Performance Evaluation of AODV, DSDV & DSR Routing Protocol in Grid Environment

Nor Surayati Mohamad Usop
Faculty of Computer Sciences and Information Technology
Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak, Malaysia

Azizol Abdullah
Faculty of Computer Sciences and Information Technology
Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia

Ahmad Faisal Amri Abidin
Faculty of Informatics
Universiti Darul Iman Malaysia
21300 Kuala Terengganu, Terengganu, Malaysia

Abstract
Grid computing has recently migrated from traditional high performance and distributed computing to pervasive and utility computing based on the advanced capabilities of the wireless networks and the lightweight, thin devices. This has as result the emergence of a new computing paradigm which is the Mobile Grid. This paper presents the simulation results in order to choose the best routing protocol to give the highest performance when implement the routing protocols in the target mobile grid application. The simulations comparing three ad hoc routing protocols named DSDV, DSR and AODV. The simulations have shown that the conventional routing protocols like DSR have a dramatic decrease in performance when mobility is high. However the AODV and DSDV are perform very well when mobility is high.

Key words:
Mobile grid, ad hoc routing protocol, NS-2

1.INTRODUCTION

One of the most critical things for understanding and realizing *Mobile Grid computing* is to have a consistent and accurate definition, or at least determination of what a Mobile Grid is. There are many attempts for the accurate definition of the Grid. However the various approaches that have been made address in a high degree of accuracy the term Grid.

Mobile Grid, in relevance to both Grid and Mobile Computing, is a full inheritor of Grid with the additional feature of supporting mobile users and resources in a seamless, transparent, secure and efficient way [3]. It has the ability to deploy underlying ad-hoc networks and provide a self-configuring Grid system of mobile resources (hosts and users) connected by wireless links and forming arbitrary and unpredictable

Mobile Grid enables both the mobility of the users requesting access to a fixed Grid and the resources that are themselves part of the Grid. Both cases have their own limitations and constraints that should be handled. In the first case the devices of the mobile users act as interfaces to the Grid to monitor and manages the activities in 'anytime, anywhere' mode, while the Grid provides them with a high reliability, performance and cost-efficiency. Physical limitations of the mobile devices make necessary the adaptation of the services that Grid can provide to the users' mobile devices. In the second case of having mobile Grid resources, we should underline that the performances of current mobile devices are significantly increased. Laptops and PDAs can provide aggregated computational capability when gathered in hotspots, forming a Grid on site. This capability can advantage the usage of Grid applications even in places where this would be imaginary.

The objective of this paper is to study the Mobile Ad-hoc Network (MANET) routing protocol in grid environment. It is to make the comparison between AODV, DSDV and DSR routing protocols, using the performance metric such as packet delivery fraction, average-end to end delay and packet loss. This paper also carry out the analysis and discuss which protocol the is best between AODV, DSR and DSDV in mobility that implement in grid environment

This paper is organized as follows: Section 2 is described about the grid from definition until the characteristic of the grid. Others than that routing protocol for MANET such as AODV, DSR and DSDV explained deeply in order to show these protocol acts as a main role to realize this project. In Section 3 described a brief information of our methodology by giving a framework of simulation tool used and pointing out some techniques that chosen for network performance evaluation. Section 4 described the design and implementation with using network simulation tools for evaluating and validated the protocol used for the thesis. In Section 5 presents the simulation result and give analysis related to thesis objectives with simple explanation. Finally we present the conclusion together with thesis contribution and identifies some areas for future work in Section 6.

Manuscript received July 5, 2009
Manuscript revised July 20, 2009

2. LITERATURE REVIEW

One of the most critical things for understanding and realizing *Mobile Grid computing* is to have a consistent and accurate definition, or at least determination of what a Mobile Grid is. There are many attempts for the accurate definition of the Grid [5] [10]. However the various approaches that have been made address in a high degree of accuracy the term Grid. The Grid can be viewed as a distributed, high performance computing and data handling infrastructure, that incorporates geographically and organizationally dispersed, heterogeneous resources (computing systems, storage systems, instruments and other real-time data sources, human collaborators, communication systems) and provides common interfaces for all these resources, using standard, open, general-purpose protocols and interfaces. However, it is also the basis and the enabling technology for pervasive and utility computing due to the ability of being open, highly heterogeneous and scalable.

