. The radio model of the tool uses a shared-media radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 meters.

Wireless nodes moves randomly within the boundaries of the simulated environment. The node movements for this project are read from a node-movement file. This file is generated using CMU's node-movement generator "setdest". To generate a random movement file, the following command is used. Parameters used for this project is given in the next sub-section.

```
./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime] \
          [-x maxx] [-y maxy] > [outdir/movement-file]
```

The Monarch project utilizes the Random-Way-Point algorithm to determine the next path and speed of a node. Both speed and destination are randomly generated between their minimum and maximum values, without regard to past or present value. Also, the node moves to the destination at a constant speed.

Traffic between the wireless nodes are also read from a traffic-pattern file. Random traffic are generated using a traffic-scenario generator script [11]. Various parameters are inputted to create a random traffic file. The following command is used. Parameters used for this project is given in the next sub-section.

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections]
                             [-rate rate]
```

4.2 Simulation Parameters

For this project, we create a square flat platform of finite dimensions for simulation. Various parameters are kept permanent while others are varied to help us analyze the performance of the three protocols.

4.2.1 Variable Parameters

Three parameters are varied in this project: routing protocols, total number of nodes in the network, and the pause times of the nodes. There will be three variances of each parameters, as outlined in the following table.

Table 1 Project Variable Parameters

	Number of Nodes	**Pause Time (sec)**
AODV	20, 60, 100	1, 50, 100
DSDV	20, 60, 100	1, 50, 100
DSR	20, 60, 100	1, 50, 100

By varying the number of nodes, we introduce traffic both in terms of data and network-configurations packets. With increased number of nodes, more routes are available to reach any given node. It would be interesting to see how each routing protocol behaves with the added overhead of increased route discovery and increased route selections.

The pause time is also varied in this project to simulate the mobility of a wireless node. When the pause time is low, the wireless node is said to be very mobile, and vice versa. We will analyze how each routing protocols behaves when the mobility of the nodes are varied.

A total of 27 (3 x 3 x 3) separate trace files are generated for this project. Initially we had hoped to simulate a much larger network (e.g. 1000 nodes). However, memory constraints prevented us from carrying out the simulation. Even at 100 nodes, several simulations ended prematurely due to lack of memory.

The combined disk usage for all trace files were in excess of 1.5 Giga-bytes. The total processing time required was in excess of 72 hours running on a Pentium IV 2 GHz CPU.

4.2.2 Fixed Parameters

Although we would like to vary all parameters to constitute a thorough analysis, we had to limit our scope due to time and hardware-software constraints. The table below outlines the fixed parameters used in this project and their respective values.

Table 2 Project Fixed Parameters

Parameter Name	**Value**
General Topology	
X-Boundary	1000 meters
Y-Boundary	1000 meters
Simulation Time	150 seconds
Node Movement	
Maximum Speed	5 m/s
Number of nodes	Variable
Pause time	Variable
Traffic Generation	
Traffic Type	Constant Bit Rate (CBR)
Maximum Connections	½ of number of nodes
Rate	5 packets / second
Packet Size	512 Bytes

4.3 Performance Metrics

In this project, we are most interested in the following performance metrics:

Application Load:
The total number of combined application-related sent and application-related forwarded messages.

Received Load:
The total number of application-related received messages.

Dropped Load:
The total number of application-related dropped messages.

Routing Load:
The total number of combined routing-related sent and routing-related forwarded messages.

We will compare the three routing protocols with respect to the above four performance metrics. Also we will analyze each routing protocol individually with respect to dropped and routed loads. Finally, we will compare network metrics such as end-to-end delay and throughput of the three routing protocols. All comparisons are made with respect to project variables set forth in previous sections.

4.4 Previous Work

Before the simulation parameters were decided, research was done to see what other individual/research groups had done previously. This was an area that had been heavily investigated, and for a project, we wanted to see how we could do something different than what had already been analyzed.

4.4.1 A. Aron et al [12]

This paper by A. Aaron et. al. showed that the research performed by these two individuals placed emphasis on the performance of the routing algorithms based on the lifetime of a node on a network [12]. A node's lifetime on a network was examined in the Stanford project because the individuals wanted to see how the routing protocols would react when a node was completely disconnected from an ad-hoc network. The disconnected nodes will not re-join the network. The routing protocols examined in this were DSDV and DSR, because both the authors felt that the AODV implementation in ns-2 was not stable enough for them to use in their simulations.

From this Stanford project, we felt that there was merit in examining the behaviors of different routing protocols as nodes connect and re-connect to the network. To perform this investigation in our project, we simulated network scenarios where the nodes move randomly within the pre-defined boundaries and may occasionally be unreachable. We also extended the comparisons to include the analysis of AODV.

4.4.2 T. Dyer et al [13]

This symposium paper states that the results were based on comparisons AODV, DSR, and AODV as the background traffic and node speeds were varied only during a specific time frame [13]. Rather than analyzing a specific time period, we decided to evaluate the results for the entire simulation. By doing so, we would be able to see how the network characteristics as the networks approached a steady-state condition.

4.4.3 M. Lakshmi et al [14]

In this paper, we examined for ideas on the scope of our project. M. Lakshmi and P.E.Sankaranarayanan performed their analyses based on simulation environments that varied the network size, the area the network would cover, the node speeds, the pause times and the simulation times [14]. The routing protocols examined in this journal were DSR, DSDV, and AODV. Because our scope was for a project, we decided that we would maintain a constant network coverage area and node speed, and would keep to our decision of varying the network size and varying the pause time between node movements.

It can be seen from the three previous work, research in this area has been quite extensive and thorough. The goal of our project was to gain a good grasp of how routing protocols behaved in an ad-hoc network as network attributes were changed – namely the network size and the node mobility.

5 Results and Analysis

5.1 Comparison of Routing Protocols

In this section, we compare the performance of the three routing. The criteria being analyzed are the cumulative application load, cumulative routing load, cumulative dropped load, and cumulative received load. In the comparisons, we analyze the results based on varying the total number of nodes in the network, and based on varying the pause time between each node's movements. In some cases however, because the variable may only have minimal effect, the cases will not be displayed in the graphs. These comparisons are done to provide a higher level of introduction to the routing protocols in terms of how the protocols compare with one another. A more in-depth investigation of each individual routing protocol will be provided in Section 5.2.

5.1.1 Cumulative Application Load Analysis

5.1.1.1 Node Variance Application Load Analysis

Figure 14 and Figure 15 show the cumulative application load where the total number of nodes on the network is being varied, and where two pause times are used.

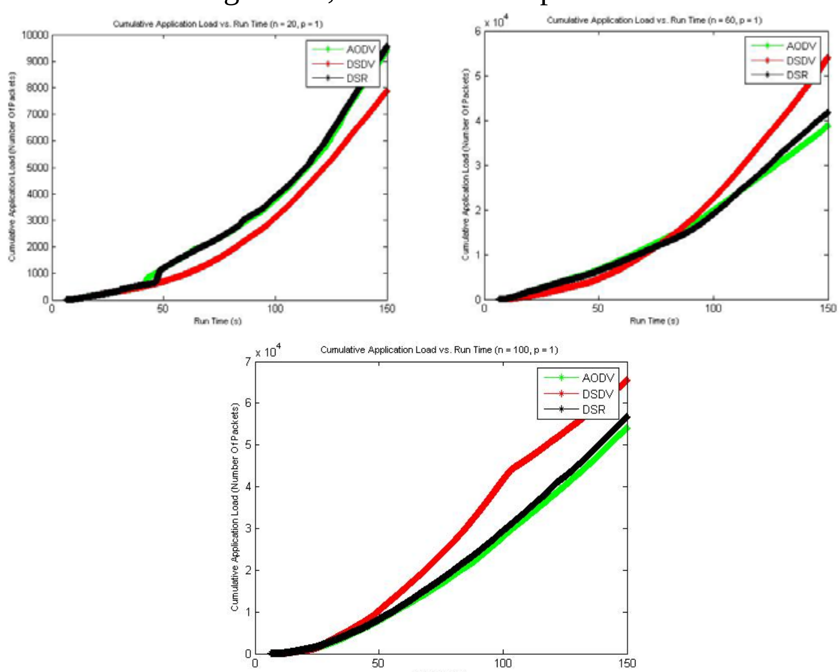


Figure 13 Cumulative Application Load With Node Variance (Pause Time = 1s)

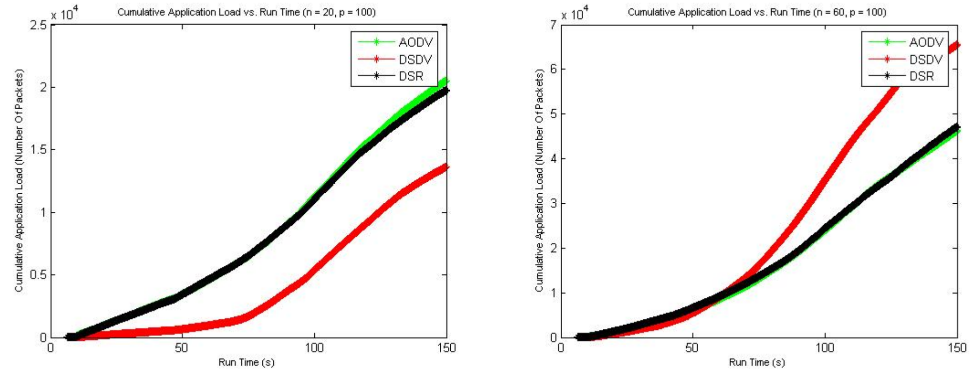


Figure 14 Cumulative Application Load With Node Variance (Pause Time = 100s)

It can be seen that as the number of nodes increases, the overhead given by DSDV is the highest. Based on the graphs, DSDV overtakes the other two routing protocols when the number of nodes is greater than 60.

5.1.1.2 Pause Time Variance Application Load Analysis

The following figure, Figure 16, shows the cumulative application load as the pause time between node movements varies.

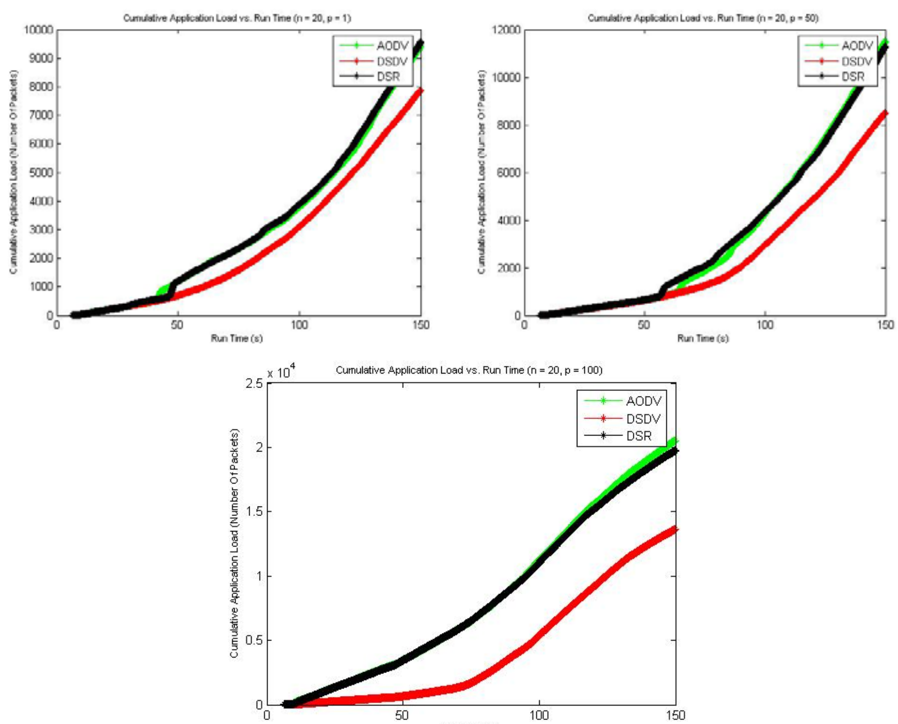


Figure 15 Cumulative Application Load With Pause Time Variance (Number of Nodes = 20)

From the figures shown, the conclusion is that as the number of nodes on a network increases, DSDV begins to increase in the amount of overhead produced. Furthermore, from the figures, it can also be concluded that the pause time is not a very significant factor on the overhead of the routing protocols. The reason that DSDV allows the most number of application-sent and application-forwarded messages is because DSDV constantly sends network configuration messages. As a result of these messages, the application messages will almost always have a valid route to travel throughout the network and hence these application messages can actually travel efficiently throughout the network.

5.1.2 Cumulative Routing Load Analysis

5.1.2.1 Node Variance Routing Load Analysis

Figure 17 and Figure 18 show the cumulative routing load as the total number of nodes on the network varies, and two constant pause times are used.

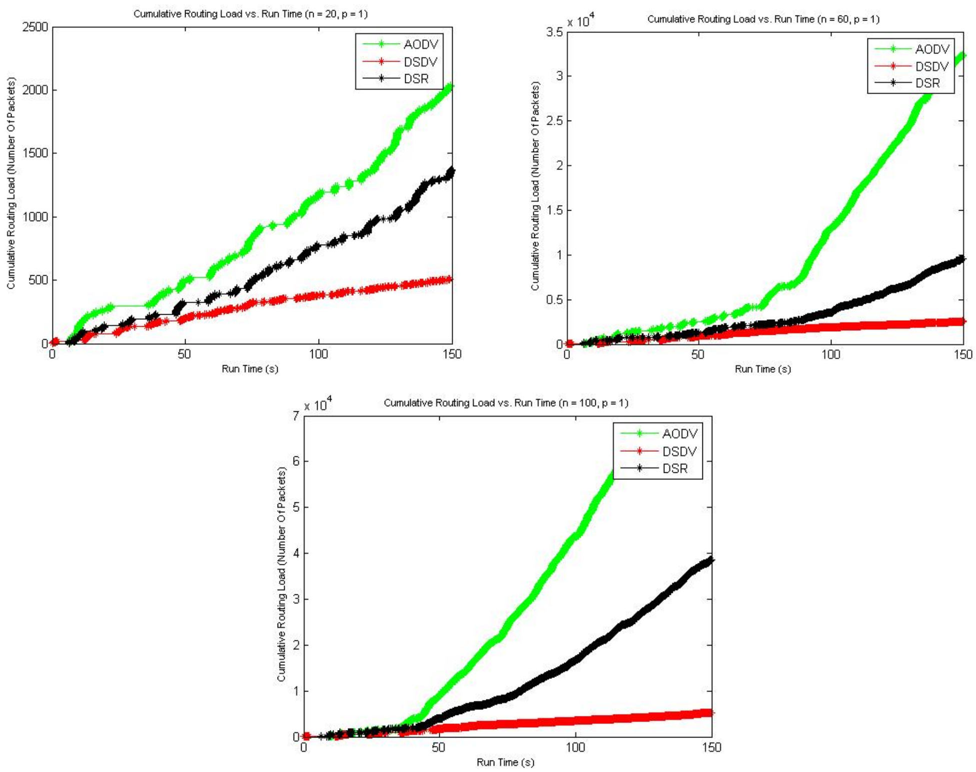


Figure 16 Cumulative Routing Load With Node Variance (Pause Time = 1s)

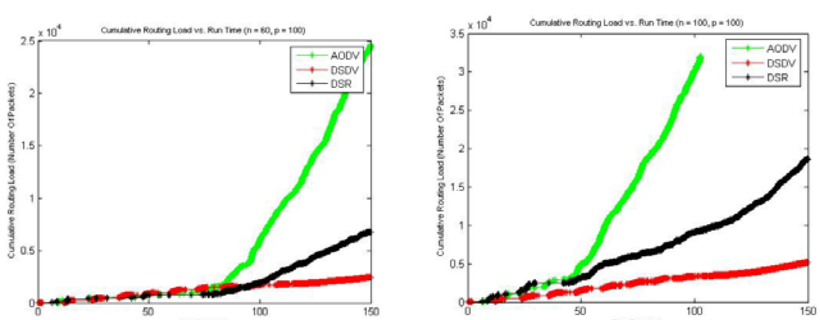


Figure 17 Cumulative Routing Load With Node Variance (Pause Time = 100s)

It can be seen that as the number of nodes increases, AODV remains the routing protocol that sends the most number of routing-messages. This is because AODV is in fact a combination of both DSR and DSDV. AODV is similar to DSR in the sense that it uses the route-discovery and route-maintenance mechanisms when a message demands it. In the case of DSDV, because AODV constantly sends messages to its neighbours to keep the links alive, this also adds to the routing load. As a result, as the graph indicates, AODV is the highest amongst the three because it incorporates both DSR and DSDV routing mechanisms.

5.1.2.2 Pause Time Variance Routing Load Analysis

The following figures, Figure 19 and Figure 20, show the cumulative routing load as the pause time between node movements varies.

