Performance Comparison of Wireless Ad Hoc Routing Protocols

Ad Hoc On-Demand Distance Vector Routing, Dynamic Source Routing and Destination-Sequenced Distance-Vector

A.F.A. Abidin[1], N.S.M. Usop[2] and M.K. Yusof[3]
Faculty of Informatics, Universiti Sultan Zainal Abidin, Kuala Terengganu, Malaysia.

Abstract- An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. This paper compared the performance of two prominent on-demand reactive routing protocols for mobile ad hoc networks: Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector Routing (AODV), along with the traditional Proactive Destination-Sequenced Distance-Vector (DSDV) protocol. A simulation model with Media Access Control (MAC) and physical layer models are used to study interlayer interactions and its performance implications. The on-demand protocols, AODV and DSR perform better than the table-driven DSDV protocol. Although DSR and AODV share similar on-demand behavior, the differences in the protocol mechanics can lead to significant performance differentials. The performance differentials are analyzed using varying network load, mobility and network size.

Key words: ***AODV, DSR, DSDV and ad hoc network.***

I. INTRODUCTION

Wireless networking is an emerging technology that allows users to access information and services electronically, regardless of their geographic position. Wireless networks can be classified in infrastructure and infrastructureless networks.

A. Infrastructure networks

Infrastructure network consists of a network with fixed and wired gateways. A mobile host communicates with a bridge in the network (called base station) within its communication radius. The mobile unit can move geographically while it is communicating. When it goes out of range of one base station, it connects with new base station and starts communicating through it. This is called handoff and in this approach the base stations are fixed.

B. Infrastructureless (Ad hoc) networks

In ad hoc networks all nodes are mobile and can be connected dynamically in an arbitrary manner. All nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. Ad hoc networks are very useful in emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information and data acquisition operations in inhospitable terrain.

These ad hoc routing protocols can be divided into two categories which are table-driven routing protocols and on-demand routing protocols. In table driven routing protocols, the routes are consistent and up-to-date the routing information to all nodes. The routes in on-demand routing protocols are created when required. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. In recent years, a variety of new routing protocols targeted specifically at this environment have been developed. There are three multi-hop wireless ad hoc network routing protocols that cover a range of design choices:

1. Destination-Sequenced Distance-Vector (DSDV)
2. Dynamic Source Routing (DSR)
3. Ad Hoc On-Demand Distance Vector Routing (AODV)

II. MOTIVATION

The main objective of this paper is to compare the performance of on-demand reactive routing protocols for mobile ad hoc networks along with the traditional other proactive protocol. A simulation model with MAC and physical layer models is used to study interlayer interactions and its performance implications.

III. OVERVIEW

A. Destination-Sequenced Distance-Vector (DSDV)

Algorithm is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements. Every mobile station maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node [1]. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven.

The routing table updates can be sent in two ways such as a "full dump" or an incremental update. A full dump sends the full routing table to the neighbors and could span many packets whereas in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence number has changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent.

B. Dynamic Source Routing (DSR)

The key distinguishing feature of DSR is the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process to dynamically determine such a route [10]. Route discovery works by flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path traversed across the network. The RREP routes back to the source by traversing this path backward. The route carried back by the RREP packet is cached at the source for future use.

If any link on a source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed. DSR makes very aggressive use of source routing and route caching. No special mechanism to detect routing loops is needed. Also, any forwarding node caches the source route in a packet it forwards for possible future use.

C. Ad Hoc On-Demand Distance Vector Routing (AODV)

AODV shares DSR’s on-demand characteristics in that it also discovers routes on an as needed basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination [2]. Without source routing, AODV relies on routing table entries to propagate an RREP back to the source and, subsequently, to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine freshness of routing information and to prevent routing loops. All routing packets carry these sequence numbers.

An important feature of AODV is the maintenance of timer-based states in each node, regarding utilization of individual routing table entries. A routing table entry is expired if not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes which use that entry to route data packets. These nodes are notified with RERR packets when the next-hop link breaks. Each predecessor node, in turn, forwards the RERR to its own set of predecessors, thus effectively erasing all routes using the broken link. In contrast to DSR, RERR packets in AODV are intended to inform all sources using a link when a failure occurs. Route error propagation in AODV can be visualized conceptually as a tree whose root is the node at the point of failure and all sources using the failed link as the leaves.