Mobile grid computing is applying the resources of many mobility computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. Mobile grid computing requires the use of software that can divide and farm out pieces of a program to as many as several thousand computers. Mobile grid computing can be thought of as distributed and large-scale cluster computing and as a form of network-distributed parallel processing. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration (in which case it is also sometimes known as a form of peer-to-peer computing). Mobile Grid computing is a form of wireless distributed computing whereby a "super and virtual computer" is composed of a cluster of networked, loosely-coupled computers, acting in concert to perform very large tasks. This technology has been applied to computationally-intensive scientific and to solve mathematical problems.

A mobile ad-hoc network (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links - the union of which form an arbitrary topology [1][9]. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless [2] topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Mobile ad hoc networks became a popular subject for research as laptops and 802.11/Wi-Fi wireless networking became widespread in the mid to late 1990s [6]. Many of the academic papers evaluate protocols and abilities assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other, and usually with nodes sending data at a constant rate. Different protocols are then evaluated based on the packet drop rate, the overhead introduced by the routing protocol, and other measures.

On-demand routing protocols build and maintain only needed routes to reduce routing overheads. Examples include Ad Hoc On-Demand Distance Vector (AODV)**,** Dynamic Source Routing (DSR), and Temporally Ordered Routing Algorithm (TOM) [6]. This is in contrast to proactive protocols Destination Sequenced Distance Vector (DSDV) that maintain routes between all node pairs all the time. In on-demand protocols, a route discovery process (typically via a network-wide flood) is initiated whenever a route is needed. Each node in on-demand routing does not need periodic route table update exchange and does not have a full topological view of the network. Network hosts maintain route table entries only to destinations that they communicate with.

In internetworking, routing the process of moving a packet of data from source to destination. Routing is usually performed by a dedicated device called a router. Routing is a key feature of the Internet because it enables messages to pass from one computer to another and eventually reach the target machine. Each intermediary computer performs routing by passing along the message to the next computer. Part of this process involves analyzing a *routing table* to determine the best path.

2.1 Ad hoc On Demand Distance Vector (AODV)

The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks [1] [7]. AODV is capable of both unicast and multicast routing [15]. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes [6].

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has

a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID [6]. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.
As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route.

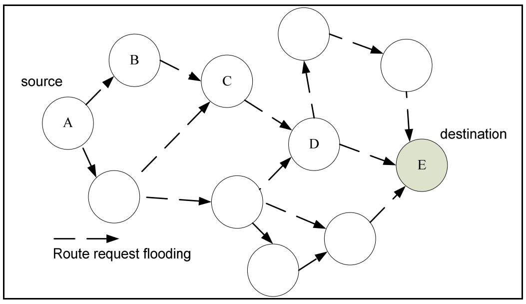

Figure 2.1. Route Request (RREQ) flooding

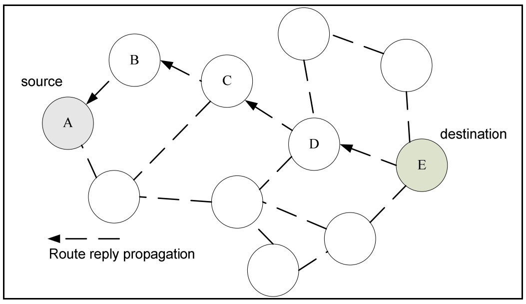

Figure 2.2. Route Reply (RREP) propagation

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

2.2 Dynamic Source Routing (DSR)

Dynamic Source Routing (DSR) is a routing protocol for wireless mesh networks [16]. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device [8]. Many successive refinements have been made to DSR, including DSRFLOW.
Determining source routes requires accumulating the address of each device between the source and destination during route discovery. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to route packets. To accomplish source routing, the routed packets contain the address of each device the packet will traverse. This may result in high overhead for long paths or large addresses, like IPv6 [14]. To avoid using source routing, DSR optionally defines a flow id option that allows packets to be forwarded on a hop-by-hop basis.
This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only 2 major phases which are Route Discovery and Route Maintenance [4]. Route Reply would only be generated if the message has reached the intended destination node (route record which is initially contained in Route Request would be inserted into the Route Reply).
To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the Route Reply message header (symmetric links). In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets are generated at a node. The erroneous hop will be removed from the node's route cache, all routes containing the hop are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.