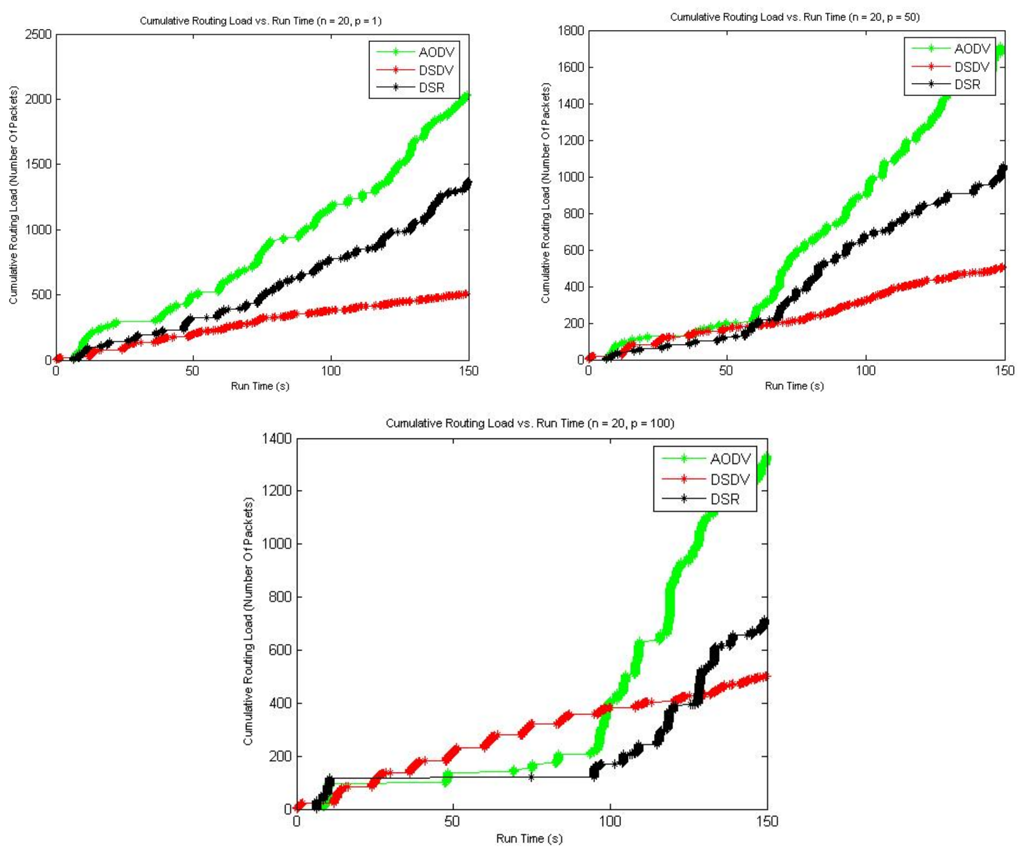


Figure 18 Cumulative Routing Load With Pause Time Variance (Number of Nodes = 20)

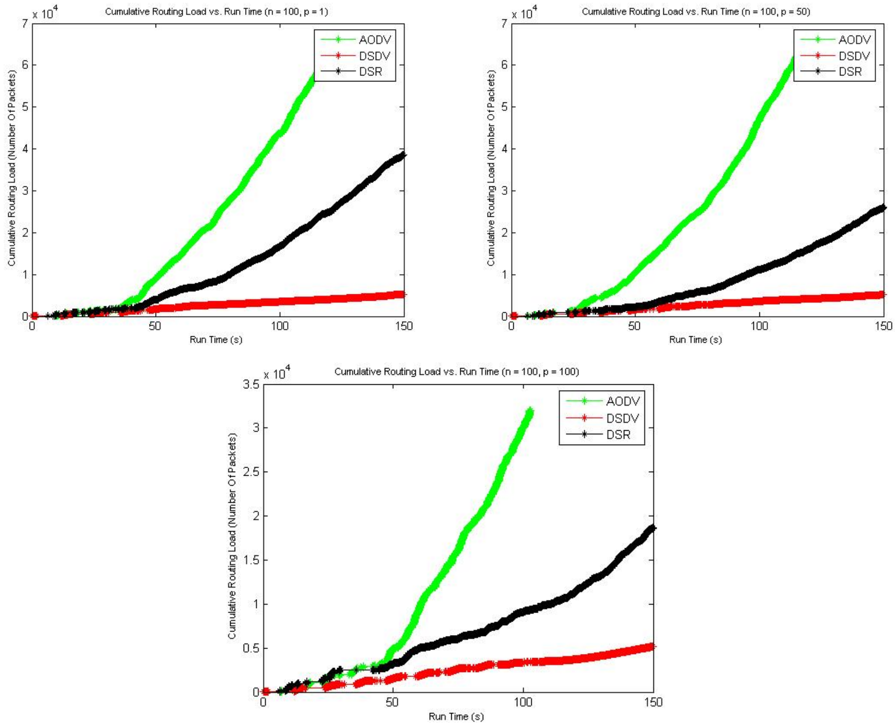


Figure 19 Cumulative Routing Load With Pause Time Variance (Number of Nodes = 100)

From the figures shown, the conclusion is that as the pause time increases, the effect on the routing load still leads to AODV being the routing protocol releasing the highest routing load into the network.

5.1.3 Cumulative Received Load Analysis

5.1.3.1 Node Variance Received Load Analysis

Figure 21 and Figure 22 show the cumulative received load as the total number of nodes on the network varies, and where two constant pause times are used.

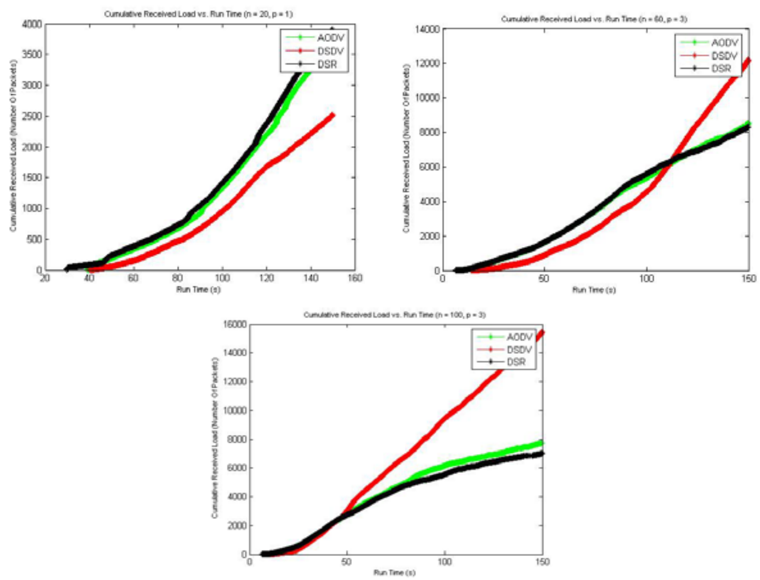
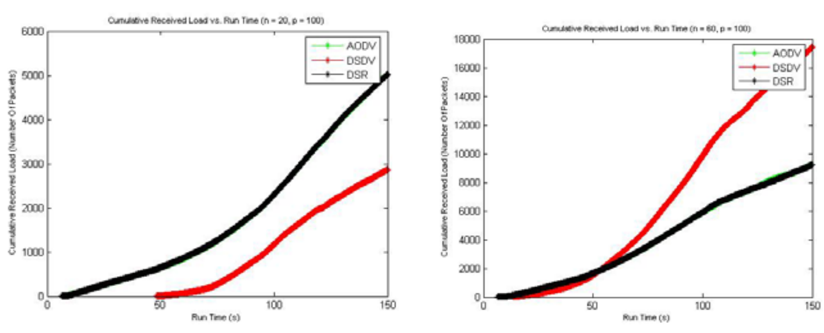


Figure 20 Cumulative Received Load With Node Variance (Pause Time = 1s)



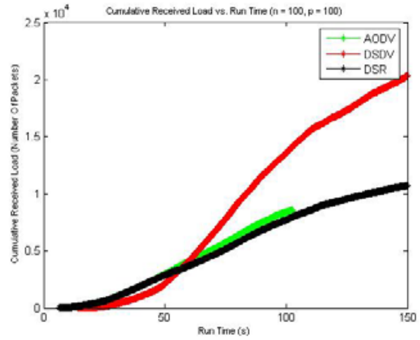


Figure 21 Cumulative Received Load With Node Variance (Pause Time = 100s)

Like the application load, it can be seen that as the number of nodes increases, the received load increases for DSDV when compared to the other two routing protocols. The reason for this however, is more to do with the timing of the overtaking of DSDV, as opposed to the number of nodes. Before DSDV overtakes the other two routing protocols, it is still in the process of sending out routing-messages to try and update its routing table. However, as time progresses, because the routing tables have already been updated, the number of application messages successfully received will increase significantly since the links already exist in the routing table. How the number of nodes affects the received load has more to do with when DSDV overtakes the other two routing protocols. As the number of nodes increases, more messages have to be sent to set the routing tables and as a result, the overtaking will take longer to take effect.

5.1.3.2 Pause Time Variance Received Load Analysis

As the pause time between node movements increases, the routing protocol placements are not changed, as shown in Figure 23.

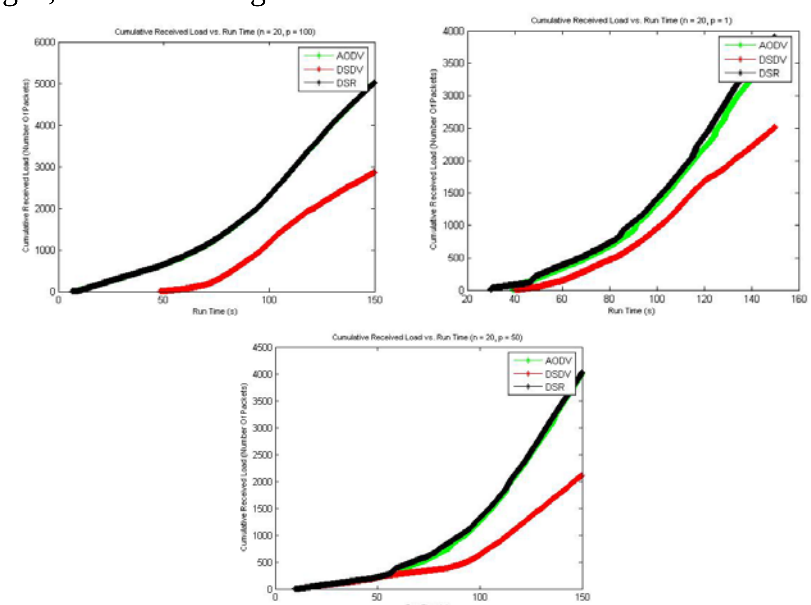


Figure 22 Cumulative Received Load With Pause Time Variance (Number of Nodes = 20)

The figures show that the routing protocols are able to maintain status quo as the pause time between node movements varies. There is no indication that a routing protocol is clearly better than the other two as time progresses in this case. The reason behind this is that having a higher pause time implies a more stable network. Hence, the more stable the network, the more the routing protocols will be able to continue their efficiency.

5.1.4 Cumulative Dropped Load Analysis

5.1.4.1 Node Variance Dropped Load Analysis

Figure 24 and Figure 25 show the cumulative dropped load as the total number of nodes on the network varies, and where two constant pause times are used.

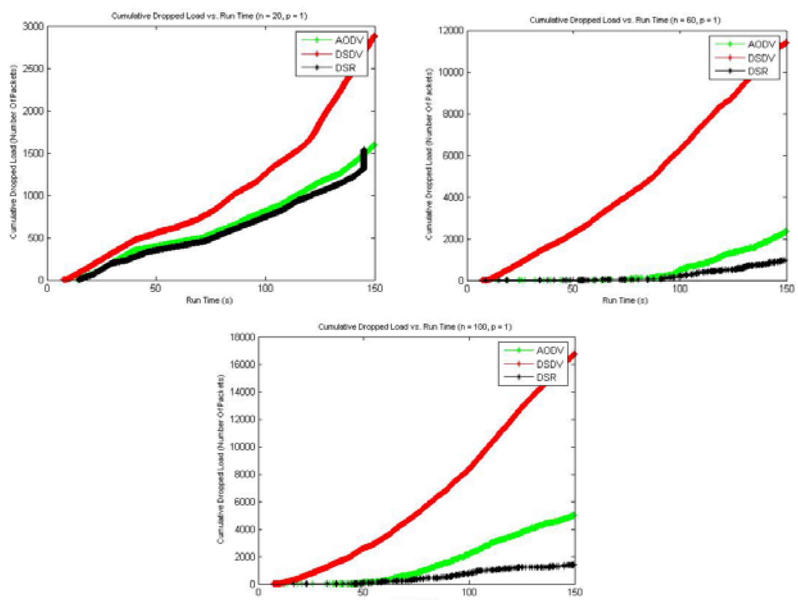


Figure 23 Cumulative Dropped Load With Node Variance (Pause Time = 1s)

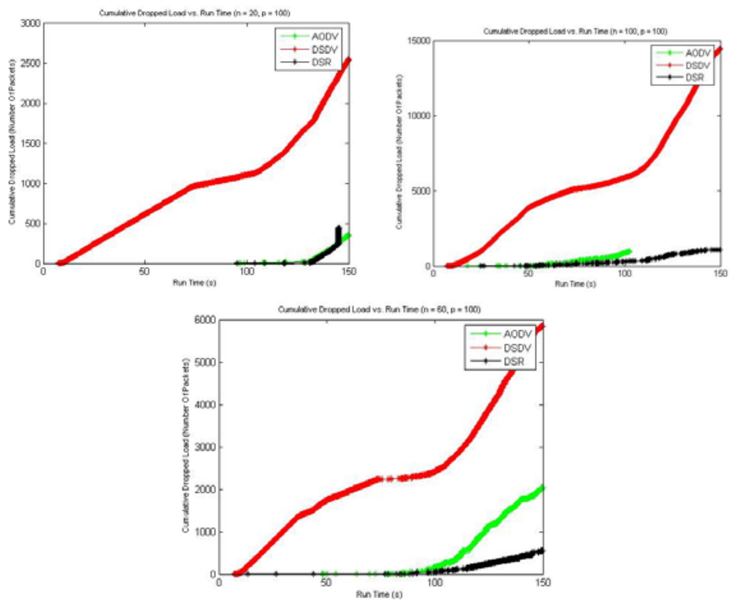


Figure 24 Cumulative Dropped Load With Node Variance (Pause Time = 100s)

It can be seen that as the number of nodes on the network increases, the routing protocol that drops the most number of messages is DSDV. Furthermore, it can be observed that the difference between the numbers of dropped packets amongst the routing protocols exponentially increases as the variable factor increases. This can be attributed to the fact that as the number of nodes on a network increases, the traffic generated throughout the network increases significantly as well. As a result, regardless of whether or not the routing protocol is efficient, the network will still experience a high number of drop packets. Of interest is that DSDV would be expected to have the lowest drop load amongst the three based on the fact that it constantly sends out inquiring messages. This however, is not entirely true. The reason that DSDV is the highest of the three dropped loads is because in between each network configuration inquiry, there is a large number of application-messages sent. As a result, if the network is overly congested, those inquiry messages may not reach the destination nodes in time, resulting in a path not being found and hence leading to the application-message being dropped.

5.1.4.2 Pause Time Variance Dropped Load Analysis

As the pause time between node movements increases, the routing protocol placements are not changed, as shown in Figure 26 and Figure 27.

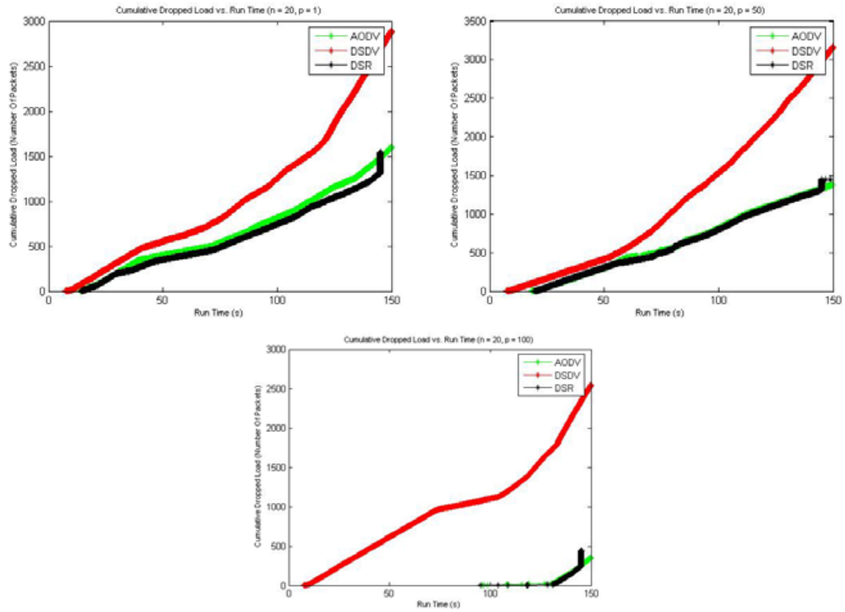


Figure 25 Cumulative Dropped Load With Pause Time Variance (Number of Nodes = 20)

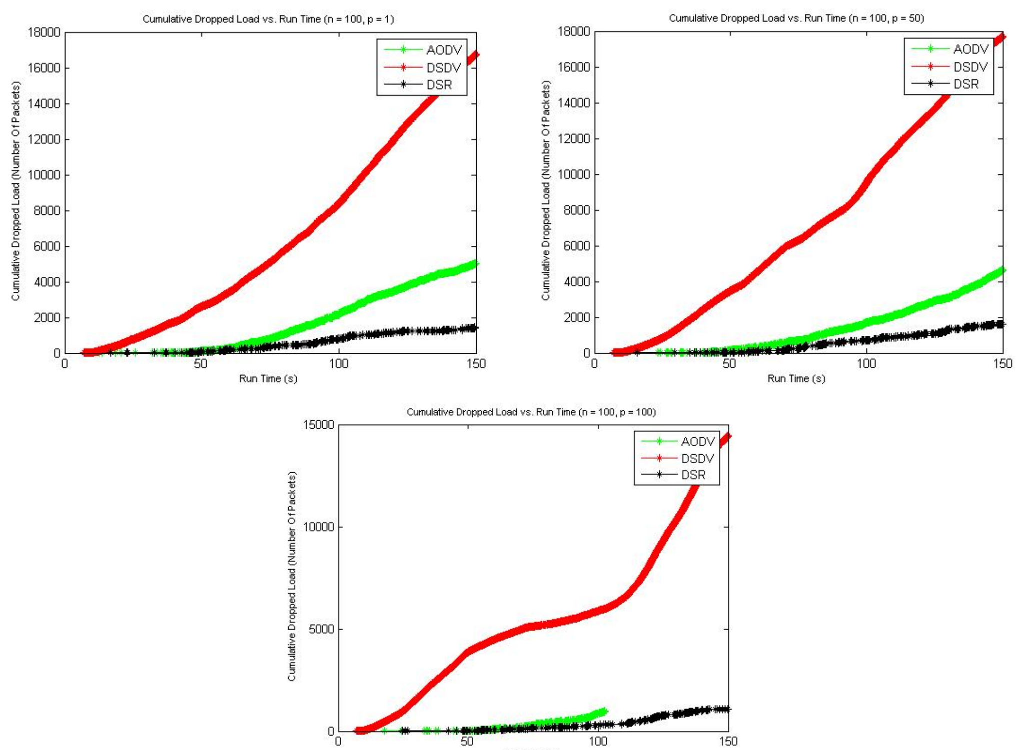


Figure 26 Cumulative Dropped Load With Pause Time Variance (Number of Nodes = 100)

Again, the figures indicate that as the pause time between the node movements increases, DSDV remains the worst routing protocol in terms of having the largest dropped load. This however, as mentioned earlier, is due more to the network traffic than the actual pause time. DSR and AODV inquire about the network topologies and link configurations when required, whereas DSDV may have to wait for the link to be established based on the constant routing-message inquiries.