IV. METHODOLOGY

A. Simulation Model

A detailed simulation model based on Network Simulation-2 (NS-2) is used in the evaluation. In a previous paper the Monarch research group at Carnegie-Mellon University developed support for simulating multihop wireless networks complete with physical, data link, and medium access control (MAC) layer models on NS-2 [1][8]. The Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer protocol. An unslotted carrier sense multiple access (CSMA) technique with collision avoidance (CSMA/CA) is used to transmit the data packets. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN. WaveLAN is modeled as a shared-media radio with a nominal bit rate of 2 mb/s and a nominal radio range of 250 m [9].

The protocols maintain a send buffer of 64 packets. It contains all data packets waiting for a route, such as packets for which route discovery has started, but no reply has arrived yet. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30s. All packets (both data and routing) sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has a maximum size of 50 packets and is maintained as a priority queue with two priori-ties each served in FIFO order. Routing packets get higher priority than data packets.

B. The Traffic and Mobility Models

The Constant Bit Rate (CBR) service is used for connections that transport traffic at a consistent bit rate, where there is an inherent reliance on time synchronization between the traffic source and destination. 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: 500 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 between 0–20 m/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 are run for 100 simulated seconds. Identical mobility and traffic scenarios are used across protocols to gather fair results.

C. Performance Metrics

Three important performance metrics are evaluated:

i. Packet delivery fraction — The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

ii. Average end-to-end delay of data packets — This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, propagation and transfer times.

iii. Normalized routing load — The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission.

The first two metrics are the most important for best-effort traffic. The routing load metric evaluates the efficiency of the routing protocol. Note, however, that these metrics are not completely independent. For example, lower packet delivery fraction means that the delay metric is evaluated with fewer samples. In the conventional wisdom, the longer the path lengths, the higher the probability of a packet drops. Thus, with a lower delivery fraction, samples are usually biased in favor of smaller path lengths and thus have less delay.

V. DESIGN AND IMPLEMENTATION

A. Generating traffic and mobility models

1) Traffic models

Random traffic connections of Transmission Control Protocol (TCP) and Constant Bit Rate (CBR) can be setup between mobile nodes using a traffic-scenario generator script. This traffic generator script is available under ~ns/indep-utils/cmu-scen-gen and called as cbrgen.tcl [1]. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. So the command line looks like the following:

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

For the simulations carried out, traffic models were generated for 50 nodes with CBR traffic sources, with maximum connections of 10,20,30 at a rate of 8kbps [1].

2) Mobility models

The node-movement generator is available under ~ns/indep-utils/cmu-scen-gen/setdest directory and consists of setdest{.cc,.h} and Makefile. The command would looks like:

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

Mobility models were created for the simulations using 50 nodes, with pause times of 0,10,20,40,100s, maximum speed of 20m/s, topology boundary of 500x500 and simulation time of 100s.

a) Parsing the Simulation trace files

After each simulation, trace files recording the traffic and node movements are generated. These files need to be parsed in order to extract the information needed to measure the performance metrics. The new trace format was used for parsing.

The new trace format looks like:

```
s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx
5.00 -Ny 2.00 -Nz 0.00 –Ne -1.000000 -
Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0
–Ii 20 -Is 0.255 -Id -1.255 –It
```

Here, the packet was sent (s) at time (t) 0.267662078 sec, from source node (Hs) 0 to destination node (Hd) 1. The source node id (Ni) is 0, it's x-co-ordinate (Nx) is 5.00, it's y-co-ordinate (Ny) is 2.00, it's z-co-ordinate (Nz) is 0.00, its energy level (Ne) is 1.000000, the trace level (Nl) is RTR and the node event (Nw) is blank. The MAC level information is given by duration (Ma) 0, destination Ethernet address (Md) 0, the source Ethernet address (Ms) is 0 and Ethernet type (Mt) is 0.

Evaluating Packet delivery fraction (pdf):

Calculate the number of "sent packets" that have the trace form:

```
/^s *- Nl AGT.*-Is (\d{1,3})\.\d{1,3} -Id
(\d{1,3})\.\d{1,3}.*-It cbr.*-Ii (\d{1,6})/
```

AGT => Agent Level Trace

Calculate the number of "received packets" of the trace form:

```
/^r -t (\d{1,3}\.\d{9}).*-Nl AGT.*-Is (\d{1,3})\.\d{1,3} -Id
(\d{1,3})\.\d{1,3}.*-It cbr.*-Ii (\d{1,6})/
```

packet delivery fraction (pdf %) = (received packets/ sent packets) *100

b) Evaluating Average End-to-End packet delivery time

For each packet with id (Ii) of trace level (AGT) and type (cbr), calculate the send(s) time (t) and the receive (r) time (t) and average it.

c) Evaluating Normalized routing load

Calculate the routing packet sent:

```
/^[s|f].*-Nl RTR.*-It (?:AODV|DSR|message) -Il (\d{1,4})/
```

f=> forward

RTR=> Routing Trace Level

Normalized routing load = (routing packets sent) / receives.