2.3 Destination-Sequenced Distance-Vector Routing (DSDV)

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. It was developed by C. Perkins and P.Bhagwat in 1994 [21]. The main contribution of the algorithm was to solve the Routing Loop problem. Each entry in the routing table contains a sequence number, the sequence numbers are generally even if a link is present; else, an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number [18]. Routing information is distributed between nodes by

sending full dumps infrequently and smaller incremental updates more frequently.
DSDV was one of the early algorithms available. It is quite suitable for creating ad hoc networks with small number of nodes. Since no formal specification of this algorithm is present there is no commercial implementation of this algorithm. Many improved forms of this algorithm have been suggested.
DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; thus, DSDV is not suitable for highly dynamic networks. (As in all distance-vector protocols, this does not perturb traffic in regions of the network that are not concerned by the topology change.)

3. METHODOLOGY

This section starts by giving a framework and overview of some techniques that are chosen for network performance evaluation. There are three techniques for performance evaluation which are analytical modeling, simulation and measurement. The reason for choosing simulation as a technique for performance evaluation in this thesis is explained as well in Section.

3.1 Selection Techniques for Network Performance Evaluation

Performance is a key criterion in the design, procurement, and use of computer systems. Computer systems professionals such as computer systems engineers, scientist, analysts and users need the basic knowledge of performance evaluation techniques as the goal to get the highest performance for a given cost. There are three techniques for performance evaluation, which are analytical modeling, simulation and measurement.
Simulation had being chosen because it is the most suitable technique to get more details that can be incorporate and less assumption is required compared to analytical modeling. Accuracy, times available for evaluation and cost allocated for the thesis are also another reason why simulation is chose. By using simulation, researchers should be allowed to study a system in well-known conditions, repeatability if necessary in order to understand events.

3.2 Computer Network Simulator Tools

There are many simulators such as Network Simulator 2 (NS-2), OPNET Modeler, GloMoSim, OMNeT++ and etc. In this project we choices Network Simulation Tool (NS-2).
NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTcl. NS-2 is primarily useful for simulating local and wide area networks. Although NS is fairly easy to use once you get to know the simulator, it is quite difficult for a first time user, because there are few user-friendly manuals. Even though there is a lot of documentation written by the developers which has in depth explanation of the simulator, it is written with the depth of a skilled NS user. The purpose of this project is to give a new user some basic idea of how the simulator works, how to setup simulation networks, where to look for further information about network components in simulator codes, how to create new network components, etc., mainly by giving simple examples and brief explanations based on our experiences. Although all the usage of the simulator or possible network simulation setups may not be covered in this project, the project should help a new user to get started quickly.

NS-2 interprets the simulation scripts written in OTcl. A user has to set the different components (e.g. event scheduler objects, network components libraries and setupmodule libraries) up in the simulation environment. The user writes his simulation as a OTcl script, plumbs the network components together to the complete simulation.

3.3 Why we chosen NS-2?

NS-2 is chosen as the simulation tool among the others simulation tools because NS-2 supports networking research and education. Ns-2 is suitable for designing new protocols, comparing different protocols and traffic evaluations.
NS-2 is developed as a collaborative environment. It is distributed freely and open source. A large amount of institutes and people in development and research use, maintain and develop NS-2. This increase the confidence in it. Versions are available for FreeBDS, Linux, Solaris, Windows, Mac OS X. NS-2 also provides substantial support for simulation of TCP, UDP, routing and multicast protocols over wired and wireless networks.

