A Workload-Based Adaptive Load-Balancing Technique for Mobile Ad Hoc Networks

Young J. Lee and George F. Riley

School of Electrical & Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email: {young, riley}@ece.gatech.edu
Tel: (404) 894-9041; Fax: (404) 894-9959

***Abstract*— A novel load-balancing technique for ad hoc on-demand routing protocols is presented. Currently, ad hoc routing protocols lack load-balancing capabilities, and thus, they often fail to provide good performance especially in the presence of a large volume of traffic. We present a simple but very effective method to achieve load balance and congestion alleviation. The new scheme is motivated by the observation that ad hoc on-demand routing protocols flood route request (RREQ) messages to acquire routes, and only nodes that respond to those messages have a potential to serve as intermediate forwarding nodes. If a node ignores RREQ messages within a specific period, it can completely be excluded from the additional communications that might have occurred for that period otherwise. Thus, a node can decide not to serve a traffic flow by dropping the RREQ for that flow. In the new scheme, RREQ messages are forwarded selectively according to the load status of each node so that overloaded nodes can be excluded from the requested paths. Each node begins to allow additional traffic flows again whenever its overloaded status is dissolved. The new scheme utilizes interface queue occupancy and workload to control RREQ messages adaptively. The enhanced versions of protocols with this scheme are compared to the base protocols. Simulation results reveal that the new scheme greatly reduces packet latency as well as routing overhead without adversely affecting the network throughput, and it successfully balances the network load among nodes.**

***Index Terms*— Load balancing, ad hoc network, mobile computing**

I. Introduction

Mobile wireless ad hoc networks usually consist of mobile nodes that are not reachable through a single hop. Therefore, the main focus of ad hoc routing protocols has been to support the wireless multi-hop routing capability. These wireless links usually have lower capacity than wired links. Hence, congestion can be normal phenomenon rather than exception in mobile ad hoc networks.

Currently, ad hoc routing protocols lack load-balancing capabilities. Thus, they often fail to provide good service quality especially in the presence of a large volume of traffic since the network load concentrates on some nodes resulting in a highly congested environment. Congestion in this environment causes several undesirable effects such as long packet latency, poor packet delivery, and high routing overhead. It also causes excessive consumption of the network resources such as bandwidth and power that are usually scarce in these networks.

We present a simple but very effective method to achieve load balance and congestion alleviation in a completely distributed way. The new scheme is motivated by the observation that ad hoc on-demand routing protocols flood route request (RREQ) messages to acquire routes, and only nodes that respond to those messages have a potential to serve as intermediate forwarding nodes.

Even if each routing protocol has different features from another, most on-demand protocols share a common route discovery mechanism, which is due to their on-demand behavior. Ad hoc on-demand protocols mainly rely on flooding to find a path to a requested destination. They issue a RREQ message toward the destination when a source node does not have a valid route to the destination, and each node that receives the request responds by forwarding it. This forwarding action leaves a state in the RREQ message (source routes) or in the forwarding node itself (routing tables) that is used to compute a route. If a node does not join this RREQ forwarding action within a specific period, it can completely be excluded from the additional communications that might have occurred for that period otherwise. Thus, a node can decide not to serve a traffic flow by ignoring the RREQ for that flow.

In the new scheme, RREQ messages are forwarded selectively according to the load status of each node. Overloaded nodes do not allow additional communications to set up through them so that they can be excluded from the requested paths within a specific period. Each node begins to allow additional traffic flows again whenever its overloaded status is dissolved.

The new scheme utilizes interface queue occupancy and workload to control RREQ messages adaptively. Each node maintains a threshold value, which is a criterion for decision of whether or not to respond to a RREQ message. The threshold value dynamically changes according to the load status of a node based on its interface queue occupancy and its workload within a specific period.

This paper is organized as follows. In Section II, the new scheme is presented in detail. In Section III, the performance

 0-7803-8966-2/05/$20.00

results of the new scheme and the base schemes are presented and compared. We overview and discuss related work in Section IV, and conclude the paper in Section V.

II. Workload-Based Adaptive Load-Balancing

In on-demand routing protocols, not every node that has responded to a RREQ is guaranteed to be on the discovered path. This is because there usually exist multiple paths for the same source–destination pair in a mobile ad hoc network. However, it is obvious that only nodes that have joined the RREQ forwarding action could have a potential to be on the found path. In other words, a node can completely be excluded from a path if the node drops the RREQ in a route discovery phase for the path. Usually, this does not happen as long as a node receives a RREQ.