VI. SIMULATION RESULTS AND DISCUSSION

For all the simulations, the same movement models were used, the number of traffic sources was fixed at 10,20 and 30 sources. The maximum speed of the nodes was set to 20m/s and the pause time was varied as 0s, 10s, 20s, 40s and 100s.

Figure 1 highlights the relative performance of the three routing protocols. All of the protocols deliver a greater percentage of the originated data packets when there is little node mobility (large pause time), converging to 100% delivery when there is no node motion.

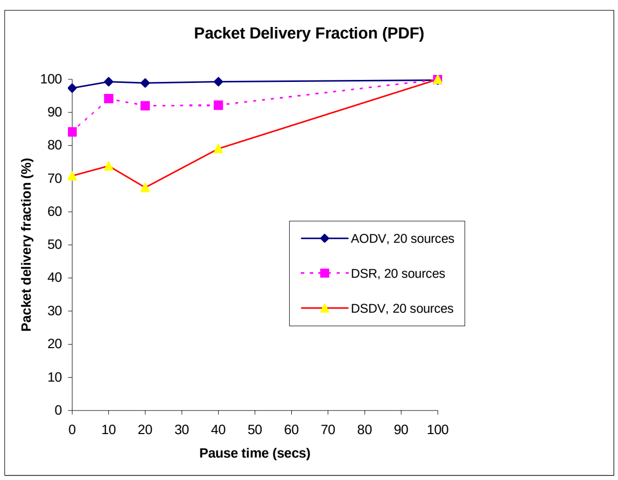

Figure 1: The relative performance of AODV, DSR and DSDV

A. *Performance comparison of the protocols*

1) Packet delivery Comparison

The on-demand protocols, DSR and AODV performed particularly well, delivering over 85% of the data packets regardless of mobility rate. AODV can adapt to the changes quickly in mobility environment since it only maintain one route that is actively used. DSDV deliver less data packet because in rapid change topology it is not as adaptive to route changes in updating its table

2) Average End to End Packet delivery

The average end to end delay of packet delivery was higher in DSDV as compared to both DSR and AODV. In summary, both the on-demand routing protocols, AODV and DSR outperform the table-driven routing protocol, DSDV.

Since both AODV and DSR did better, an attempt was made to evaluate the performance difference between the two by varying the mobility pattern and number of traffic sources.

3) Packet delivery Comparison

The packet delivery fractions for DSR and AODV are similar with 10 sources (figure 2). However, with 20 and 30 sources, AODV outperforms DSR by about 15 percent (figure 3 and 4) at lower pause times (higher mobility).