3.4 Simulation Model

We run the simulation in Network Simulator (NS-2) accepts as input a *scenario file* that describes the exact motion of each node and the exact packets originated by each node, together with the exact time at which each change in motion or packet origination is to occur. The detailed trace file created by each run is stored to disk, and analyzed using a variety of scripts, particularly one called *file *.tr* that counts the number of packets successfully delivered and the length of the paths taken by the packets, as well as additional information about the internal functioning of each scripts executed. This data is further analyzed with AWK file and Microsoft Excel to produce the graphs.

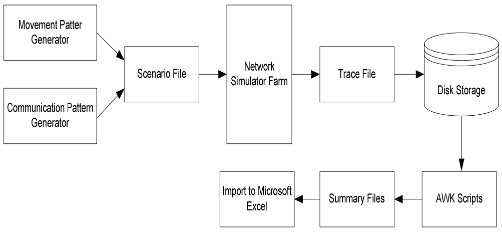

Figure 3.1 Overview of simulation model

The simulation models are built using the Network Simulator tool (NS-2) version 2.32 and it is run under a nominal bit rate of 2 Mbps. The experiments use a fixed number of packet sizes (512-bytes) with a changing of pause times (0s, 100s, 200s, 300s, 400s, 500s, 600s, 700s, 800s, and 900s). The simulation used fixed number of source nodes which are 50 nodes (*50 sources*) and a packet rate of 4 packets per seconds.

The mobility model used is a random waypoint model [27] in a rectangular field. The field configurations used is 1500m X 500m with 50 nodes and the stations are assumed to be evenly distributed in the area. Here, each packet starts its journey from a random location to a random destination with a randomly chosen speed. Once the destination is reached, another random destination is targeted after a pause. The pause time, which affects the relative speeds of the mobiles, is varied. Simulations are run for 900 simulated seconds. Identical mobility and traffic scenarios are used across protocols to gather fair results.

3.5 The Simulation Assumptions

The following assumptions are considered when building the Tcl script:

1. For simplicity, all flows in the system are assumed to have the same type of traffic source. Each sender has constant bit rate (CBR) traffic with the rate of data rate/number of stations packet per second;
2. The source node is fixed to 50 nodes with maximum connection is 40 nodes (to show a density condition); and
3. The implementation of grid and integrate between grid and routing protocols.

3.6 Performance Metrics

The project focuses on 3 performance metrics which are quantitatively measured. The performance metrics are important to measure the performance and activities that are running in NS-2 simulation. The performance metrics are:

Packet delivery fractions (PDF) — also know as the ratio of the data packets delivered to the destinations to those generated by the CBR sources. The PDF shows how successful a protocol performs delivering packets from source to destination. The higher for the value give use the better results. This metric characterizes both the completeness and correctness of the routing protocol also reliability of routing protocol by giving its effectiveness.

$$PktDelivery\% = \frac{\sum_{1}^{n} CBRrecv}{\sum_{1}^{n} CBRrecv} \times 100 ......Equation1$$

Figure 3.2. Formula for packet delivery fraction

Average end-to-end delay of data packets — There are possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times. The thesis use Average end-to-end delay. Average end-to-end delay is an average end-to-end delay of data packets. It also caused by queuing for transmission at the node and buffering data for detouring. Once the time difference between every CBR packet sent and received was recorded, dividing the total time difference over the total number of CBR packets received gave the average end-to-end delay for the received packets. This metric describes the packet delivery time: the lower the end-to-end delay the better the application performance.

$$Avg\_End-to-End\_Delay = \frac{\sum_{1}^{n}(CBRsentTime - CBRrecvTime\_)}{\sum_{1}^{n} CBRrec} ......Equation2$$

Figure 3.3. Formula for end-to-end delay performance metric

Data Packet Loss (Packet Loss) — Mobility-related packet loss may occur at both the network layer and the MAC layer. In the thesis packet loss concentrate for network layer. When a packet arrives at the network layer. The routing protocol forwards the packet if a valid route to the destination is known. Otherwise, the packet is buffered until a route is available. A packet is dropped in two cases: the buffer is full when the packet needs to be buffered and the time that the packet has been buffered exceeds the limit.

$$PLose = DataAgtSent - DataAgtRec .....Equation\ 3$$

AGT– agent trace (use in new trace file format)

Figure 3.4. Formula for packet loss performance metric

4. SIMULATION ENVIRONMENT

This section described how to design and implement the comparison between the AODV, DSR and DSDV routing protocol using the average end to end delay, packet loss and packet delivery fraction performance metrics [11]. The value of simulation in studies of protocols is that it allows near perfect experimental control: experiments can be designed at will and then rerun while varying an experimental variable and holding all other variables constant [13]. With simulation, it is also possible to test the behavior of networks with more nodes than physical equipment is available for, or networks with equipment that does not even exist yet.

