Routing with Load Balancing in Wireless Ad hoc Networks

Hossam Hassanein and Audrey Zhou

Department of Computing and Information Science
Queen's University
Kingston, Ontario, Canada, K7L 3N6
{hossam, zhou}@cs.queensu.ca

Abstract

An ad hoc wireless mobile network is an infrastructure-less mobile network that has no fixed routers; instead, all nodes are capable of movement and can be connected dynamically in an arbitrary manner. In order to facilitate communication of mobile nodes that may not be within the wireless range of each other, an efficient routing protocol is used to discover routes between nodes so that messages may be delivered in a timely manner. In this paper, we present a novel Load-Balanced Ad hoc Routing (LBAR) protocol for communication in wireless ad hoc networks. LBAR defines a new metric for routing known as the degree of nodal activity to represent the load on a mobile node. In LBAR routing information on all paths from source to destination are forwarded through setup messages to the destination. Setup messages include nodal activity information of all nodes on the traversed path. After collecting information on all possible paths, the destination then makes a selection of the path with the best-cost value and sends an acknowledgement to the source node. LBAR also provides efficient *path maintenance* to patch up broken links by detouring traffic to the destination. A comprehensive simulation study was conducted to evaluate the performance of the proposed scheme. Performance results show that LBAR outperforms existing ad hoc routing protocols in terms of packet delivery and average end-to-end delay.

Key words: Wireless Ad hoc Networks, Routing, Load Balancing, Performance Evaluation, Path Discovery, and Path Maintenance

MSWiM 2001 7/01 Rome, Italy
© 2001 ACM ISBN 1-58113-378-2/01/07...$5.00

1. Introduction

A mobile ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any centralized administration or standard support services regularly available in wide-area networks to which the hosts may normally be connected. In such an environment, it may be necessary for one mobile host to seek the aid of others in forwarding a packet to its destination, due to the limited propagation range of each mobile host's wireless transmissions.

A critical challenge in the design of ad hoc networks is the development of efficient routing protocols that can provide high-quality communication between two mobile nodes. Numerous routing protocols have been developed for ad hoc mobile networks. These protocols may generally be categorized *as table-driven* and *on-demand* routing. Table driven routing protocols [1-5] attempt to maintain consistent, up-to-date routing information in each node by propagating updates throughout the network. Such protocols, and although a route to every other node is always available, incur substantial signaling traffic and power consumption. Since both bandwidth and battery power are scarce resources in mobile computers, this becomes a serious limitation to table-driven routing protocols. On the other hand, on-demand routing protocols [6-10] overcome this limitation. This type of routing protocols does not maintain routing information at every node, but create routes only when desired by the source node. When a source has a packet to transmit, it invokes a route discovery mechanism to find the path to the destination. The route remains valid until the destination is reachable or until the route is no longer needed. In fact, on-demand routing is dominating the tendency for wireless ad hoc communication. Among all proposed wireless mobile ad hoc routing protocols, Dynamic Source Routing (DSR) [8, 9] and Ad hoc On-demand Distance Vector (AODV) [10] are the most prominent, and have been submitted to the Internet Engineering Task Force (IETF) Mobile Ad hoc NETworking (MANET) working group [11] as candidates for standardization. DSR [8, 9] utilizes source routing in ad hoc networks to discover routes from source nodes to destination nodes. AODV [10] maintains routes as long as they are needed by the sources. If a source node moves, or a hop on the route from the source node to the destination node becomes unreachable, route discovery from the source to the destination must be reinitiated if it still requires a route to the corresponding destination.

It has been long believed that the performance of ad hoc networks routing protocols is enhanced when nodal mobility is reduced. This is true when considering performance measures such as packet delivery fraction and routing overhead. This may not be the case, however, when we consider packet delay. It was shown in [12] that the packet delay for both AODV and DSR increases as the nodal mobility is reduced. This is because there is a tendency in ad hoc networks routing protocols to use a few "centrally located" nodes in a large number of routes. This causes congestion at the medium access control (MAC) level, which in turn may lead to high packet delays, since few nodes have to carry excessive loads. Such nodes may also suffer from high battery power consumption. This is an undesirable effect, which is compounded by the limited battery power of the mobile terminals. In fact, a major drawback of all existing ad hoc routing protocols is that they do not have provisions for conveying the load and/or quality of a path during route setup. Hence they cannot balance the load on the different routes.