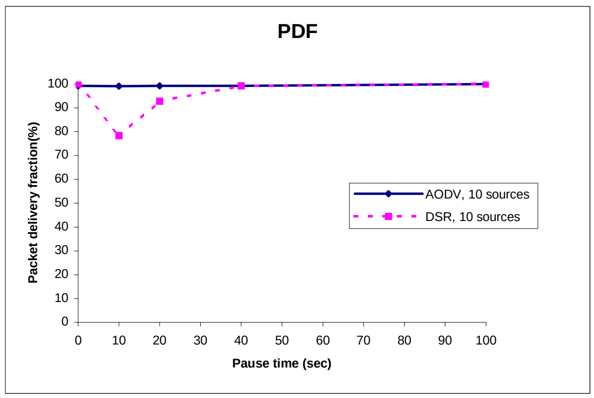

Figure 2: The packet delivery fractions for DSR and AODV

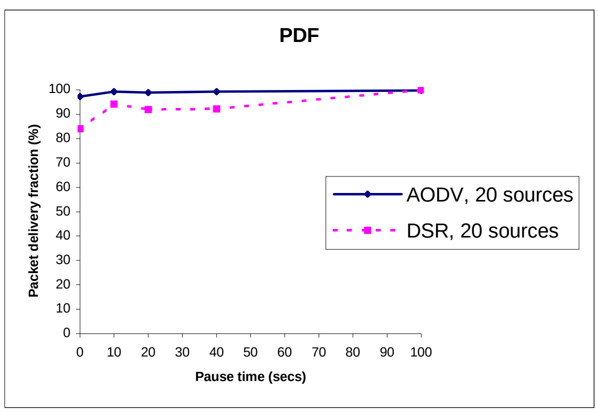

Figure 3: The packet delivery fractions for DSR and AODV

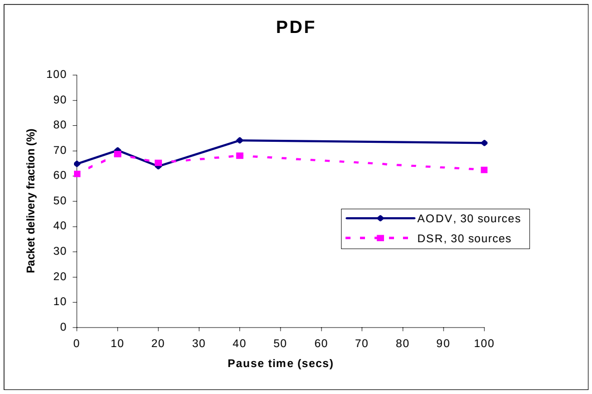

Figure 4: The packet delivery fractions for DSR and AODV

4) Normalized Routing Load Comparison

In all cases, DSR demonstrates significantly lower routing load than AODV (figure 5,6 and 7), with the factor increasing with a growing number of sources.

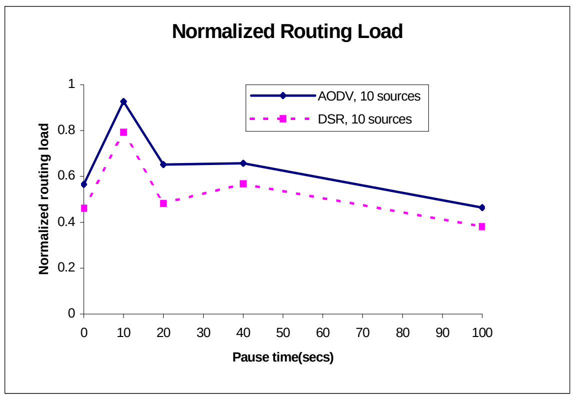

Figure 5: Normalized routing load for AODV and DSR

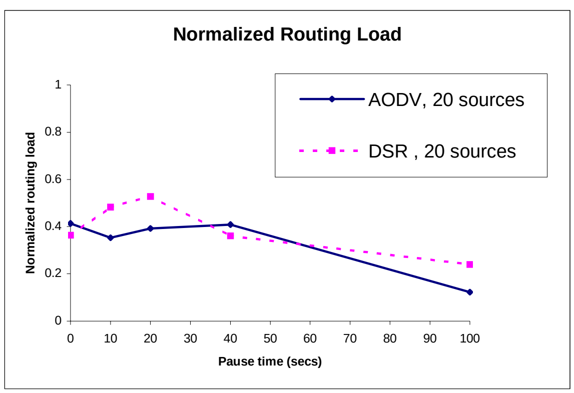

Figure 6: Normalized routing load for AODV and DSR

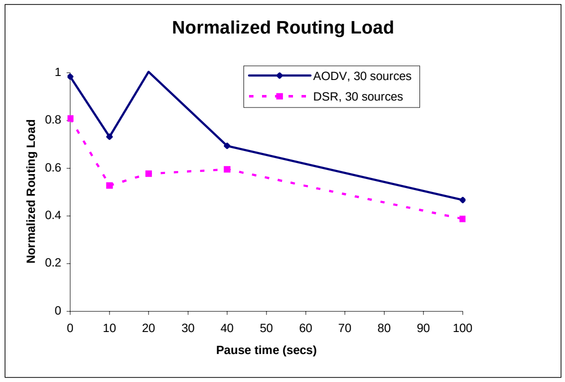

Figure 7: Normalized routing load for AODV and DSR

In summary, when the number of sources is low, the performance of DSR and AODV is similar regardless of mobility. With large numbers of sources, AODV starts outperforming DSR for high-mobility scenarios. As the data from the varying sources demonstrate, AODV starts outperforming DSR at a lower load with a larger number of nodes. DSR always demonstrates a lower routing load than AODV. The major contribution to AODV routing over-head is from route requests, while route replies constitute a large fraction of DSR routing overhead. Furthermore, AODV has more route requests than DSR and the converse is true for route replies.

The simulation results bring out some important characteristic differences between the routing protocols. The presence of high mobility implies frequent link failures and each routing protocol reacts differently during link failures. The different basic working mechanism of these protocols leads to the differences in the performance.

DSDV fails to converge below lower pause times. At higher rates of mobility (lower pause times), DSDV does poorly, dropping to a 70% packet delivery ratio. Nearly all of the dropped packets are lost because a stale routing table entry directed them to be forwarded over a broken link. As described in the earlier section, DSDV maintains only one route per destination and consequently, each packet that the MAC layer is unable to deliver is dropped since there are no alternate routes.