The drawback of simulation is that inherently runs the risk of over simplification. It is not possible to exactly replicate the entire world inside a computer model, so when creating a simulation some factors must be statistically or otherwise approximated. The failure to properly capture the behavior of first-order factors can lead to dramatically incorrect results.

4.1 The TCL Script

In the TCL script had defined the scene of communication in which ours agent of Density has operated. In the part it begins them come defined the parameters that define the characteristics of the topology and those of the communication model.

4.2 CMU's wireless extensions to NS-2

Carnegie Mellon University (CMU's) wireless extension to NS-2 (incorporated in the current release NS-2.1b9a) provides the implementation of the routing protocols.

NAM is the basic visualization tool used for NS-2 simulations. However, it doesn't support the ad-hoc simulations. Ad-hockey is a Perl/Tk program that supports the visualization of ad-hoc simulations. Unfortunately, I couldn't get the Ad-hockey visualizer tool to work because of its compatibility issues with the new versions of Perl/Tk module.

4.2.1 Installing Density Agent in NS-2 Simulator

In this section described the way to add an agent into the NS-2 environment. Used GAF model as a comparison to install a new agent in NS-2. There are 5 main file to setup (lib/ns_packet.tcl, common/packet.h, lib/ns-default.tcl, lib/ns-mobilenode.tcl and cmu-trace.cc).

4.3 Generating traffic and mobility models

Continuous bit rate (CBR) traffic sources are used. The source-destination pairs are spread randomly over the network. Only 512-byte data packets are used. The number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network. The mobility model uses the random waypoint model in a rectangular field. The field configurations used is: 1500 m x 500 m field with 50 nodes. Here, each packet starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed in 20m/s). Once the destination is reached, another random destination is targeted after a pause. The pause time, which affects the relative speeds of the mobiles, is varied. Simulations run for 900 simulated seconds. Identical mobility and traffic models generated only once to gather fair results for this thesis. Its related the scenario that relevant to this thesis.

5. RESULT & DISCUSSION

Details of analysis are focusing on packet-delivery fraction, packet loss, average end to end delay and send/received ratio in term mobility.

5.1 Packet Delivery Fraction (PDF) Result and Analysis

PDF is the ratio between the number of packets originated by the application layer sources and the number of packets received by the sinks at the final destination. It will describe the loss rate that will be seen by the transport protocols, which in turn affects the maximum throughput that the network can support.

This simulation chooses 0, 100, 200, 300, 400, 500, 600, 700, 800 and 900seconds pause time. This simulation generates 50 nodes. Figure below shown at pause time 0 seconds (high mobility) environment, **AODV** outperforms DSDV and DSR in high mobility environment, topology change rapidly and AODV can adapt to the changes quickly since it only maintain one route that is actively used. **DSDV** deliver less data packet compare to AODV because in rapid change topology it is not as adaptive to route changes in updating its table. **DSR** does not have mechanism in knowing which route in the cache is stale, data packet is forwarded to broken link.