The new scheme enables a node to join the RREQ forwarding action selectively. Each node maintains a threshold value. This threshold value is a criterion for each node's decision of how to react to a RREQ message. When a node receives a RREQ, the node takes a simple action according to the threshold value. If the interface queue length of a node is greater than the threshold value, the node simply drops the RREQ. Otherwise, the node forwards the RREQ by rebroadcasting it. By doing so, additional traffic flows are not allowed to set up through overloaded nodes, and therefore, the overloaded nodes are naturally excluded from the newly requested paths.

The threshold value is initially set to a pre-determined value. The threshold value keeps changing according to the load status of a node. If a node experiences overload to an extent, its threshold value decreases. When the node senses that its load has been low for a long enough period, it is considered as an indication that the node's overloaded status is dissolved, and its threshold value returns to the initial value. From that time on, the node allows additional communications to set up through it as long as not overloaded. The detailed operations of the new scheme are presented in the following section.

A. Detailed Operations

In this paper, *workload* of a node is defined as the area under the graph when the interface queue length of the node is plotted over time. Thus, the unit of workload is packet · seconds. This workload is incremental since the area is accumulated over time. In the new scheme, the queue occupancy and the workload increment are used as input parameters for calculation of the threshold value.

The threshold value of a node is initially set to the maximum threshold value (max_{th}), and a node is not allowed to have the threshold value greater than max_{th} or less than the minimum threshold (min_{th}). The minimum and the maximum threshold values are pre-determined.

The threshold value of a node ranges from min_{th} to max_{th} as described above. Hence, if the queue length at the moment a node received a RREQ is greater than max_{th} or less than min_{th}, the RREQ is dropped or forwarded deterministically. However, if the queue length is between min_{th} and max_{th}, the threshold value is updated first, and then a correspondent decision is made according to the updated threshold. This situation is detailed in Fig. 1.

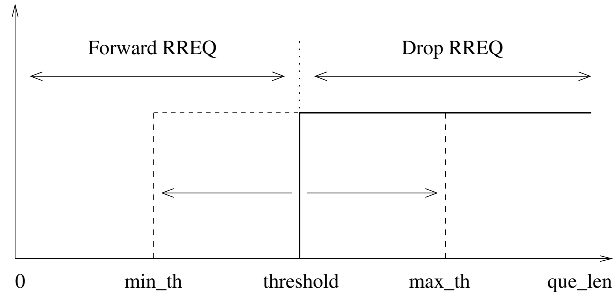

Fig. 1. Adaptive Threshold Adjustment

It is reasonable for a node to drop RREQs more frequently if the node experiences severer overload, and this feature can be achieved by making the possibility of dropping a RREQ higher, i.e., by decreasing its threshold value gradually as the node's overloaded status lasts.

In the new scheme, a node is considered as overloaded if the following conditions are met simultaneously:

- The current queue length is greater than the average of min_{th} and max_{th}.
- The *outstanding workload* ($workload_{out}$) is greater than the workload threshold ($workload_{th}$).

The outstanding workload is the workload increment within the specific period in which the first condition is satisfied. The queue occupancy information alone is not enough to evaluate the load status of a node because the queue length can be high for a short period in a transient situation even though the node is not overloaded. The outstanding workload is the mixed information of the length and the residence time of packets in the interface queue. Thus, using both information can prevent a wrong decision on the node's load status, which is the reason that the queue length and the outstanding workload are used herein together. When a node is considered as overloaded, its threshold value is decremented by the amount of $thresh_{dec}$.

A node reverts its threshold value to raise the possibility of forwarding a RREQ when its overloaded status is considered as dissolved. If the queue length of a node has been less than min_{th} for at least $dissolve_{th}$ seconds, the threshold value of the node returns to the initial value.

Each node maintains a small number of states to update its threshold value. The threshold value is updated whenever the node receives a RREQ message. The threshold update algorithm is detailed in Fig. 2:

Even though effective suppression of RREQs can be beneficial to load balance and congestion alleviation, excessive suppression can lead to partition in a sparse network. To deal with this problem, a *priority flag* is introduced in a RREQ to

Initialize:
$threshold \leftarrow max_{th}$
$load\_state \leftarrow 0$
$time_{updated} \leftarrow 0$
$workload_{prev} \leftarrow 0$