5.2 Individual Routing Protocol Analysis

In this section, we examine in detail each of the three protocols, AODV, DSDV and DSR. There are two main criteria for each RP analysis, routing load and dropped packets. We vary both the total number of nodes in the network and the pause time (mobility)of each node.

5.2.1 DSR Routing Load Analysis

5.2.1.1 DSR Routing Load with node variance

The figure shown below shows the routing loads of DSR with varying number of nodes.

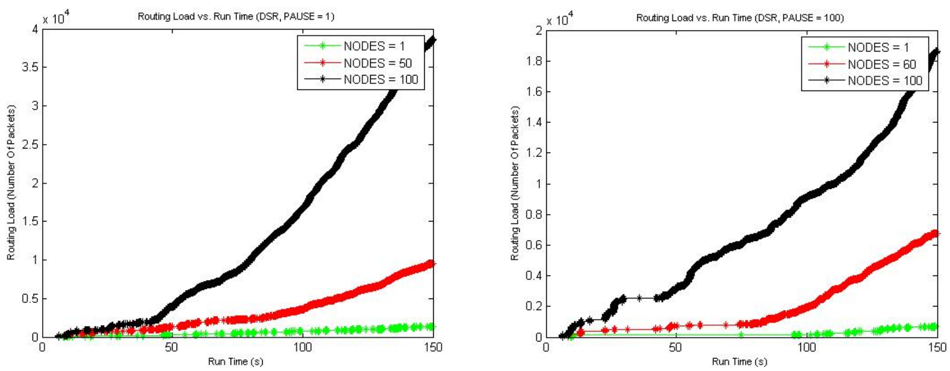


Figure 27 DSR Routing Load with varying Nodes

In both scenarios, the routing load increases significantly when the number of nodes increases, regardless of pause time.   In DSR, the routing load is heavily dependent on the total number of nodes in the network, as seen in above cases There is an exponential relationship between the routing load and the number of nodes.

5.2.1.2 DSR Routing Load with pause-time variance

The figure shown below depicts the routing loads of DSR with varying pause times.

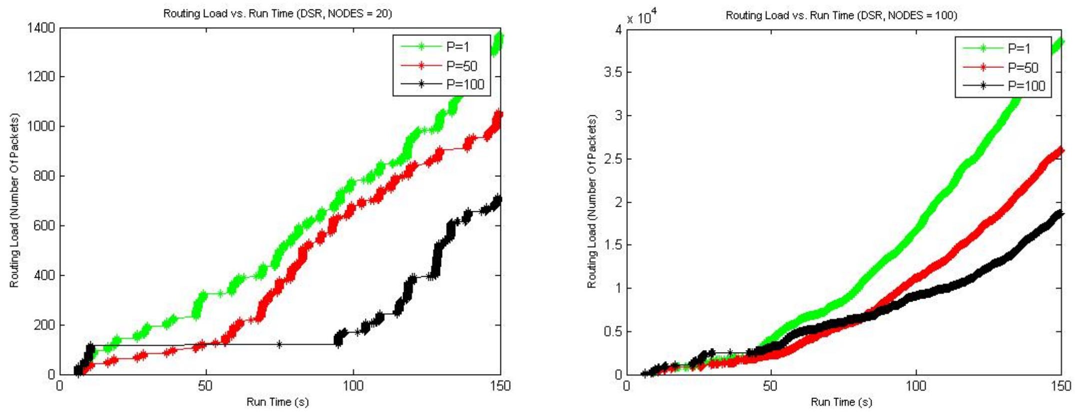


Figure 28 DSR Routing Load with varying Pause Time

We see that as the pause time decreases (e.g. increased mobility), the routing load of the network utilizing DSR increases significantly. This is true for both networks of 20 nodes and 100 nodes.

The route caches in nodes using DSR is only updated when there is a request to send a packet. When the nodes are highly mobile, new routes needs to be discovered via the route-discovery mechanisms when there is a send request. Thus there is an inverse relationship between the pause time of nodes and the routing load of the network. This is reflected in the graphs.

5.2.2 DSR Dropped Packet Analysis

5.2.2.1 DSR Dropped Load with node variance

The figure shown below shows the dropped loads of DSR with varying number of nodes.

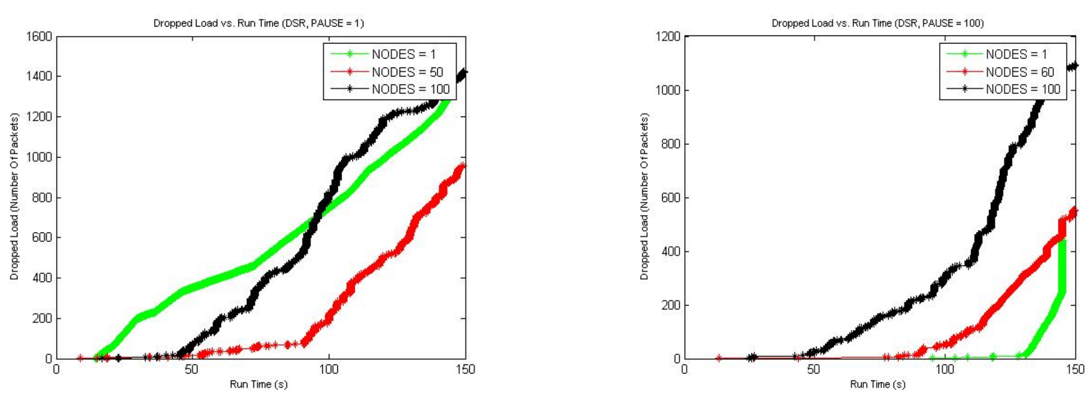


Figure 29 DSR Dropped Loads with varying Node

The number of dropped loads increase with an increase in the number of nodes. The amount of traffic in a network multiplies with an increase in nodes, thus inherently more packets will be dropped due to the increase in congestion.

5.2.2.2 DSR Dropped Load with pause-time variance

The figure shown below shows the dropped loads of DSR with varying pause times.

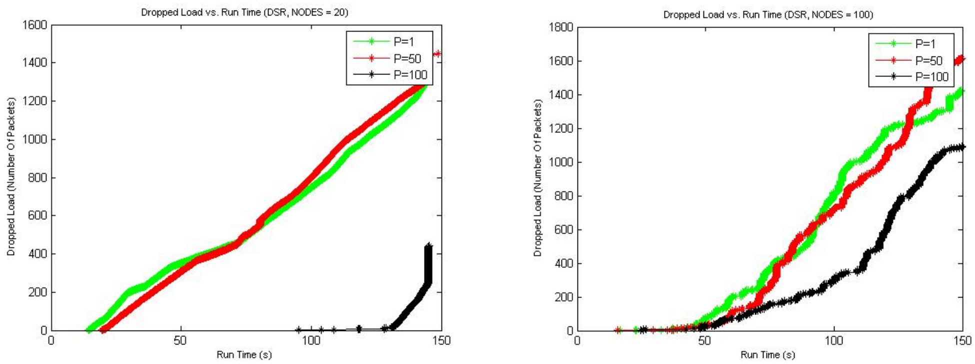


Figure 30 DSR Dropped Loads with varying Pause

As pause time decreases (increased mobility), the amount of network traffic will increase. More network-configuration packets are needed to provide nodes with updated routes, thus the more likelihood of a load being dropped by the network.

The following table summarizes the effects of varying pause time and number of nodes have on DSR.

Table 3 Summary of effects of variance on DSR

	Route Load	**Dropped Load**
Pause Time Increase	Decrease	Decrease
Pause Time Decrease	Increase	Increase
Nodes Increase	Increase	Increase
Nodes Decrease	Decrease	Decrease

5.2.3 DSDV Routing Load Analysis

5.2.3.1 DSDV Routing Load with node variance

The figure shown below shows the routing loads of DSDV with varying number of nodes.

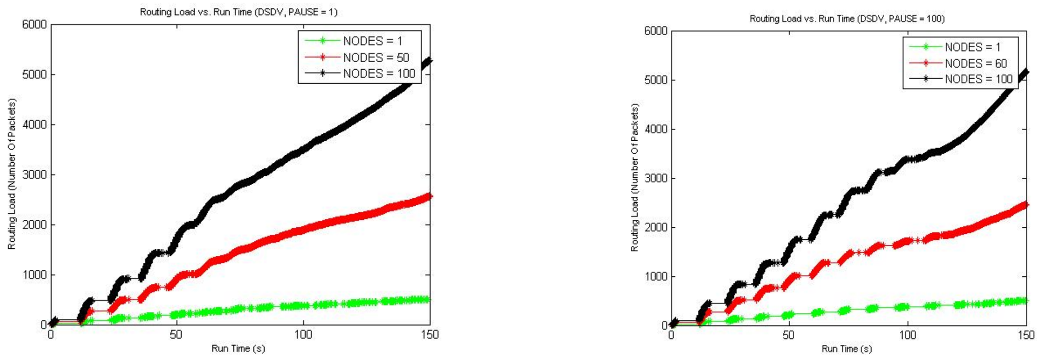


Figure 31 DSDV Routing Load with varying Nodes

In both scenarios, regardless of pause times, the routing load is significantly higher when N=100 than when N=1. The DSDV protocol specifies that network configuration packets are constantly broadcasted and existing network routes are constantly updated. Thus increasing nodes will increase the routing load exponentially.

5.2.3.2 DSDV Routing Load with pause-time variance

The figure shown below shows the routing loads of DSDV with varying pause times.

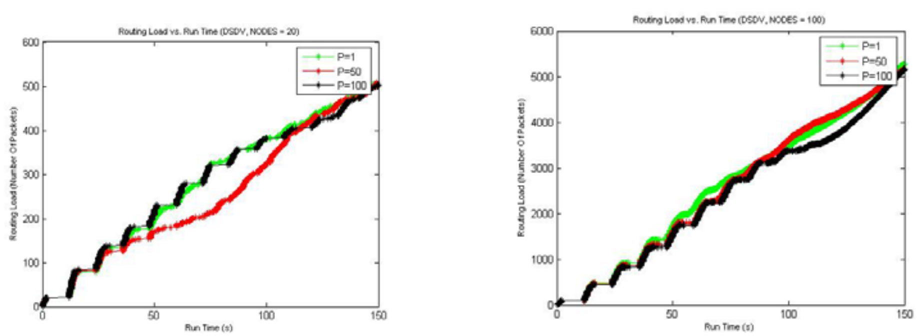


Figure 32 DSDV Routing Load with varying Pause Time

Varying the pause times in DSDV does not have a major effect on the routing loads of the network. In both scenarios, the routing load remains relatively equal. DSDV constantly broadcast network-configuration packets regardless of the pause time of an individual node. se time of nodes and the routing load of the network. This is reflected in the graphs.

5.2.4 DSDV Dropped Packet Analysis

5.2.4.1 DSDV Dropped Load with node variance

The figure shown below shows the dropped loads of DSDV with varying number of nodes.

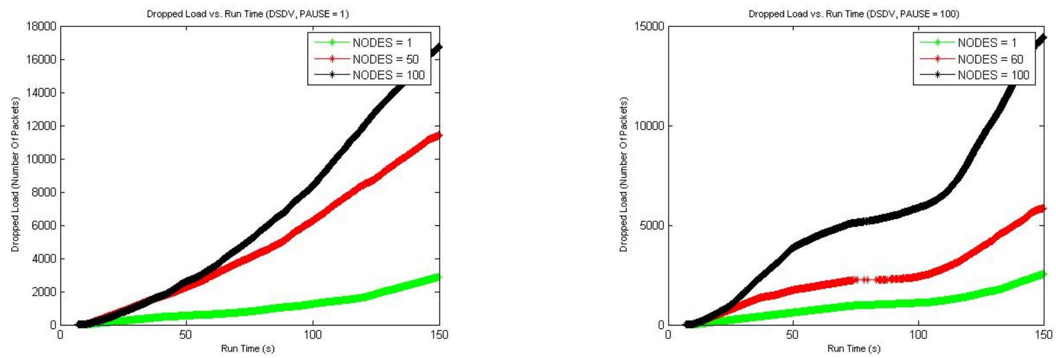


Figure 33 DSDV Dropped Loads with varying Nodes

In both scenarios, the number of dropped loads increases with number of nodes. Because nodes utilizing DSDV constantly broadcast network-configuration packets, we would expect network congestion to be an issue with increased number of nodes. Thus, the dropped load will inherently increase with network congestion.

5.2.4.2 DSDV Dropped Load with pause-time variance

The figure shown below shows the dropped loads of DSDV with varying pause times.

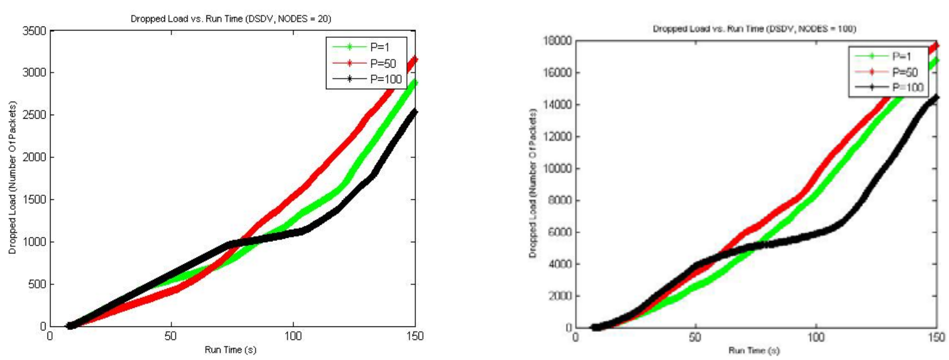


Figure 34 DSDV Dropped Loads with varying Pause

From the figure, pause times seem to have minimal effects on the dropped loads in DSDV. Differences in total dropped loads in each scenario could possibly be due to network parameters (e.g. size, simulation) and the random traffic generated for this simulation. Because nodes constantly generate network-configuration packets, variance in node pause time should have little effect on cumulative network dropped loads.

The following table summarizes the effects of varying pause time and number of nodes have on DSDV.

Table 4 Summary of effects of variance on DSDV

	Route Load	Dropped Load
Pause Time Increase	Minimal Effect	Minimal Effect
Pause Time Decrease	Minimal Effect	Minimal Effect
Nodes Increase	Increase	Increase
Nodes Decrease	Decrease	Decrease

5.2.5 AODV Routing Load Analysis

5.2.5.1 AODV Routing Load with node variance

The figure shown below shows the routing loads of AODV with varying number of nodes.

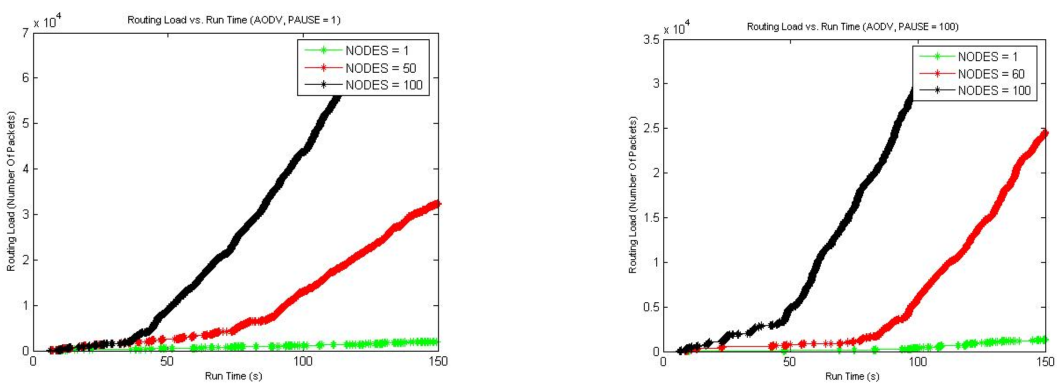


Figure 35 AODV Routing Load with varying Nodes

In both scenarios, the routing load is significantly higher when N=100 than when N=1. Routing tables are kept in each node that utilizes AODV. By increasing the number of nodes, the amount of routing information that needs to be shared between nodes will inherently increase as well.

5.2.5.2 AODV Routing Load with pause-time variance

The figure shown below shows the routing loads of AODV with varying pause times.

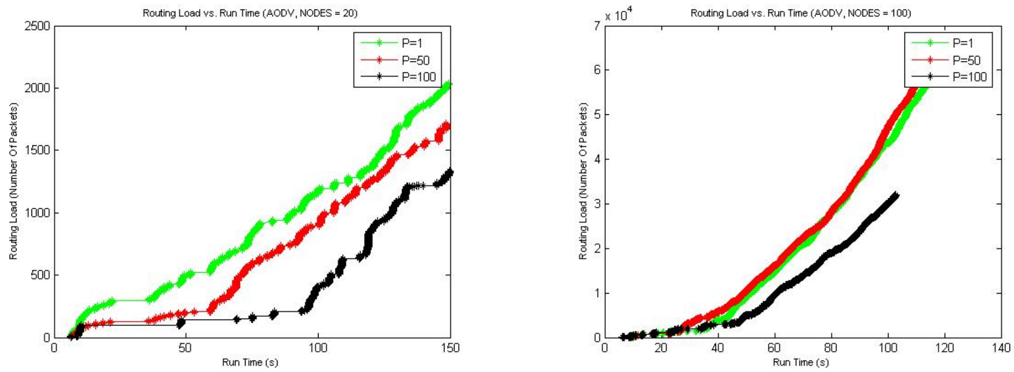


Figure 36 AODV Routing Load with varying Pause Time

In both scenarios, number of routing loads increases with decreases in pause times. This indicates that more routing is required when the nodes are more mobile. Individual nodes need to update its routing table more often due to the increase mobility and thus the routing overhead will also increase.