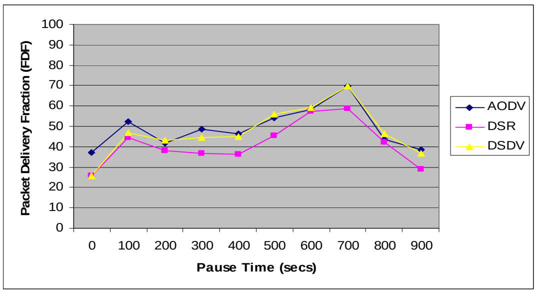

Figure 5.1. Packet Delivery Fraction (PDF) vs Pause Time (simulation time)

5.2 Average End to End Delay Result and Analysis

The delay is affected by high rate of CBR packets as well. The buffers become full much quicker, so the packets have to stay in the buffers a much longer period of time before they are sent. This can be seen at the DSR routing protocol when it was reach around 2400 packets at the 0 mobility.
Refer to the graph below, at the 600 pause time the delay is very high for all protocols. This is because at this pause time happen extremely high data rate and the low mobility. The low mobility will mean that already found routes are valid for much longer time period. This means that found routes can be used for more packets. From the graph above would be able to conclude that **AODV,** since routes are established on demand and destination sequence numbers are used to find the latest route to the destination. The connection setup delay is less. **DSDV,** whenever the topology of the network changes, a new sequence number is necessary before the network re-converges. **DSR** does not have mechanism in knowing which route in the cache is stale, data packet is forwarded to broken link.

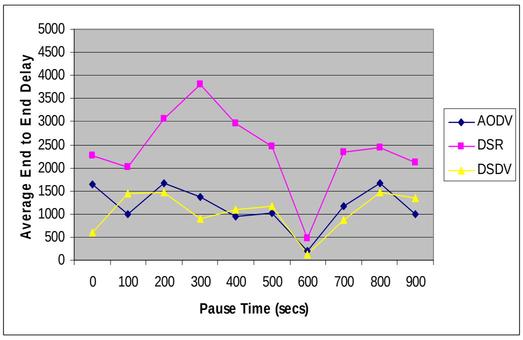

Figure 5.2. Average End to End Delay vs Pause Time (simulation time)

5.3 Packet Loss Result and Analysis

Refer to the graph below, show not much packet loss on AODV side. This is because when a link fails, a routing error is passed back to a transmitting node and the process repeats. Meanwhile for DSDV, this routing protocol shows it is as good as AODV if packet loss be as indicator. This can be prove by the characteristics of DSDV which information on new Routes, broken Links, metric change is immediately propagated to neighbors. For DSR, show the packet loss higher than DSDV and AODV because the route maintenance mechanism does not locally repair a broken link.

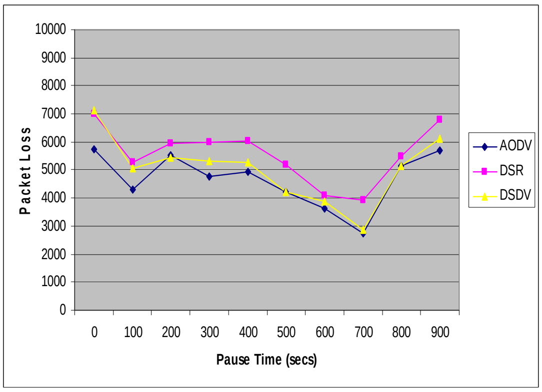

Figure 5.3. Packet Loss vs Pause Time (simulation time)

6. CONCLUSION

This paper does the realistic comparison of three routing protocols DSDV, AODV and DSR. The significant observation is, simulation results agree with expected results based on theoretical analysis. As expected, reactive routing protocol AODV performance is the best considering its ability to maintain connection by periodic exchange of information, which is required for TCP, based traffic. AODV performs predictably. Delivered virtually all packets at low node mobility, and failing to converge as node mobility increases. Meanwhile DSR was very good at all mobility rates and movement speeds and DSDV performs almost as well as DSR, but still requires the transmission of many routing overhead packets. At higher rates of node mobility it's actually more expensive than DSR.

For the future work, this area will investigate not only the comparison between AODV, DSDV and DSR routing protocols in grid but more on the vast areas. As we know, routing protocol in grid environment is a rather hot concept in computer communications. Maybe for the future we would be able to focus more on security issue. As we know, routing protocols are prime targets for impersonation attacks. Due to the mobile grid environment is formed without centralized control, security must be handled in a distributed fashion. This will probably mean that IPSec authentication headers will be deployed, as well as the necessary key management to distribute keys to the members of the mobile grid.