Update:
$time_{elapsed} \leftarrow now - time_{updated}$
$workload_{current} \leftarrow workload_{prev} + que\_len \times time_{elapsed}$
if $que\_len > (min_{th} + max_{th})/2$ **then**
  if $load\_state \leq 0$ **then**
    $load\_state \leftarrow 1$
    $time_{updated} \leftarrow now$
    $time_{elapsed} \leftarrow 0$
  end if
  if $time_{elapsed} > 0$ **then**
    increase $workload_{out}$ by $(workload_{current} - workload_{prev})$
  else
    $workload_{out} \leftarrow 0$
  end if
  $workload_{prev} \leftarrow workload_{current}$
  if $workload_{out} > workload_{th}$ **then**
    $threshold \leftarrow max(threshold - thresh_{dec}, min_{th})$
    $time_{updated} \leftarrow now$
    $workload_{out} \leftarrow 0$
  end if
else if $que\_len < min_{th}$ **then**
  if $load\_state \geq 0$ **then**
    $load\_state \leftarrow -1$
    $time_{updated} \leftarrow now$
    $time_{elapsed} \leftarrow 0$
  end if
  if $time_{elapsed} \geq dissolve_{th}$ **then**
    $threshold \leftarrow max_{th}$
    $time_{updated} \leftarrow now$
  end if
else
  $load\_state \leftarrow 0$
end if

Fig. 2. Threshold Update Algorithm

make each node differently process RREQs according to the flag. Each node unconditionally forwards a RREQ if the flag in the RREQ is set to one.

For example, the priority flag can be set to zero for the first cycle of route discovery so that each node can operate in a selective RREQ forwarding mode. If the try fails, then the source can attempt again with the flag set to one.

III. Performance Evaluation

In this section, we evaluate the performance of the enhanced versions of protocols with the new scheme and compare them to the base protocols. The simulation environment is explained, and the simulation results are presented and discussed.

TABLE I
The parameter values for WAL

parameter	value
min_{th}	1
max_{th}	5
$thresh_{dec}$	2
$workload_{th}$	0.5
$dissolve_{th}$	30

A. The Simulation Environment

All of our simulations were performed using the *Georgia Tech Network Simulator (GTNetS)* [3]. GTNetS is a scalable simulation tool designed specifically to support large–scale simulations. The design of the simulator closely matches the design of real network protocol stacks and hardware. Moreover, the simulator is implemented completely in object-oriented C++, which leads to easy extension for new or modified behavior of existing simulation models. For more information, refer to the GTNetS web page at [4].

The distributed coordination function (DCF) of the IEEE 802.11 [5] standard was used as the MAC protocol with a 2 Mbps-bandwidth shared medium in the simulation.

We chose AODV [1] and DSR [2] as the base routing protocols and implemented the enhanced versions with the workload-based adaptive load-balancing scheme for each base routing protocol (termed as -WAL). Each routing protocol model has a send buffer of 64 data packets with a timeout value of 30 seconds. After the timeout, the packet is expunged from the send buffer. The send buffer holds pending packets while waiting for route replies (RREPs). In addition, each wireless interface of a node has a queue that can hold up to 50 packets. This queue gives higher priority to routing protocol messages than data packets. The parameter values used for the new scheme are specified in Table I.

We constructed 30 different scenarios for the simulations. Each scenario is a set of mobility patterns and traffic patterns. The mobility model used was random-waypoint [6], and the pause time was 100 seconds. The node speed was uniformly distributed between 0 and 20 m/s (average 10 m/s).

The simulated network consists of 100 nodes with 250 m transmission range. We used 40 traffic flows and gradually increased the data rate of each flow from two to six packets/second (or 330 to 1000 Kbps) per experiment. All traffic was created with a constant-bit-rate (CBR) data source, and every packet size was 512 bytes. In the simulation, each mobile node is placed within a rectangle of 2200 m $\times$ 600 m. Each simulation executed for 500 simulated seconds.

B. Performance Results

To reflect various aspects of the routing protocols, the following performance metrics were used:

- Packet latency
- Packet delivery fraction
- Routing overhead

The packet latency is the average time taken to transfer a data packet from a CBR source to its target. In general, data packets wait in the send buffer some time until a route reply is received. This amount of time is the route acquisition time. The route acquisition time is also reflected in the packet latency. The packet delivery fraction is the ratio of the number of received data packets at the destinations to the number of data packets generated by the CBR sources. This metric captures the network throughput.

The routing overhead is the ratio of the number of routing messages generated by a routing protocol to the number of received data packets at the destinations. This metric is a measure of how many routing messages are needed to receive one data packet. It captures the efficiency of the routing protocol.