5.2.6 AODV Dropped Packet Analysis

5.2.6.1 AODV Dropped Load with node variance

The figure shown below shows the dropped loads of AODV with varying number of nodes.

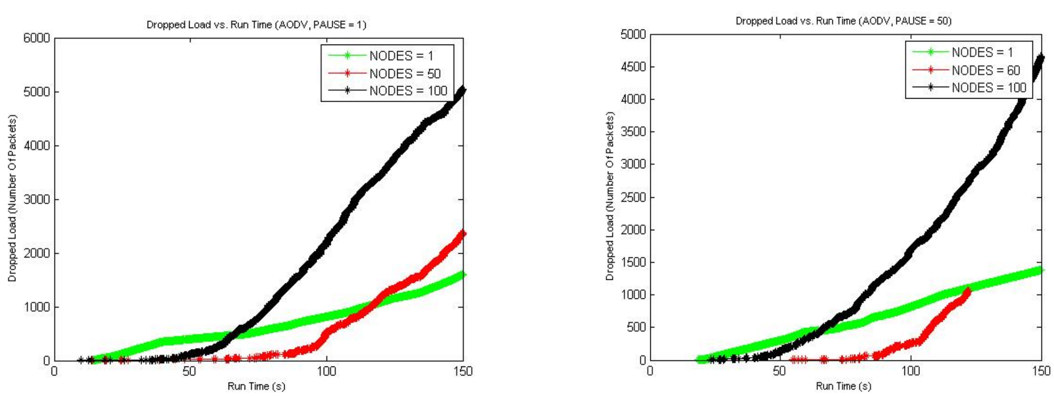


Figure 37 AODV Dropped Loads with varying Nodes

In both scenarios, regardless of pause time, the number of dropped loads increases with number of nodes. By introducing more traffic (increased nodes) into the network, congestion will occur and thus increased number of dropped loads.

5.2.6.2 AODV Dropped Load with pause-time variance

The figure shown below shows the dropped loads of AODV with varying pause times.

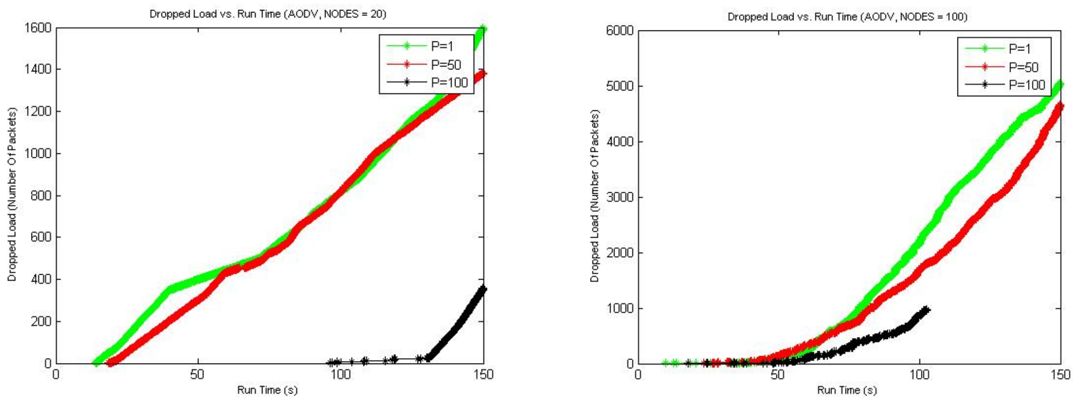


Figure 38 AODV Dropped Loads with varying Pause

From the figure, an increase in pause time will result in a decrease in dropped loads. When the nodes are highly mobile (e.g. P=1), nodes will be required to constantly updating and discovering new routes to other packets. However, when the nodes are stationary (e.g. P=100), relatively few new routes are needed to send data packets. Therefore, less traffic is in the network when the nodes are stationary and thus fewer number of dropped loads.

The following table summarizes the effects of varying pause time and number of nodes have on AODV.

Table 5 Summary of effects of variance on AODV

	Route Load	**Dropped Load**
Pause Time Increase	Decrease	Decrease
Pause Time Decrease	Increase	Increase
Nodes Increase	Increase	Increase
Nodes Decrease	Decrease	Decrease

5.3 Throughput Analysis

In this section, an analysis of the throughput is provided to ensure a comprehensive analysis of the network level.

Figure 40 represents the throughput of the network as the number of nodes is being varied.

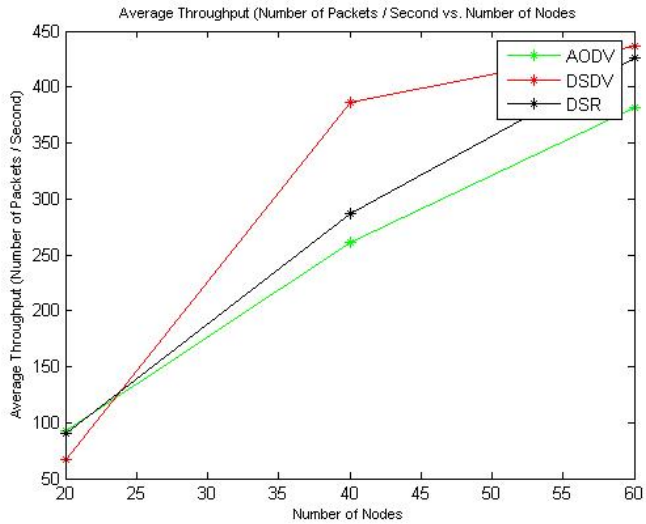


Figure 39 Throughput of Network With Varying Number of Nodes

It can be seen that for all three routing protocols, the throughput increases as the number of nodes increases. This shows that all three routing protocols are capable of handling a more congested network. As the number of nodes increases, the number of packets sent (application-related) will increase significantly, hence causing an even more congested network.

Figure 41 shows the throughput of the network as the pause time between node movements is varied.

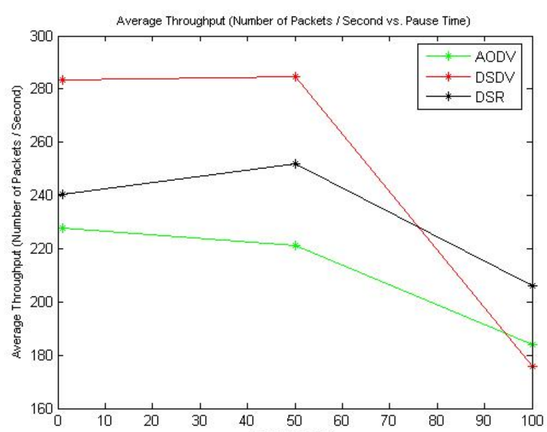


Figure 40 Throughput of Network with Varying Pause Times

As the pause time increases, the network becomes more stable and thus a decrease in the overall number of routing-related messages. As a result, the throughput of the network should decrease as the network becomes more stable.

5.4 End-To-End Delay Analysis

In this section, we will analyze the end-to-end delay that is experienced throughout the network. Figure 42 shows the end-to-end delay experienced throughout the network as the number of nodes increases.

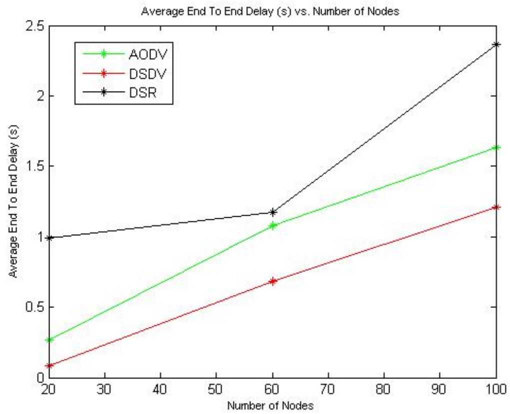


Figure 41 Average End-To-End Delay of Network with Varying Number of Nodes

Based on the above figure, we can see that as the number of nodes increases, the end-to-end delay also increases. This is accurate because it means that as the number of nodes increases, the traffic generated throughout the traffic will also be significantly larger. As a result, the time taken for a packet to travel through the congestion must increase.

The figure is also accurate in terms of showing how the different protocols behave as the number of nodes increases. Because DSR does not keep an updated routing table, the time taken for the packet to travel from the source node to the destination node will be larger than the other protocols. This is because each time a path has to be rediscovered, the message will have to wait for the protocol to detect an available path before it can propagate to the destination node. AODV is slightly better than DSR in that it keeps the routing table fairly updated until the link is no longer considered active. So, when the links are not active, AODV has to rediscover the paths for the messages to travel. DSDV, unlike the other two routing protocols, is perhaps the best in terms of handling delays since it constantly inquires about the network topology. Because the network topology is constantly refreshed, any changes will be detected fairly quickly. As a result, DSDV will be able to propagate a message from the source node to the destination node in the shortest delay time.

6 Conclusion and Future Work

In this project, we compared three routing protocols (DSDV, AODV, DSR) implemented in ns-2. We created a simulated WDS and varied parameters such as routing protocols, size of network and node mobility.

Initially we had hoped to simulate a much larger network (>1000 nodes) to present a more thorough analysis. However, due to memory issues and time constraints, this task was not carried out.

Data extraction from the trace files generated by ns-2 were needed so we could analyze our results in Matlab. Due to the enormous size of our trace files (>1.5 Giga-bytes), the Java-based parse program is unable to perform this task in a timely manner. We estimate that it would take many *weeks* running continuously on a Pentium IV 2 GHz. To solve this problem, we devised a second Java-based program to divide a trace file in smaller portions to extract the data. After all portions have been completed, a third Java-based program is used to combine the extracted data. This data is then plotted using Matlab and analyzed.

After the analysis, we found that DSR required the most nodal overhead due to the storage of all possible routes to all destinations in its cache. DSDV introduced most traffic into the network due to its periodic update of the entire network from every node.

When the network is small, we found that DSDV is most efficient in delivering data packets to destinations. When the network is large, the network should utilize AODV or DSR to avoid excessive forwarding of data packets.

This is also the case with total number of dropped packets. Data packets in a network utilizing DSDV are being dropped more frequently than AODV and DSDV, regardless of size of the network. This difference grows exponentially as the network grew larger.

However, on average, data packets routed using DSDV experienced least End-to-End delay, regardless of the size of the network. Therefore, we concluded that AODV and DSR are more reliable routing protocols while DSDV minimized End-to-End delay and maximized throughput.

Future works remain to further our analysis. Other types of traffic can be used (e.g. TCP) and trace files from existing networks, if available, can be used to accurately model a live network. Longer simulation time could also identify possible bottlenecks and routing protocol limitations. Also, complex wireless parameters (e.g. shadowing, energy model) can also be incorporated into future analysis.

7 References

[1] Agent Development Kit, http://www.madkit.org/

[2] E. Cortese, F. Ouarta, and G. Vitaglione, "Scalability and Performance of the JADE Message Transport System," *Proc. Of the AAMAS Workshop on AgentCities*, Bologna, Italy, July 2002

[3] S.I. Kumaran, *JINI Technology, An Overview*, Upper Saddle River, NJ, USA, 2002

[4] J. F. Kurose and K. W. Ross, *Computer Networking*, AW Education Group, USA, 2002.

[5] E. Chen, D. Sabaz, and W.A. Gruver, "JADE and wireless distributed environments," *IEEE International Conference on Systems, Man, and Cybernetics*, The Hague, Netherlands, 2004.

[6] E. Chen, *Jade and JXTA Extensions for Implementing a Better Distributed Systen,* Master's Thesis, School of Engineering Science, Simon Fraser University, Canada, 2005.

[7] AODV, http://moment.cs.ucsb.edu/AODV/aodv.html, accessed March, 2006.

[8] DSDV, http://www.cs.virginia.edu/~c17v/cs851-papers/dsdv-sigcomm94.pdf, accessed March, 2006.

[9] DSR, http://www.cs.cmu.edu/~dmaltz/internet-drafts/draft-ietf-manet-dsr-09.txt, accessed March, 2006.

[10] New Wireless Trace-File Format, http://k-lug.org/~griswold/NS2/ns2-trace-formats.html#wireless:new, accessed March 2006.

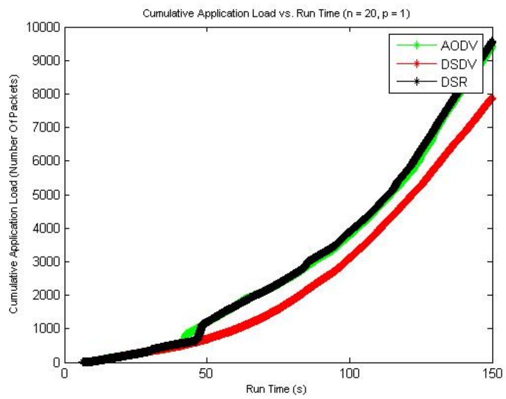
[11] The Network Simulator ns-2: Documentation, http://www.isi.edu/nsnam/ns/ns-documentation.html, accessed April 2006.

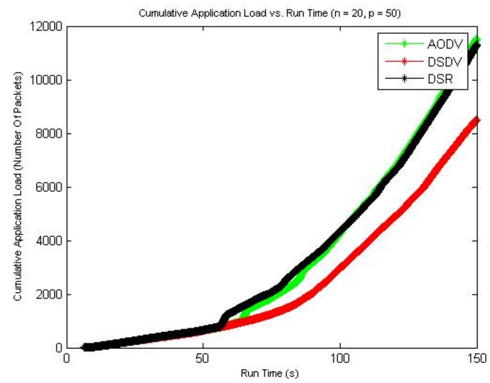
[12] Anne Aaron and Jie Weng, *"Performance Comparison of Ad-hoc Routing Protocols for Networks with Node Energy Constraints"*, Project Report, Stanford University, USA, 2001.

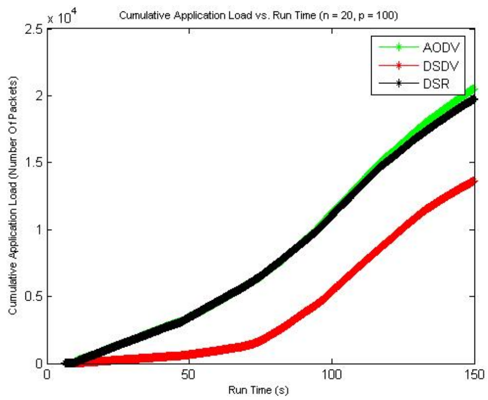
[13] Thomas D. Dyer and Rajendra V. Boppana, "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," *ACM Symposium on Mobile Ad Hoc Networking & Computing (Mobihoc),* October 2001.

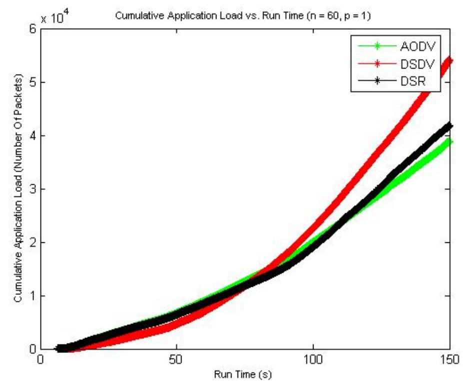
[14] M. Lakshmi and P.E.Sankaranarayanan, “Performance Analysis of Three Routing Protocols in Mobile Ad Hoc Wireless Networks,” *Asian Journal of Information Technology*, 2005.