REFERENCES

[1] Azizol Abdullah, Norlida Ramly, Abdullah Muhammed, Mohd Noor Derahman: Performance Comparison Study of Routing Protocols for Mobile Grid Environment, pp 82-88, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.2, February 2008

[2] C. S. R. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architecture and Protocols, ch. Routing Protocols for Ad Hoc Wireless Networks, pp. 299{364. Prentice Hall Communications Engineering and Emerging Technologies Series, New Jersey: PrenticeHall Professional Technical Reference, 2004.

[3] Irving,M., Taylor.G., and Hobson.P. (2004). Plug in to Grid Computing. IEEE Power & Energy Magazine, March/April, pp.40-44.

[4] Staub.T., (2004). Ad-hoc and Hybrid Networks: Performance Comparison of MANET Routing Protocols in Ad-hoc and Hybrid Networks. Institute of Computer Science and Applied Mathematics, University of Berne, Switzerland, pp.1-38.

[5] Chao,C-M., Sheu,J-P.,and Hu,C-T. (2003). Energy-Conserving Grid Routing Protocol in Mobile Ad Hoc Networks. IEEE Proceedings of the 2003 International Conference on Parallel Processing (ICPP'03).

[6] Z.Alexander, Performance Evaluation of AODV Routing Protocol: Real-Life Measurements, SCC,June 2003

[7] Altman E, and Jimenez T., (2003). NS Simulator for Beginners. Lecture notes. Univ.de Los Andes, Merida, Venezuela and ESSI.Sophia-Antipolis, France.

[8] D. Johnson, Dynamic Source Routing for Mobile Ad Hoc Networks, IEFT MANET Draft, April 2003

[9] S. Nesargi, R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Networks", 21st Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM), Jun. 2002.

[10] Wen,Y., Mobile Grid. (2002). Department of Computer Science, University of California, Santa Barbara,pp.1-7.

[11] S. Das, C. Perkins, and E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In INFOCOM'2000 (1), pages 3–12, 2000.

[12] K. Fall. ns notes and documentation. The VINT Project, 2000.

[13] D.Johnson. Validation of wireless and mobile network models and simulation. In DARPA/NIST Network Simulation Validation Workshop, Fairfax, Virginia, USA, May 1999.

[14] D. Johnson, D. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. In Ad Hoc Networking, C. Perkins (ed), Chapter 5, pages 139–172, 2001.

[15] Charles E. Perkins and Elizabeth M. Royer, "Ad Hoc On-Demand Distance Vector Routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Syslems and Applications, Feb. 1999, pp. 90-100.

[16] David B. Johnson and David A. Maltz, ''Dynamic source routing in adhoc wireless networks," in Mobile computing, T. lmielinski and H. Kmh, Eds, Kluwer Academic, 1996, ch.5 .

[17] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[18] Perkins,C.E., and Bhagwat.P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. ACM, pp.234 – 244.

[19] The Network Simulator NS-2 homepage, http://www.isi.edu/nsnam/ns/

[20] The Network Simulator NS-2 tutorial homepage, http://www.isi.edu/nsnam/ns/tutorial/index.html

Nor Surayati Mohamad Usop obtained her Master of Science Computer from Faculty of Computer Science and Information Technology, Universiti Putra Malaysia in 2009. Currently, she is a Lecturer at Department of Communication Technologies, Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak.

Azizol Abdullah obtained his Master of Science in Engineering (Telematics) from the University of Sheffield, UK in 1996. He is a Senior Lecturer at Department Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Currently, he is pursuing his Ph.D at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. His main research areas include grid computing, peer-to-peer computing, wireless and mobile computing, computer networks, Computer Support Collaborative Workgroup (CSCW) and Computer Support Collaborative Learning (CSCL).

Ahmad Faisal Amri Abidin obtained his Master of Science Computer from Faculty of Computer Science and Information Technology, Universiti Putra Malaysia in 2008. Currently, he is a Lecturer at Department of Computer Sciences, Faculty of Informatics, Universiti Darul Iman Malaysia. His main research areas include computer security, mobile computing and computer networks.