In the simulation results, each data point represents an average of 30 runs with different mobility patterns and traffic patterns. To insure a fair comparison however, an identical set of mobility and traffic patterns were applied to the routing protocols in each simulation.

As shown in Fig. 3, the new scheme greatly reduces the end-to-end delay for AODV. The delay is decreased up to 32 % by applying the new scheme to AODV. On the average, AODV-WAL demonstrated 27 % smaller delay than the base protocol. The delay performance gain gets bigger as the offered load increases. The delay performance of DSR was also improved with the new scheme. DSR-WAL showed about 14 % smaller delay than DSR on the average. DSR, however, turned out not to benefit from the new scheme as much as AODV.

This different aspect of the two protocols is mainly due to the different route discovery behavior. DSR adopts a very aggressive route discovery strategy. Thus, a DSR node learns much more route information than a AODV node in a route request cycle, which leads to high hit rate in the route cache. This means most of route replies are generated from intermediate nodes rather than destination nodes in case of DSR. Hence, even though an intermediate node is already overloaded, a route reply can be generated toward the source node without forwarding the RREQ via the overloaded intermediate node in DSR. If the RREQ had arrived at the overloaded node, it might be dropped at the node, and another path not including the overloaded node could be found, which can happen more frequently in AODV.

Fig. 4 shows that the new scheme does not adversely affect the network throughput. Rather, the performance was somewhat improved for both protocols in terms of packet delivery fraction. If RREQs are excessively suppressed in a route discovery phase, it can happen that a route is not found, and fairly lots of packets are dropped degrading the network throughput. However, this was not the case for the new scheme because it adaptively suppresses the RREQs according to the local load status, and the simulation result supports this fact.

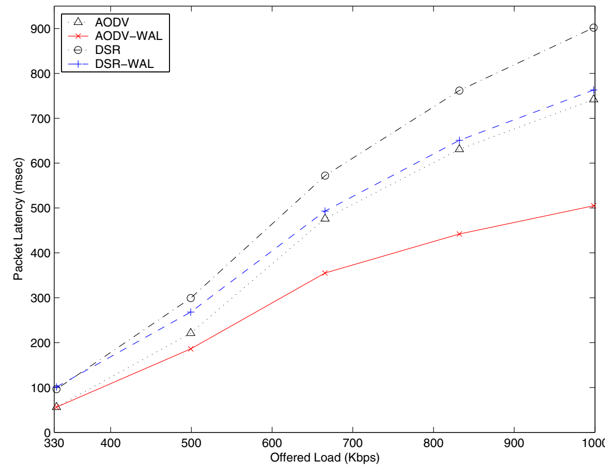

Fig. 3. Packet Latency

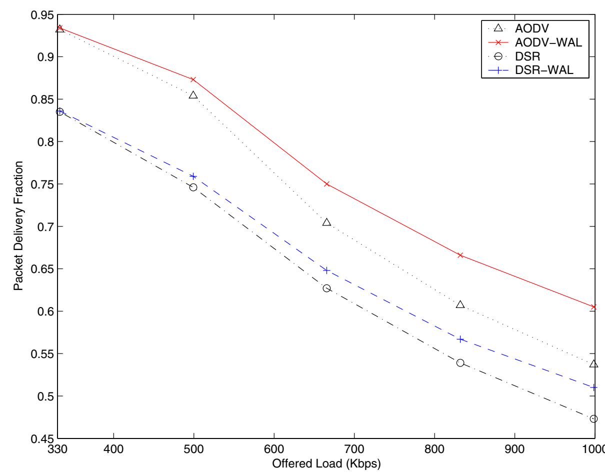

Fig. 4. Packet Delivery Fraction

As can be seen in Fig. 5, the routing overhead performance was improved a lot for AODV. AODV-WAL demonstrated up to 32 % less routing overhead than the base protocol. This performance gain was obtained mainly from suppression of RREQs. In AODV, a large portion of routing messages are RREQs [7]. AODV-WAL suppresses these RREQs effectively, and thus prevents unnecessary propagation of RREQs over the network while reducing overall routing overhead very much. Also, reduced link breakage somewhat contributed to this performance improvement. In general, a path in AODV-WAL is more stable than in the base protocol since it tries to exclude overloaded nodes from the path. Actually, the number of link breakage events in AODV-WAL was about 16 % less than in AODV. This reduced link breakage directly translates to reduction of routing messages because a link breakage event triggers a route error message. On the average, AODV-WAL showed about 20 % less routing overhead than AODV.