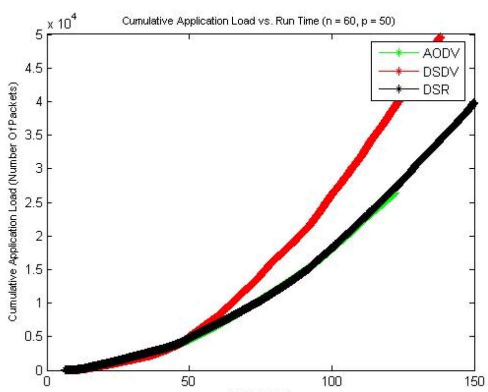
Appendix A – Graphs of Application Load Analysis

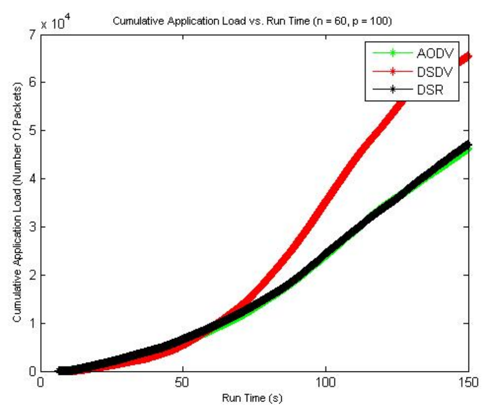


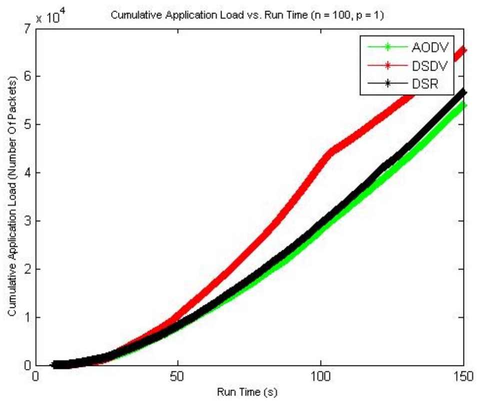


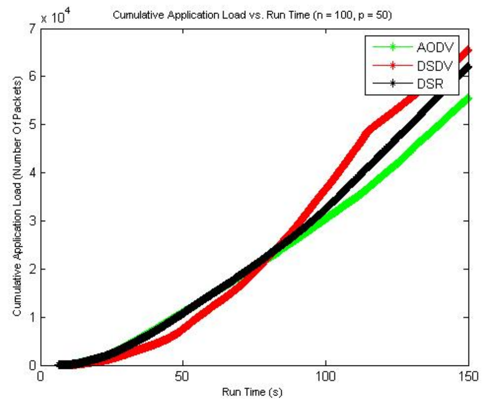


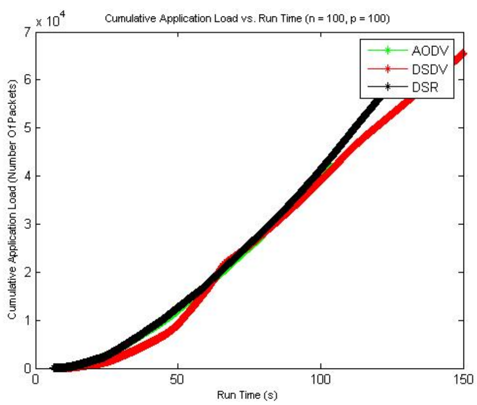




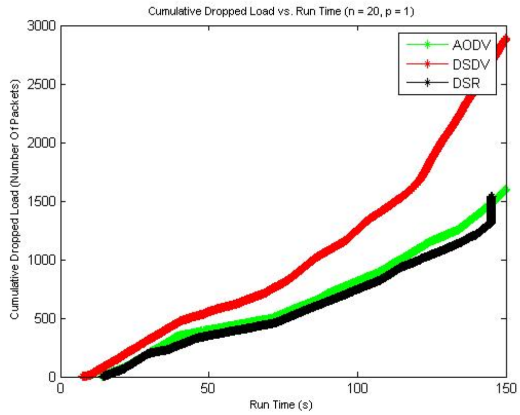


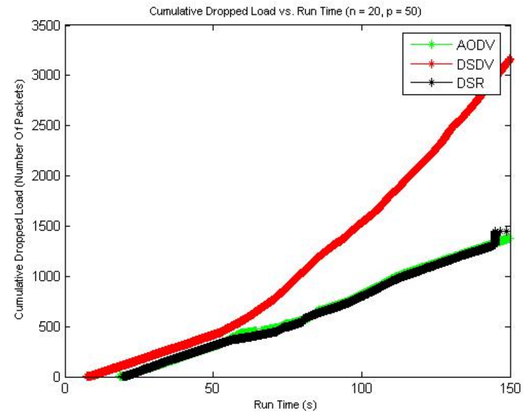

Cumulative Application Load vs. Run Time (n = 100, p = 1)
x 10^4
Cumulative Application Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

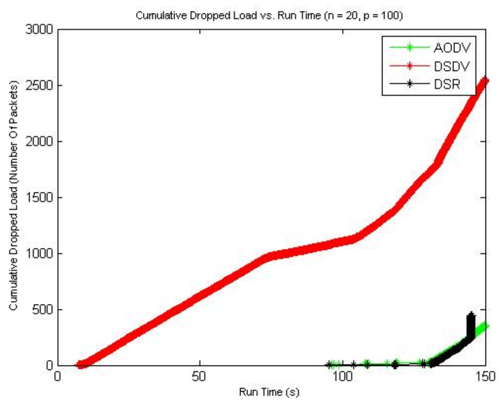

Cumulative Application Load vs. Run Time (n = 100, p = 50)
x 10^4
Cumulative Application Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

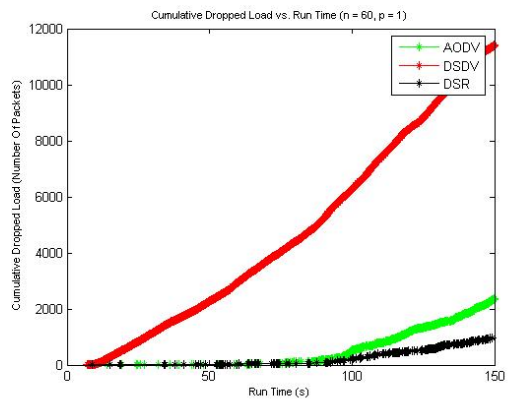

Cumulative Application Load vs. Run Time (n = 100, p = 100)
x 10^4
Cumulative Application Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

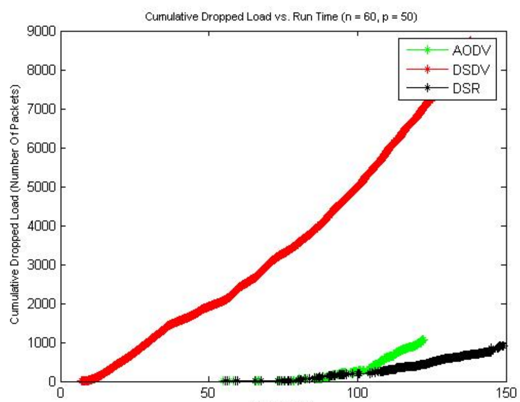
Appendix B – Graphs of Dropped Load Analysis

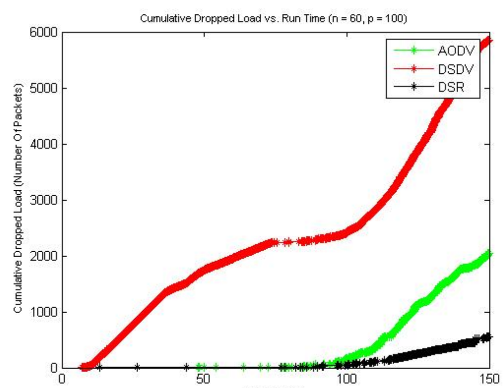


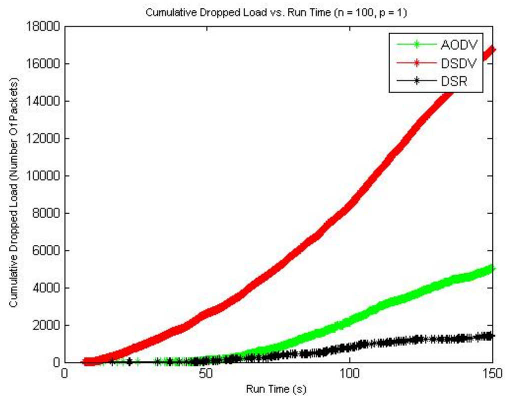


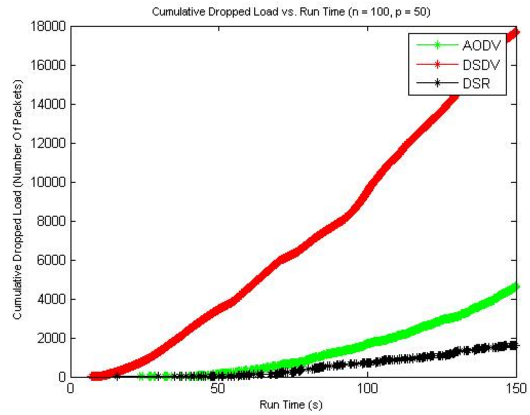


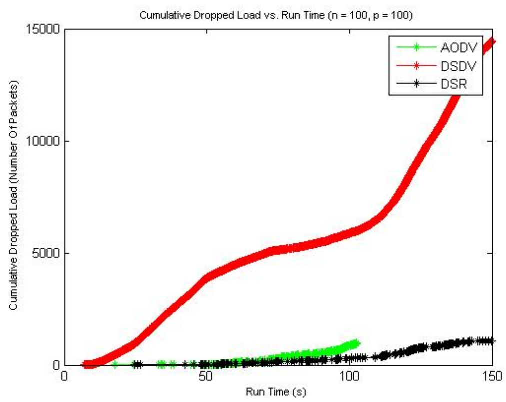




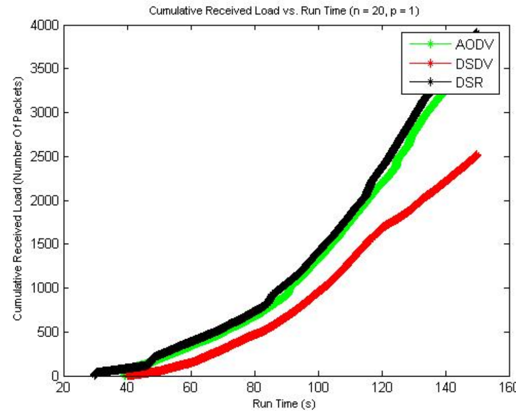


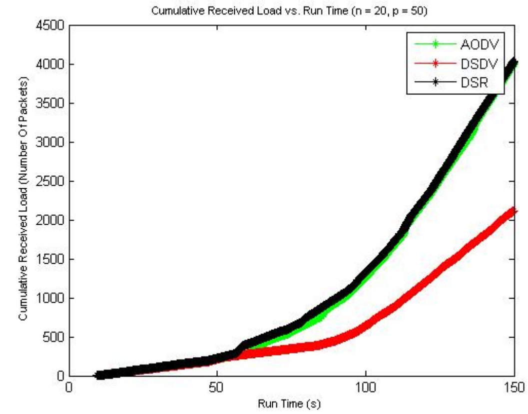

Cumulative Dropped Load vs. Run Time (n = 100, p = 1)
AODV
DSDV
DSR
Cumulative Dropped Load (Number Of Packets)
Run Time (s)

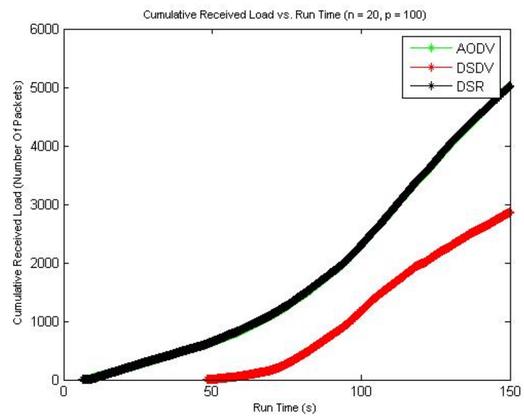

Cumulative Dropped Load vs. Run Time (n = 100, p = 50)
AODV
DSDV
DSR
Cumulative Dropped Load (Number Of Packets)
Run Time (s)

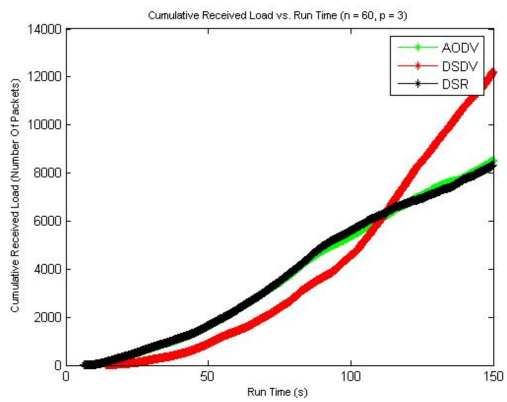

Cumulative Dropped Load vs. Run Time (n = 100, p = 100)
AODV
DSDV
DSR
Cumulative Dropped Load (Number Of Packets)
Run Time (s)

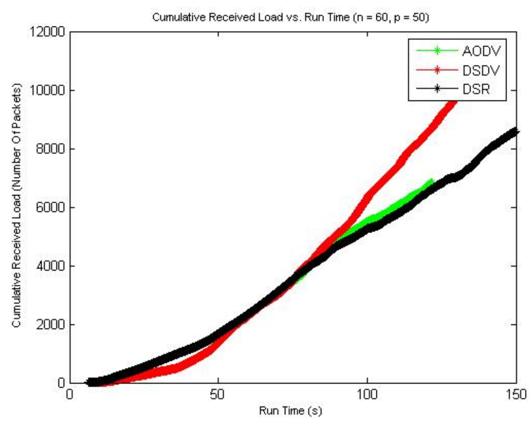
Appendix C – Graphs of Received Load Analysis

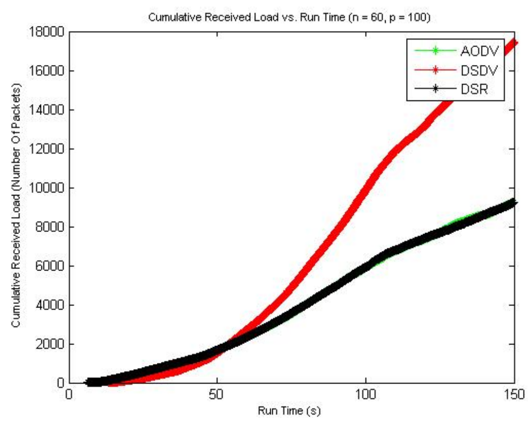


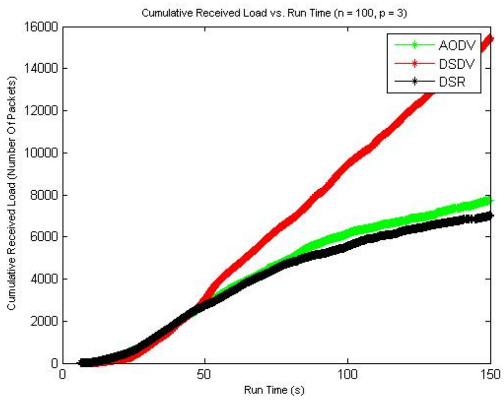


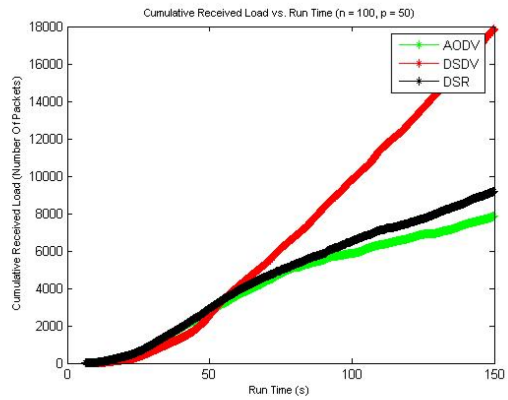


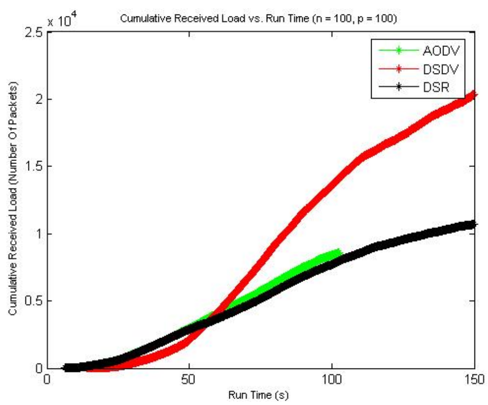




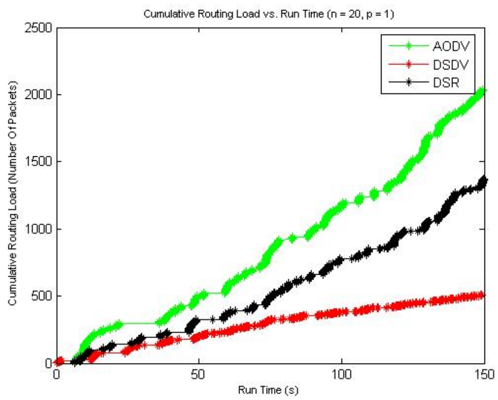


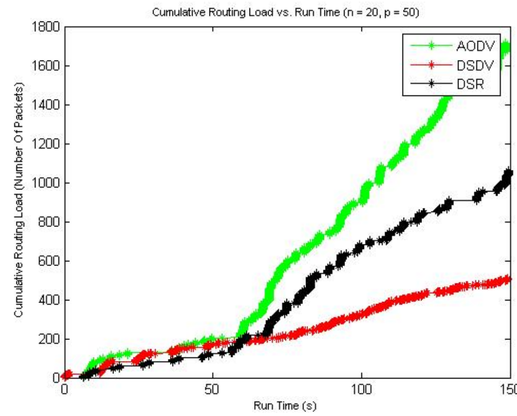

Cumulative Received Load vs. Run Time (n = 100, p = 3)
Cumulative Received Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

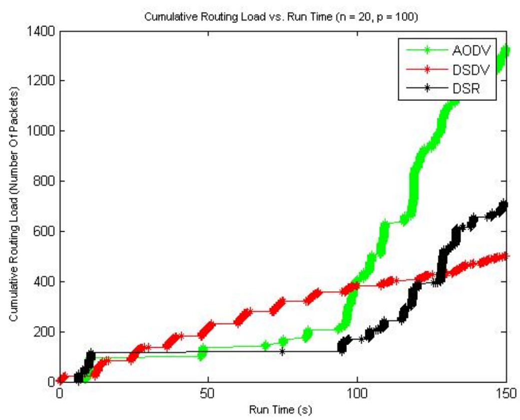

Cumulative Received Load vs. Run Time (n = 100, p = 50)
Cumulative Received Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

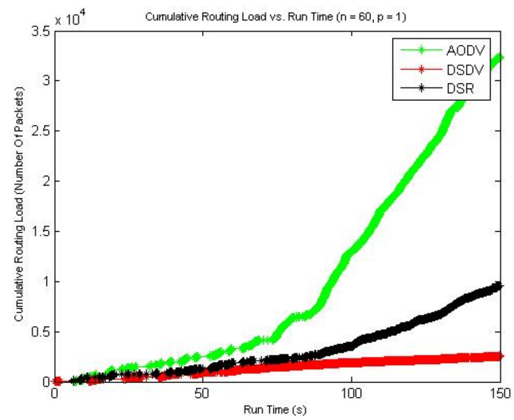

Cumulative Received Load vs. Run Time (n = 100, p = 100)
x 10^4
Cumulative Received Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

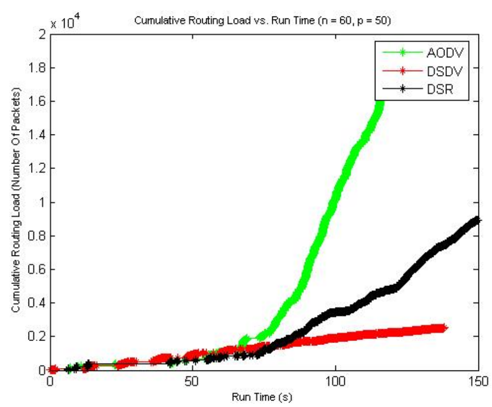
Appendix D – Graphs of Routing Load Analysis

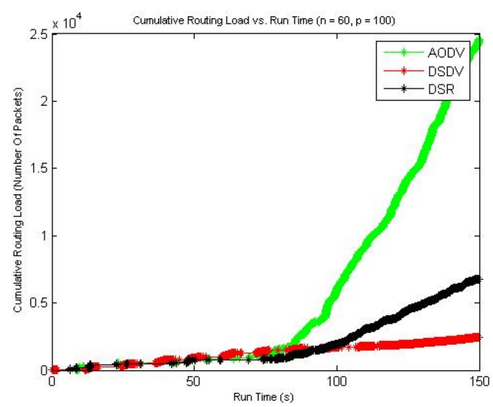