For DSR and AODV, packet delivery ratio is independent of offered traffic load, with both protocols delivering between 85% and 100% of the packets in all cases.

Since DSDV uses the table-driven approach of maintaining routing information, it is not as adaptive to the route changes that occur during high mobility. In contrast, the basic approach used by the on-demand protocols, AODV and DSR to build the routing information as and when they are created make its more adaptive and result in better performance (high packet delivery fraction and lower average end-to-end packet delays).

In the presence of high mobility, link failures can happen very frequently. Link failures trigger new route discoveries in AODV since it has at most one route per destination in its routing table. Thus, the frequency of route discoveries in AODV is directly proportional to the number of route breaks. The reaction of DSR to link failures in comparison is mild and causes route discovery less often. The reason is the abundance of cached routes at each node. Thus, the route discovery is delay in DSR until all cached routes fail. But with high mobility, the chance of the caches being stale is quite high in DSR. Eventually when a route discovery is initiated, the large number of replies received in response is associated with high MAC overhead and cause increased interference to data traffic. Hence, the cache staleness and high MAC overhead together result in significant degradation in performance for DSR in high mobility scenarios.

In lower mobility scenarios, DSR often performs better than AODV, because the chances of find the route in one of the caches is much higher. However, due to the constrained simulation environment (lesser simulation time and lesser mobility models), the better performance of DSR over AODV could not be observed.

DSR almost always has a lower routing load than AODV. This can be attributed to the caching strategy used by DSR. By virtue of aggressive caching, DSR is more likely to find a route in the cache, and hence resorts to route discovery less frequently than AODV.

VII. CONCLUSION

This paper compared the performance of DSDV, AODV and DSR routing protocols for ad hoc networks using NS-2 simulations. DSDV uses the proactive table-driven routing strategy while both AODV and DSR use the reactive on-demand routing strategy. Both AODV and DSR perform better under high mobility simulations than DSDV. High mobility results in frequent link failures and the overhead involved in updating all the nodes with the new routing information as in DSDV is much more than that involved AODV and DSR, where the routes are created as and when required.

DSR and AODV both use on-demand route discovery, but with different routing mechanics. In particular, DSR uses source routing and route caches, and does not depend on any periodic or timer-based activities. DSR exploits caching aggressively and maintains multiple routes per destination. AODV, on the other hand, uses routing tables, one route per destination, and destination sequence numbers, a mechanism to prevent loops and to determine freshness of routes. The general observation from the simulation is for application-oriented metrics such as packet delivery fraction and delay, AODV outperforms DSR in more “stressful” situations for example in a smaller number of nodes and lower load and mobility, with widening performance gaps with increasing stress such as more load and higher mobility. DSR however, consistently generates less routing load than AODV. The poor performances of DSR are mainly attributed to aggressive use of caching, and lack of any mechanism to expire stale routes or determine the freshness of routes when multiple choices are available. Aggressive caching, however, seems to help DSR at low loads and also keeps its routing load down.

REFERENCES

[1] N.S.M. Usop, A. Abdullah, and A.F.A Abidin, “Performance Evaluation of AODV,DSDV and DSR routing Protocol in Grid Environment,” International Journal of Computer and Network Security, vol. 9, no. 7, 2009.

[2] Samir R.Das, Charles E. Perkins, Elizabeth M. Royer. Performance comparison of Two On-demand Routing Protocols for Ad Hoc Networks.

[3] J.Broch,D.A. Maltz, D.B.Johnson, Y-C. Hu and J.Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols.

[4] 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.

[5] 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: Prentice Hall Professional Technical Reference, 2004.

[6] 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.

[7] 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).

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

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

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

AUTHORS PROFILE

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 Sultan Zainal Abidin (UniSZA). His main research areas include computer security, mobile computing and computer networks.

Nor Surayati Mohamad Usop obtained her Master of Computer Science from Faculty of Computer Science and Information Technology, Universiti Putra Malaysia in 2009. Currently, she is a Lecturer at Faculty of Informatics, Universiti Sultan Zainal Abidin (UniSZA), Malaysia.

Mohd Kamir Yusof obtained his Master of Computer Science from Faculty of Computer Science and Information System, Universiti Teknologi Malaysia in 2008. Currently, he is a Lecturer at Department of Computer Science, Faculty of Infomatics, Universiti Sultan Zainal Abidin (UniSZA), Kuala Terengganu, Terengganu, Malaysia. His main research areas include information retrieval, database integration and web semantics.