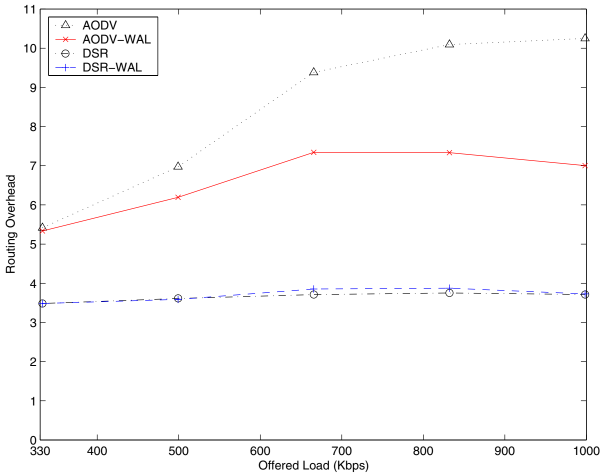

Fig. 5. Routing Overhead

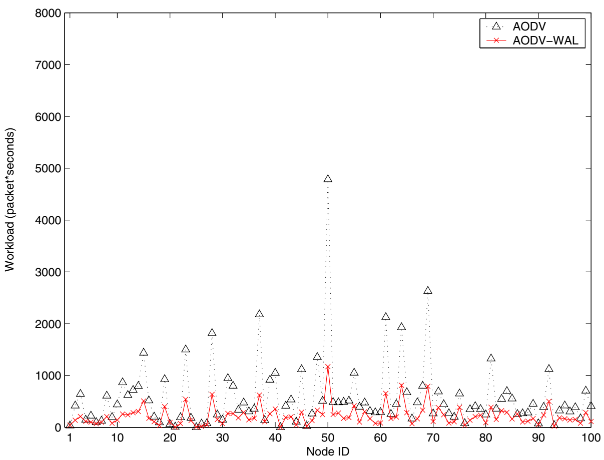

Fig. 6. Workload Distribution (Offered Load = 832 Kbps)

On the other hand, the routing overhead of DSR-WAL is very similar to that of DSR. In DSR, RREPs occupy a large portion in the entire routing messages [7]. Moreover, as explained in the delay performance case, lots of RREPs are issued from the route caches without having chances to suppress RREQs in many cases for DSR. These factors explain the similar efficiency of DSR and DSR-WAL.

Fig. 6 displays the total workload distribution over the mobile nodes in the simulated network. It is to see how well the new scheme balances load among the network nodes. Each point is the total workload of a node after the simulation and represents its load status. We can observe many peak spots for AODV nodes in the figure. They show that those nodes were overloaded severely. Also, large fluctuations are observed, which means load is biased over the network for the base protocol. For AODV-WAL, however, the total workload distribution does not fluctuate much. It is more evenly distributed than the base protocol, which shows that the new scheme successfully balances load among the network nodes.

TABLE II

PACKET LATENCY (SEC)

Data Rate (Kbps)	330	500	670	830	1000
A	0.056	0.186	0.355	0.442	0.505
B	0.056	0.181	0.340	0.420	0.478

TABLE III

PACKET DELIVERY FRACTION (%)

Data Rate (Kbps)	330	500	670	830	1000
A	93.4	87.3	75.0	66.6	60.5
B	93.3	87.7	75.8	67.6	61.5

TABLE IV

NORMALIZED ROUTING LOAD

Data Rate (Kbps)	330	500	670	830	1000
A	5.336	6.194	7.342	7.334	7.002
B	5.387	6.096	7.243	7.217	6.956

C. Parameter Setting and Sensitivity

It is not easy to choose optimal values for WAL parameters as they depend on a various range of factors such as mobility, traffic pattern, etc. Roughly speaking, RREQ suppression cannot be done effectively if the threshold is bounded within a range with high values, and the network may be throttled if it is limited to very low values. However, the threshold is adjusted adaptively according to the load status of each node. Thus, even if it starts from a different value, it will try to settle down at the point where it is considered optimal only if the range is properly set.

In order to see how different setting of WAL parameters can affect its performance, we carried out experiments for AODV-WAL with two different sets of parameters. The parameter set A represents the one used for the experiments in the previous section while the set B has different parameters with $max_{th} = 10$ and $thresh_{dec} = 3$.

The results are shown in Table II – IV. As can be observed in these tables, AODV-WAL shows very similar performance with two different sets of parameters, which implies that the WAL technique is not very sensitive to its parameter setting.