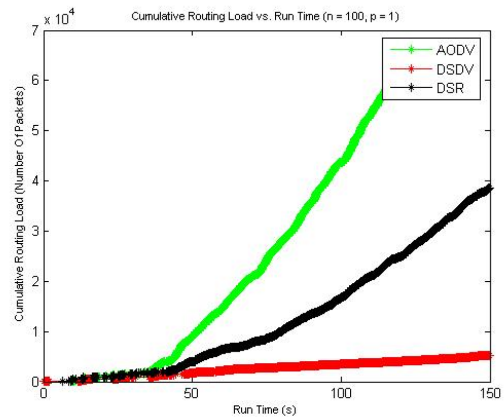


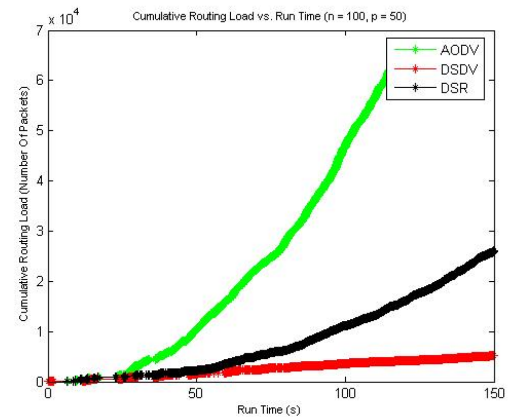


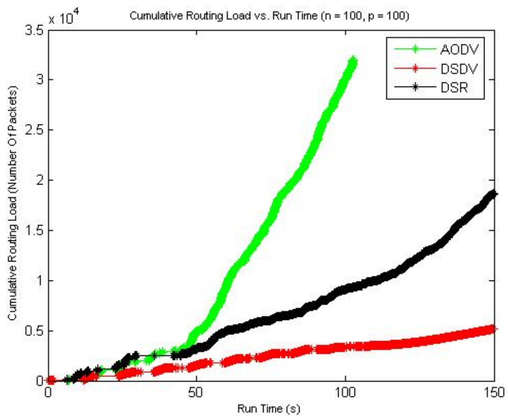




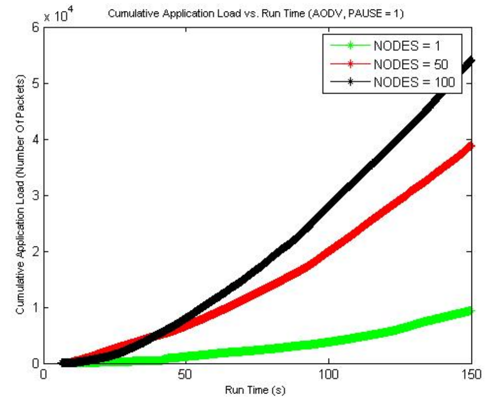


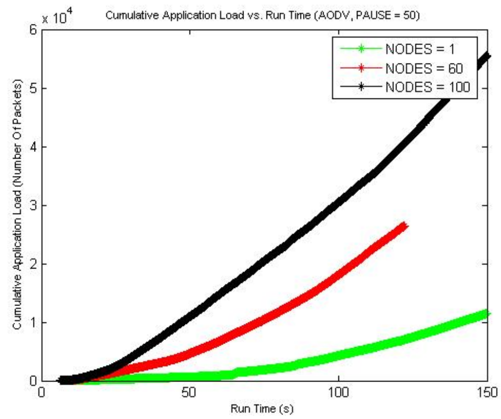

Cumulative Routing Load vs. Run Time (n = 100, p = 1)
Cumulative Routing Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

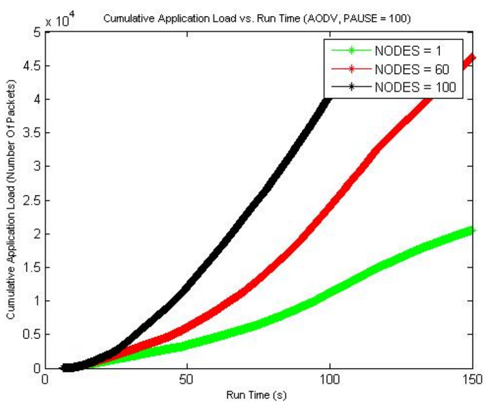

Cumulative Routing Load vs. Run Time (n = 100, p = 50)
Cumulative Routing Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

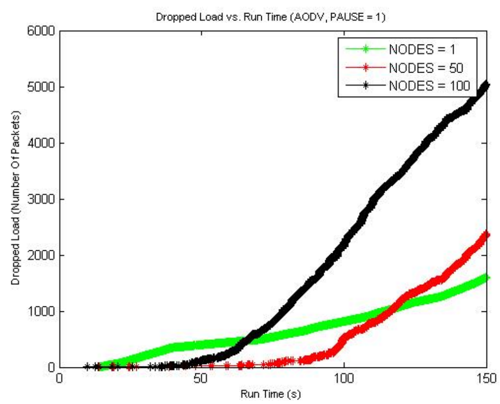

Cumulative Routing Load vs. Run Time (n = 100, p = 100)
Cumulative Routing Load (Number Of Packets)
Run Time (s)
AODV
DSDV
DSR

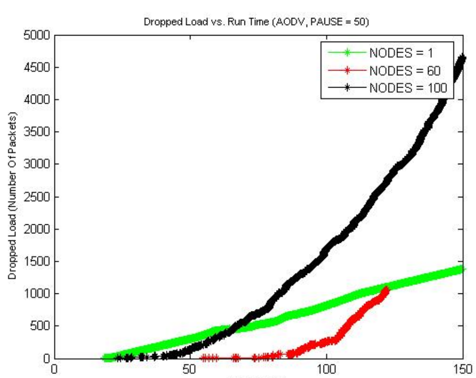
Appendix E – Graphs of AODV Analysis (Varying Total Number of Nodes on Network)

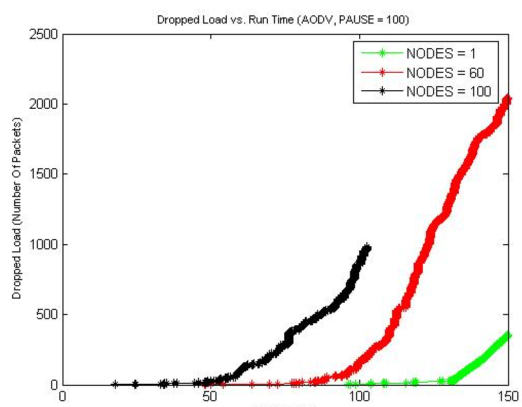


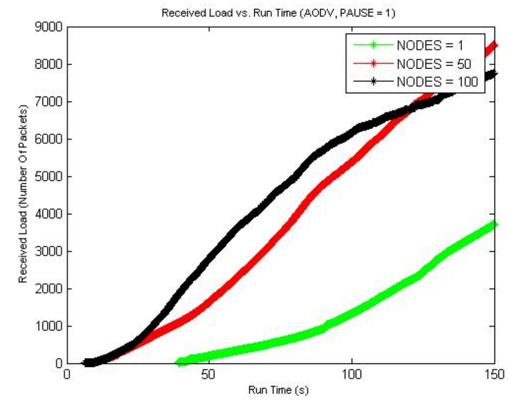


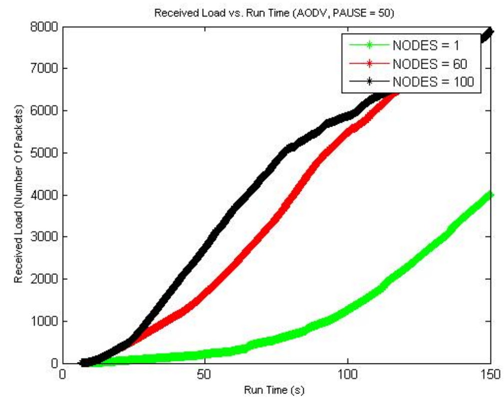


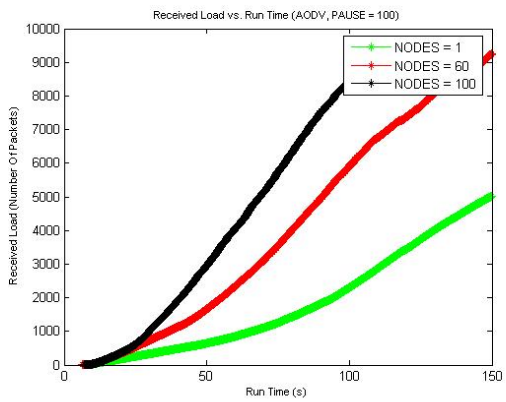


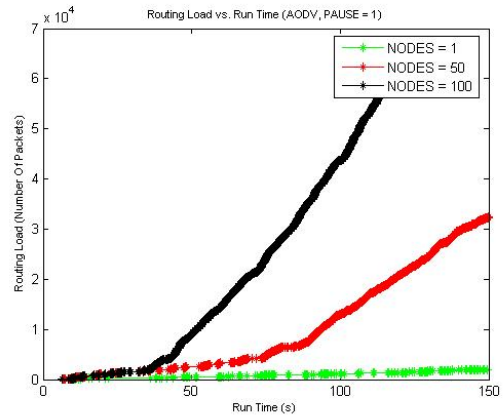


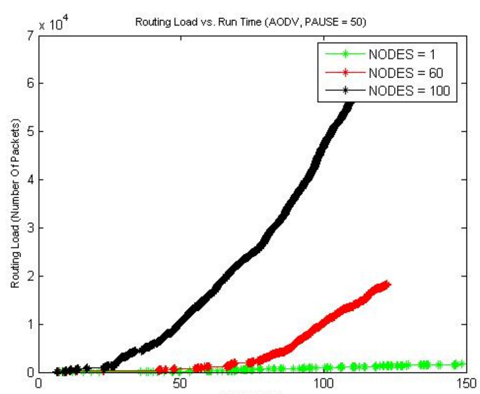


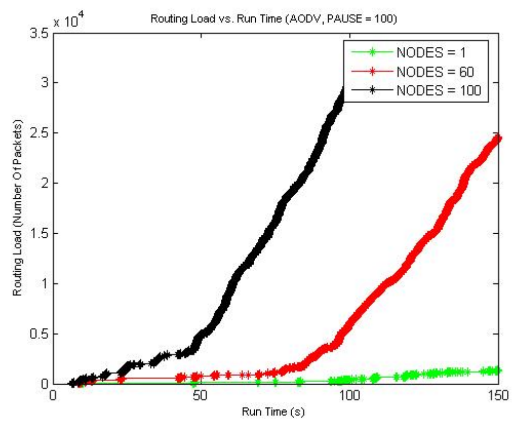

Received Load vs. Run Time (AODV, PAUSE = 1)
NODES = 1
NODES = 50
NODES = 100
Received Load (Number Of Packets)
Run Time (s)


Received Load vs. Run Time (AODV, PAUSE = 50)
NODES = 1
NODES = 60
NODES = 100
Received Load (Number Of Packets)
Run Time (s)

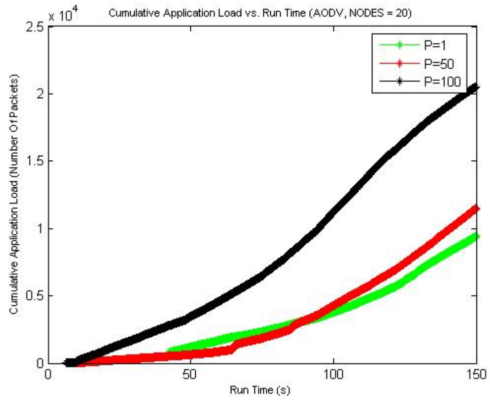

Received Load vs. Run Time (AODV, PAUSE = 100)
NODES = 1
NODES = 60
NODES = 100
Received Load (Number Of Packets)
Run Time (s)

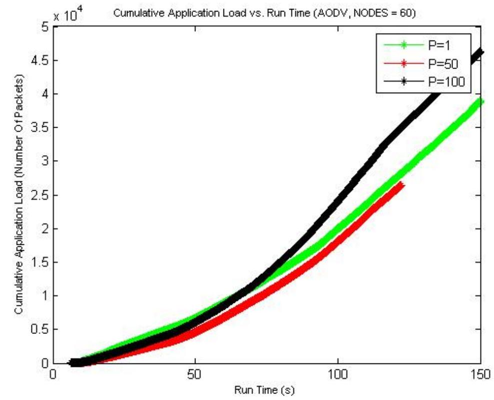

Routing Load vs. Run Time (AODV, PAUSE = 1)
x 10^4
NODES = 1
NODES = 50
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

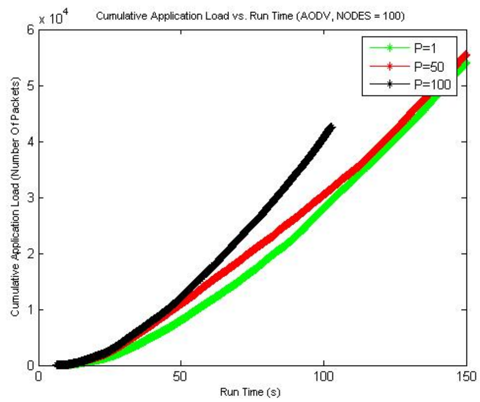

Routing Load vs. Run Time (AODV, PAUSE = 50)
x 10^4
NODES = 1
NODES = 60
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

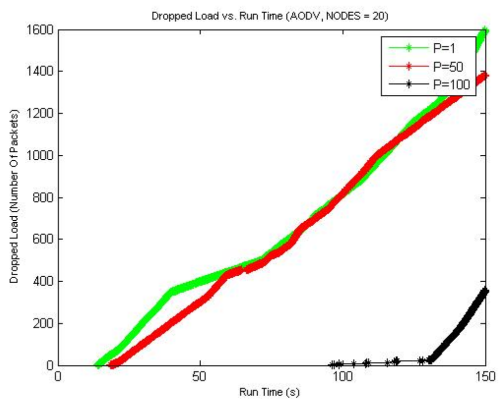

Routing Load vs. Run Time (AODV, PAUSE = 100)
x 10^4
NODES = 1
NODES = 60
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

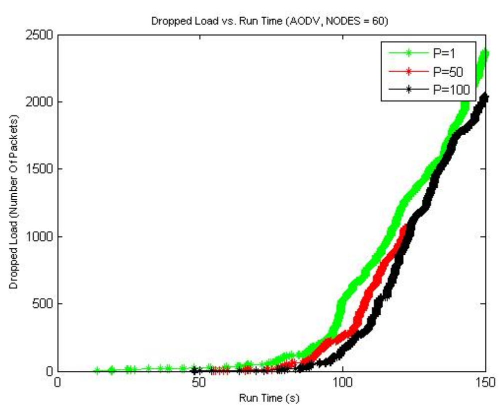
Appendix F – Graphs of AODV Analysis (Varying Pause Time)

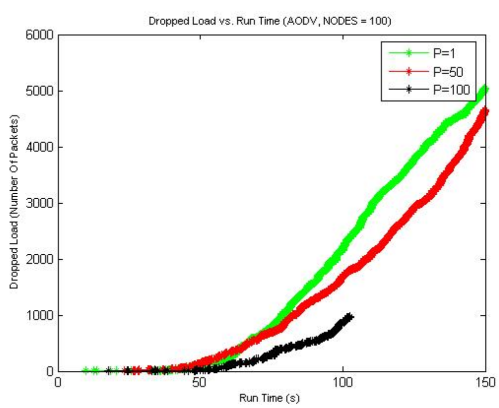


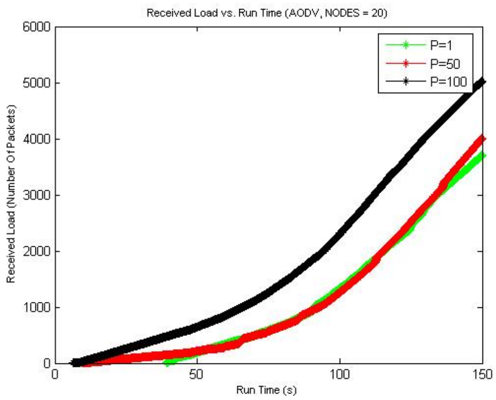


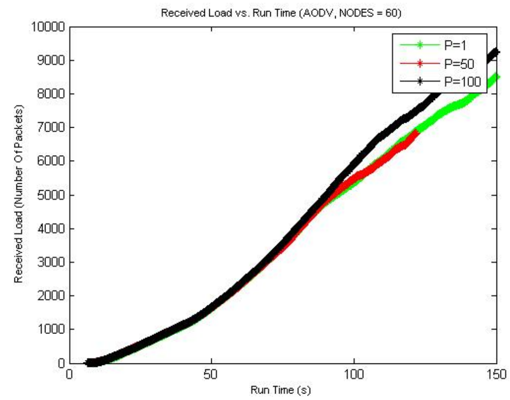


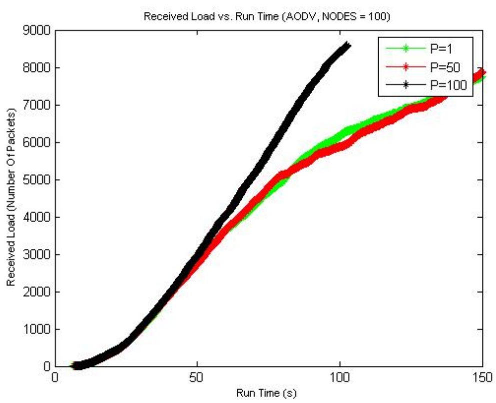


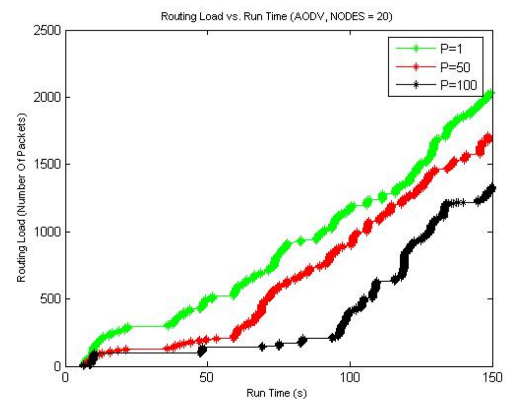


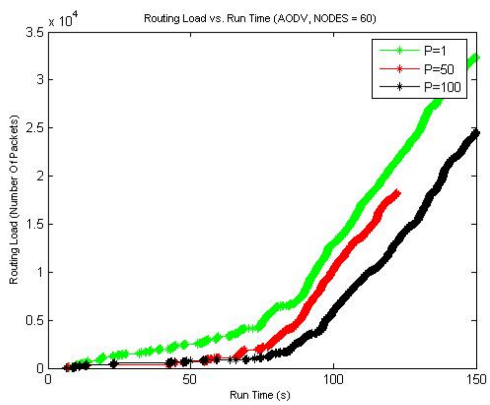


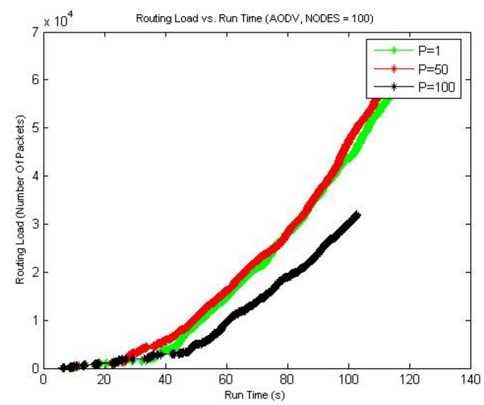

Received Load vs. Run Time (AODV, NODES = 20)
P=1
P=50
P=100
Received Load (Number Of Packets)
Run Time (s)


Received Load vs. Run Time (AODV, NODES = 60)
P=1
P=50
P=100
Received Load (Number Of Packets)