IV. RELATED WORK

Recently, several researches have been performed in the ad hoc networking domain in order to balance the network load, to mitigate congestion, and to provide stable packet delivery.

In [8], routing load of the intermediate nodes is used as the primary route selection metric. A RREQ message keeps recording queue occupancy information of each node it visits, and the destination selects a path that it considers as the best based on the queue occupancy information recorded in the RREQs. This scheme, however, lacks path diversity since it is a single-path mechanism. Therefore, its load-balancing capability is limited.

In [9], [10], path diversity is a main concern. To utilize path diversity, multiple paths are found per destination, and they are maintained at source nodes and used in turn for routing. The basic idea of these schemes is to distribute traffic among multiple paths. These schemes, however, need to maintain complex states to dynamically select a routing path among the multiple discovered paths. These multi-path protocols also incur additional routing overhead due to maintaining more than one routes per destination when compared to single-path protocols [11]. Moreover, it is known that multi-path routing is effective when the alternate paths are disjointed, which is not easy to achieve in mobile ad hoc networks [12], [13].

All of these schemes can be classified into *end-to-end* approaches since source and/or destination nodes are responsible for selecting and maintaining single or multiple paths. On the other hand, our new scheme is differentiated from the other schemes in that it operates in a fully distributed manner and does not utilize any global information. The new scheme makes each node react to RREQs according to a simple rule based on the local information of the node, and it runs on top of existing routing protocols.

V. Conclusions

We presented a novel load-balancing technique for mobile ad hoc networks. The new scheme is simple but very effective to achieve load balance and congestion alleviation. It enables each node to forward RREQ messages selectively according to the load status of the node. Overloaded nodes do not allow additional communications to set up through them so that they can be excluded from the requested paths within a specific period. Each node allows additional traffic flows as long as it is not overloaded.

The new scheme utilizes interface queue occupancy and workload to control RREQ messages adaptively. In the new method, each node maintains a threshold value, which is a criterion for decision of how to react to a RREQ message. The threshold value of a node dynamically changes according to the load status of the node based on its queue occupancy and its workload within a specific period.

We showed via simulation that the new scheme significantly reduces packet latency as well as routing overhead especially for *AODV-type* on-demand protocols, where RREQs dominate the entire routing messages. It was also shown that the network throughput is not adversely affected but rather improved by applying the new scheme to the base protocols. The new scheme successfully balances the network load among nodes, and it can easily be incorporated with existing on-demand routing protocols to work on top of them.

References

[1] C. Perkins and E. Royer, “Ad hoc on-demand distance vector routing,” *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA’99),* February 1999.

[2] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” *Mobile Computing, edited by T. Imielinski and H. Korth, Ch. 5, pp. 153-181,* Kluwer Academic Publishers, 1996.

[3] G. Riley, “The Georgia Tech Network Simulator,” *ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools’03),* August 2003.

[4] http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS

[5] IEEE Computer Society, “802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” June 1997.

[6] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” *ACM Int’l Conference on Mobile Computing and Networking (MobiCom’98),* October 1998.

[7] S. Das, C. Perkins, and E. Royer, “Performance comparison of two on-demand routing protocols for ad hoc networks,” *IEEE Conference on Computer Communications (INFOCOM 2000),* March 2000.

[8] S. Lee and M. Gerla, “Dynamic load-aware routing in ad hoc networks,” *IEEE Int’l Conference on Communications (ICC 2001),* June 2001.

[9] M. Pearlman, Z. Haas, P. Scholander, and S. Tabrizi, “Alternate path routing in mobile ad hoc networks,” *IEEE Military Communications Conference (MILCOM 2000),* October 2000.

[10] P. Pham and S. Perreau, “Multi-path routing protocol with load balancing policy in mobile ad hoc network,” *IFIP Int’l Conference on Mobile and Wireless Communications Networks (MWCN 2002),* September 2002.

[11] P. Pham and S. Perreau, “Performance analysis of reactive shortest path and multi-path routing mechanism with load balance,” *IEEE Conference on Computer Communications (INFOCOM 2003),* March 2003.

[12] M. Pearlman, Z. Haas, P. Scholander, and S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks,” *ACM Int’l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2000),* August 2000.

[13] A. Nasipuri, R. Castaneda, and S. Das, “Performance of multipath routing for on-demand protocols in ad hoc networks,” *ACM/Kluwer Mobile Networks and Applications (MONET) Journal, Vol. 6, No. 4, pp. 339-349* August 2001.