Run Time (s)

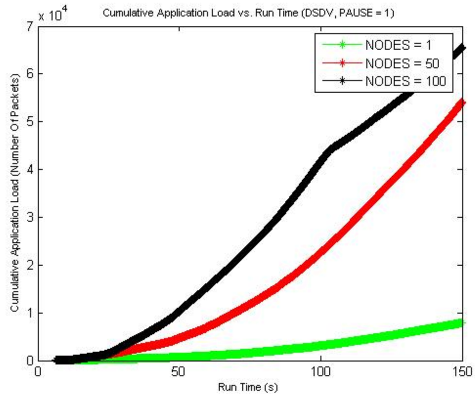

Received Load vs. Run Time (AODV, NODES = 100)
P=1
P=50
P=100
Received Load (Number Of Packets)
Run Time (s)

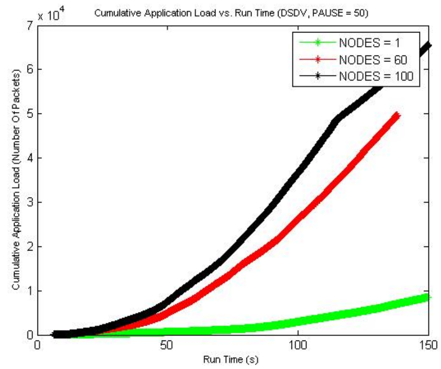

Routing Load vs. Run Time (AODV, NODES = 20)
P=1
P=50
P=100
Routing Load (Number Of Packets)
Run Time (s)

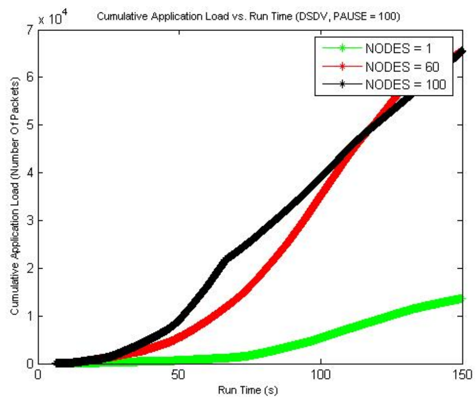

Routing Load vs. Run Time (AODV, NODES = 60)
x 10^4
P=1
P=50
P=100
Routing Load (Number Of Packets)
Run Time (s)

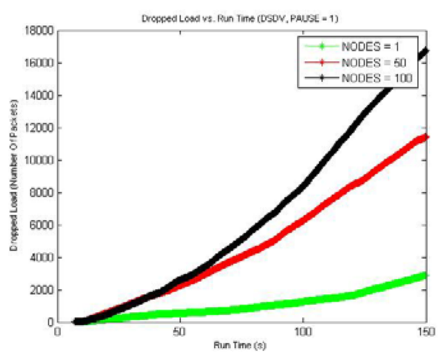

Routing Load vs. Run Time (AODV, NODES = 100)
x 10^4
P=1
P=50
P=100
Routing Load (Number Of Packets)
Run Time (s)

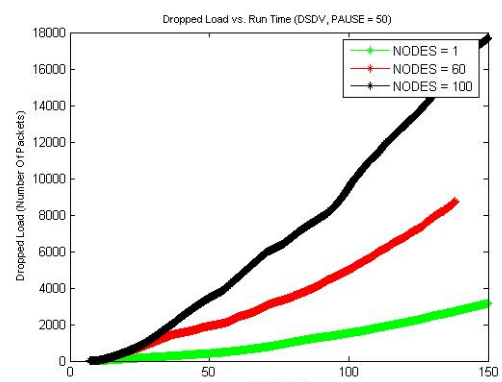
Appendix G – Graphs of DSDV Analysis (Varying Total Number of Nodes on Network)

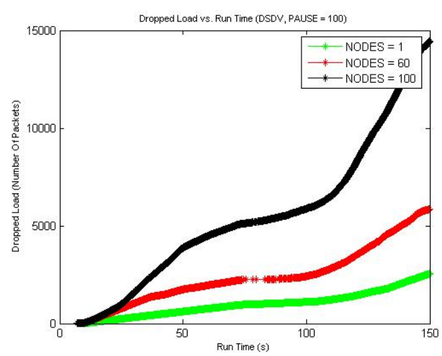


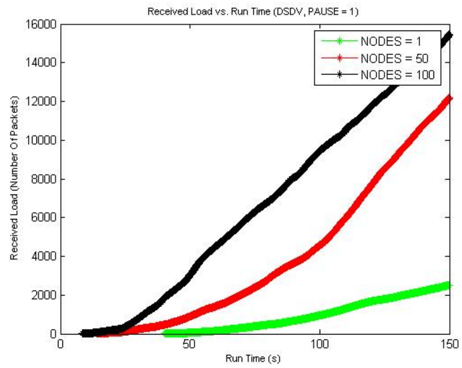


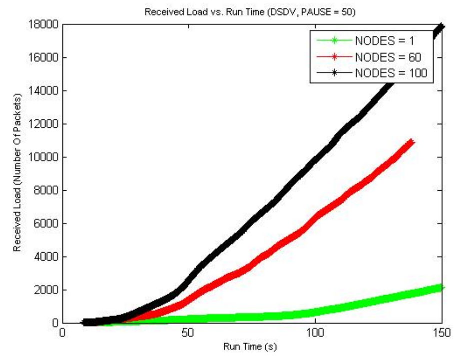


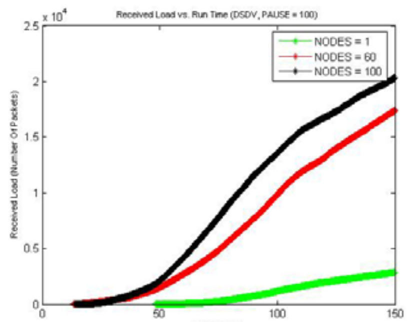


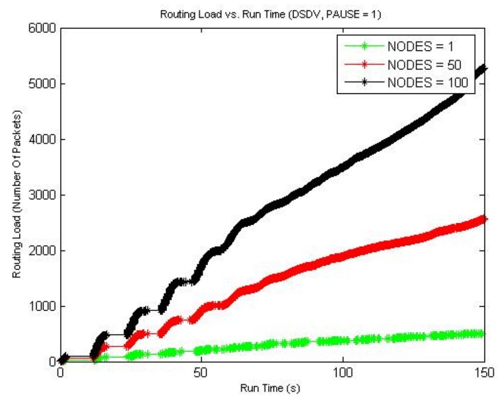


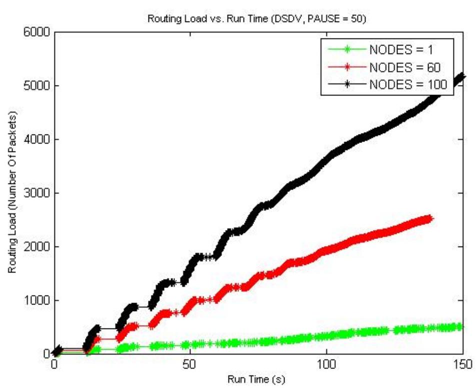


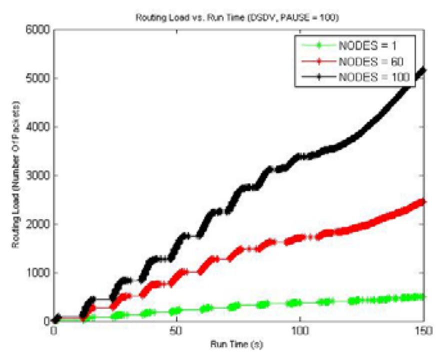

Received Load vs. Run Time (DSDV, PAUSE = 1)
NODES = 1
NODES = 50
NODES = 100
Received Load (Number Of Packets)
Run Time (s)


Received Load vs. Run Time (DSDV, PAUSE = 50)
NODES = 1
NODES = 60
NODES = 100
Received Load (Number Of Packets)
Run Time (s)

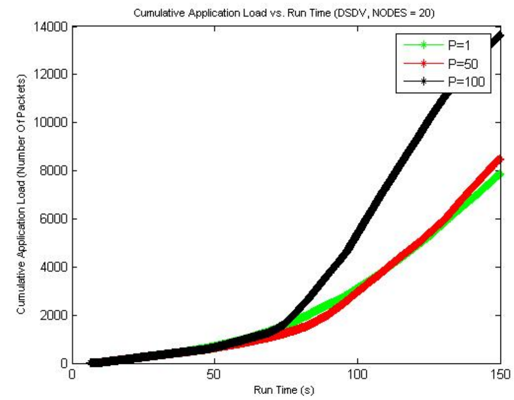

Received Load vs. Run Time (DSDV, PAUSE = 100)
NODES = 1
NODES = 60
NODES = 100
Received Load (Number Of Packets)
Run Time (s)

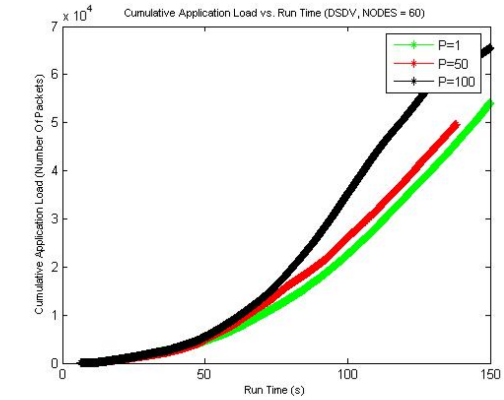

Routing Load vs. Run Time (DSDV, PAUSE = 1)
NODES = 1
NODES = 50
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

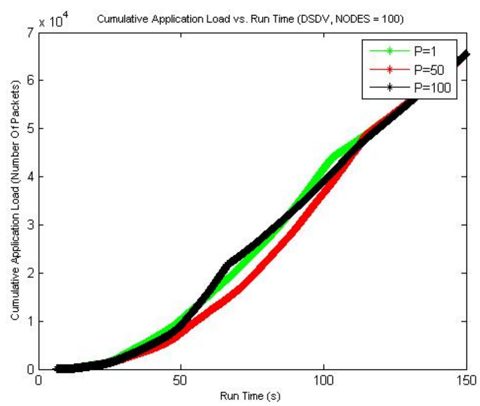

Routing Load vs. Run Time (DSDV, PAUSE = 50)
NODES = 1
NODES = 60
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

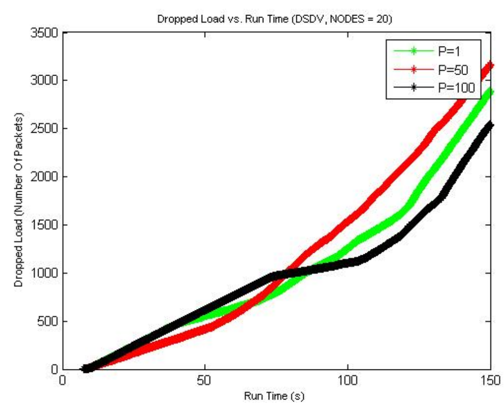

Routing Load vs. Run Time (DSDV, PAUSE = 100)
NODES = 1
NODES = 60
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

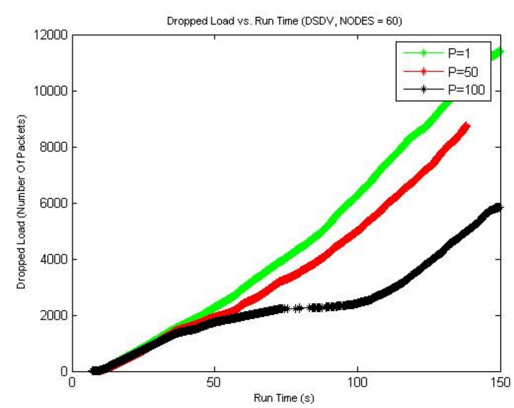
Appendix H – Graphs of DSDV Analysis (Varying Pause Time)

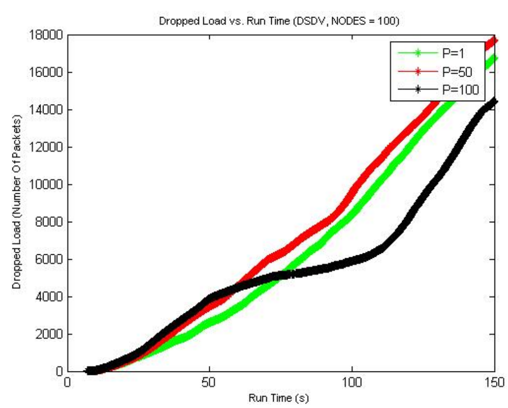


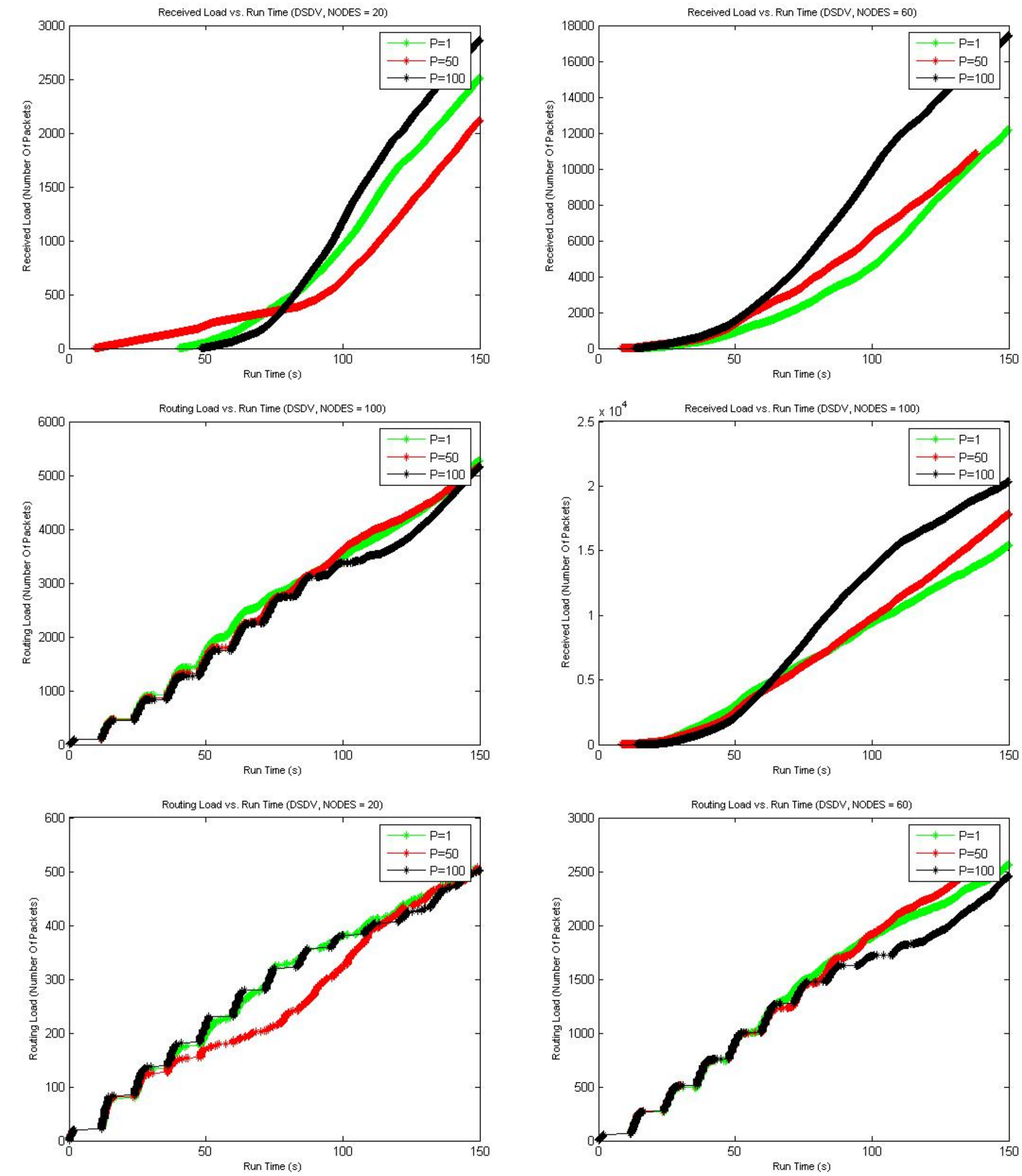




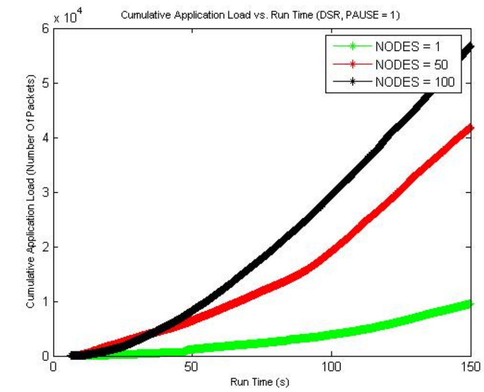


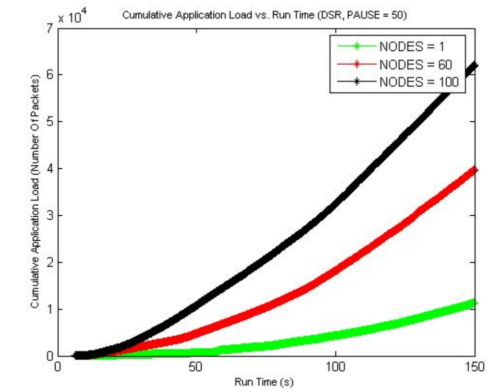


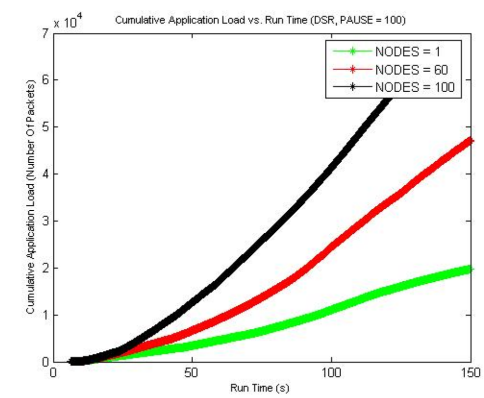


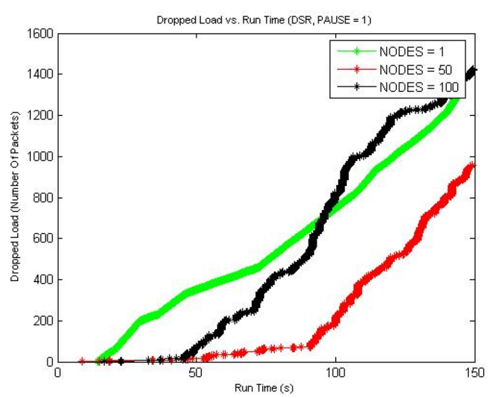

Received Load vs. Run Time (DSDV, NODES = 20)
Received Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100
Received Load vs. Run Time (DSDV, NODES = 60)
Received Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100
Routing Load vs. Run Time (DSDV, NODES = 100)
Routing Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100
Received Load vs. Run Time (DSDV, NODES = 100)
x 10^4
Received Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100
Routing Load vs. Run Time (DSDV, NODES = 20)
Routing Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100
Routing Load vs. Run Time (DSDV, NODES = 60)
Routing Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100

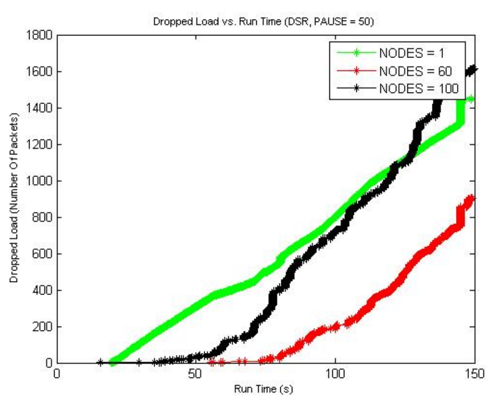
Appendix I – Graphs of DSR Analysis (Varying Total Number of Nodes on Network)

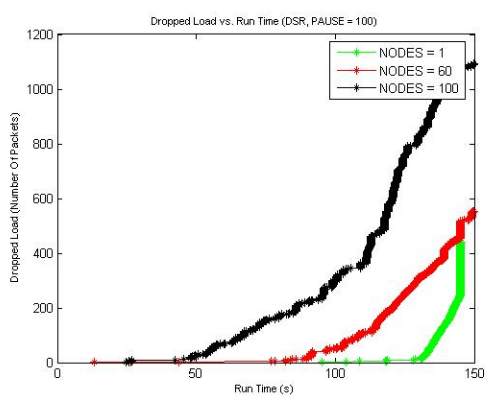


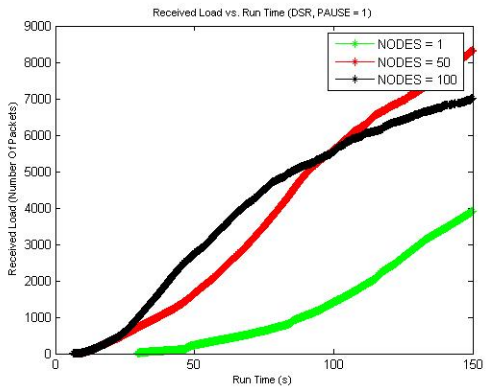


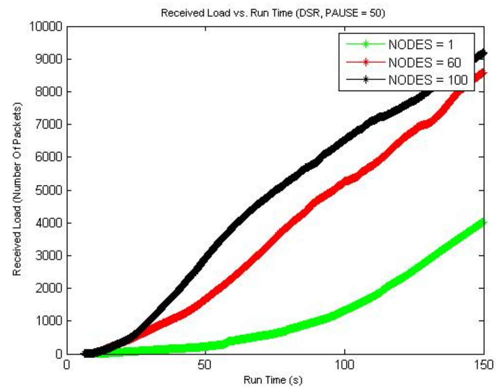


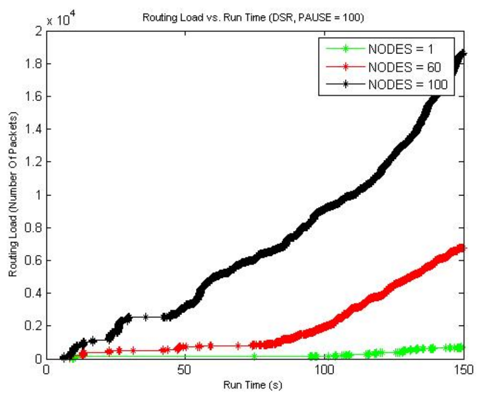


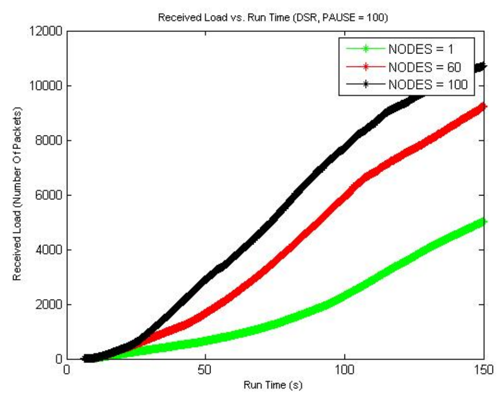


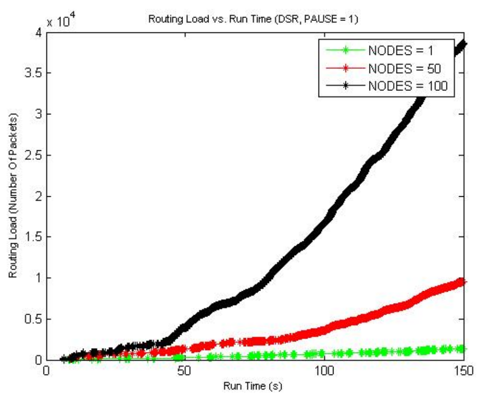


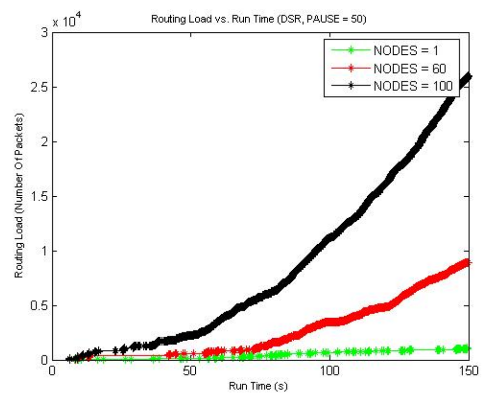

Received Load vs. Run Time (DSR, PAUSE = 1)
NODES = 1
NODES = 50
NODES = 100
Received Load (Number Of Packets)
Run Time (s)


Received Load vs. Run Time (DSR, PAUSE = 50)
NODES = 1
NODES = 60
NODES = 100
Received Load (Number Of Packets)
Run Time (s)

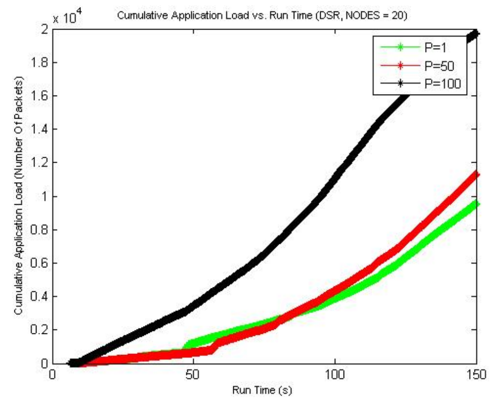

Routing Load vs. Run Time (DSR, PAUSE = 100)
x 10^4
NODES = 1
NODES = 60
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

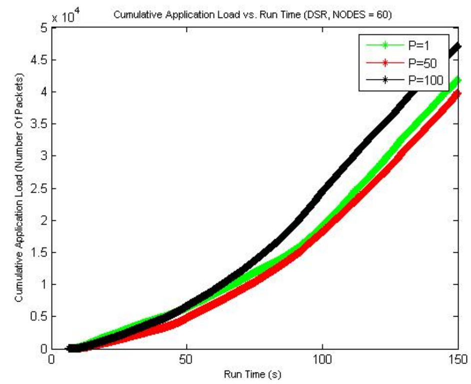

Received Load vs. Run Time (DSR, PAUSE = 100)
NODES = 1
NODES = 60
NODES = 100
Received Load (Number Of Packets)
Run Time (s)

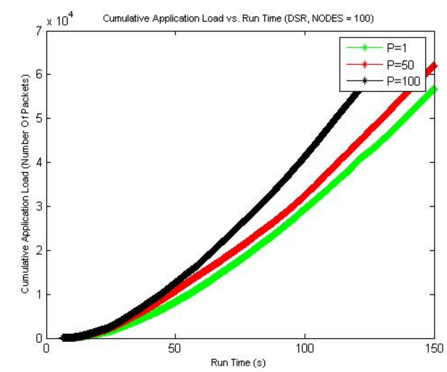

Routing Load vs. Run Time (DSR, PAUSE = 1)
x 10^4
NODES = 1
NODES = 50
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

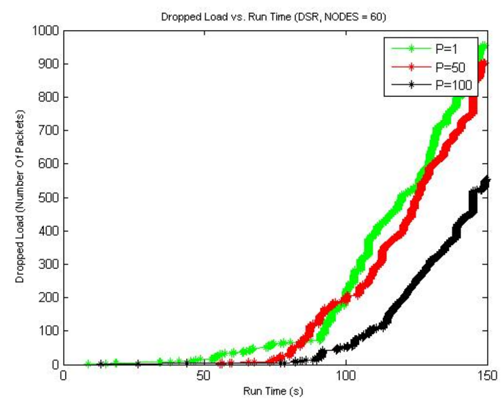

Routing Load vs. Run Time (DSR, PAUSE = 50)
x 10^4
NODES = 1
NODES = 60
NODES = 100
Routing Load (Number Of Packets)
Run Time (s)

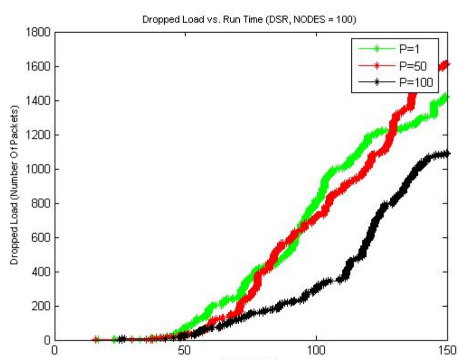
Appendix J – Graphs of DSR Analysis (Varying Pause Time)

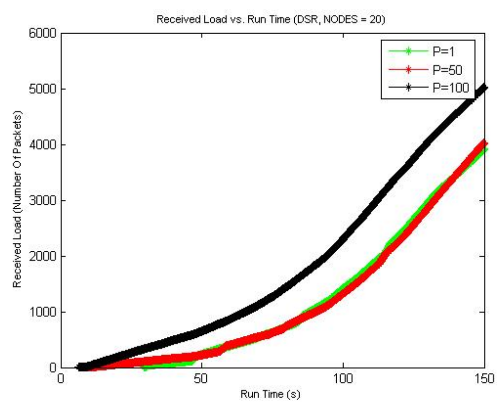


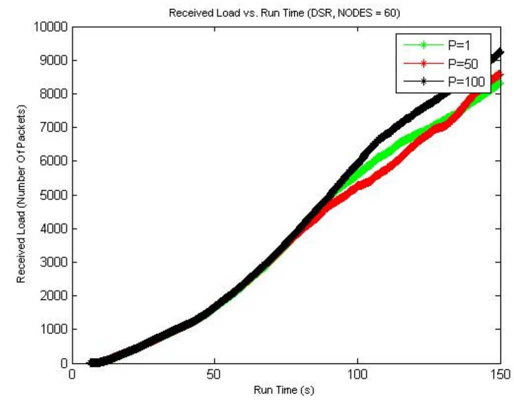


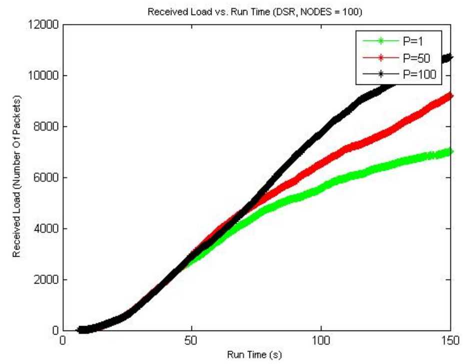


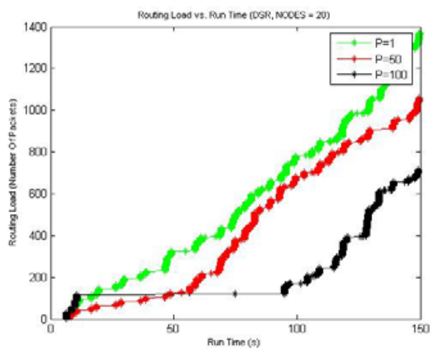


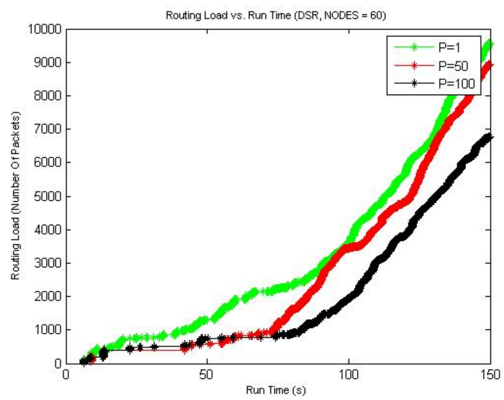


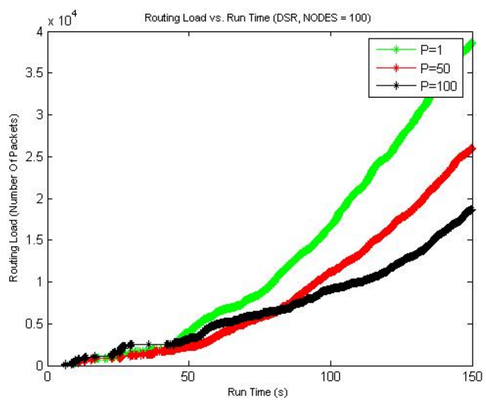



Received Load vs. Run Time (DSR, NODES = 60)
Received Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100


Received Load vs. Run Time (DSR, NODES = 100)
Received Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100


Routing Load vs. Run Time (DSR, NODES = 20)
Routing Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100


Routing Load vs. Run Time (DSR, NODES = 60)
Routing Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100


Routing Load vs. Run Time (DSR, NODES = 100)
Routing Load (Number Of Packets)
Run Time (s)
P=1
P=50
P=100

Appendix K – Code Listings