In this paper, we propose an efficient routing protocol, based on the concept of balancing traffic load, namely, the Load-Balanced Ad hoc Routing (LBAR) protocol. The proposed scheme is intended to route data packets circumventing congested paths so as to balance traffic load over the network and lower end-to-end delay. Additionally, the protocol demonstrates quick response to link failures incurred by topology changes in the ad hoc network and thereby improves data delivery reliability. Performance results indicate that LBAR outperforms both AODV and DSR in terms of packet delivery fraction and average end-to-end delay. The paper is organized as the follows. In Section 2, the details of the proposed LBAR scheme are described. Simulation results and analysis are reported in Section 3. Finally, Section 4 presents conclusions and future work.

2. LBAR Routing

The proposed Load-Balanced Ad hoc Routing (LBAR) is an on-demand routing protocol intended for delay-sensitive applications where users are most concerned with packet transmission delay. Hence, LBAR focuses on how to find a path, which would reflect least traffic load so that data packets can be routed with least delay.

The route discovery process is initiated whenever a source node needs to communicate with another node for which it does not have a known route. The process is divided into two stages: *forward* and *backward*. The forward stage starts at the source node by broadcasting setup messages to its neighbors. A setup message carries the cost seen from the source to the current node. A node that receives a setup message will forward it, in the same manner, to its neighbors after updating the cost based on its nodal activity value. In order to prevent looping when setup messages are routed, all setup messages are assumed to contain a route record, including a list of all node IDs used in establishing the path fragment form the source node to the current intermediate node. The destination node collects arriving setup messages within a route-select waiting period, which is a predefined timer for selecting the best-cost path. The backward stage begins with an ACK message forwarded backward towards the source node along the selected path, which we call the *active path*. If a link on the selected path breaks, the ACK message is discarded and an error message is sent backward along the path fragment to the destination. The destination node will then choose another path, which does not contain any of the previous broken links. When the source node receives an ACK message, it knows that a path has been established to the destination and then starts transmission.

When either the destination node or some intermediate node moves outside the active path, path maintenance will be initiated to correct the broken path. Once the next hop becomes unreachable, the node upstream of the broken hop propagates an error message to the destination node. Upon receiving notification of a broken link, the destination node picks up an alternative best-cost partial route passing through the node propagating the error message and then sends an ACK message to the initiator of the error message. If the destination has no alternative path passing through the node sending the error message, the destination picks up another route and sends an ACK message to the source. The source will use this new route to send data packets if it still has data to send. By then, a new active path is defined. In the worst case, where the destination has no alternate paths, it propagates an error message to the source and lets it restart route discovery.

Nodes learn about their neighbors in one of two ways. Whenever a node receives a broadcast from a neighbor, it updates its local connectivity information in its *Neighborhood table* to ensure that it includes this neighbor. In the event that a node has not sent data packets to any of its active neighbors within a predefined timeout, *hello_interval*, it broadcasts a *hello* message to its neighbors, containing its identity and activity. This hello message is prevented from being rebroadcast outside the neighborhood of the node. Neighbors that receive this packet update their local connectivity information in their Neighborhood tables. Receiving a broadcast or a hello from a new neighbor, or failing to receive consecutive hello messages from a node previously in the neighborhood, is an indication that the local connectivity has changed. If hello messages are not received from the next hop along an active path, the upstream active neighbors using that next hop send notification of link failure and the path maintenance protocol is invoked.

A cost function is used to find a path with the least traffic so that data packets can be transmitted to the destination as fast as possible while achieving the goal of balancing load over the network. The following definitions are used:

- *Active path*: a path from a source to a destination, which is followed by packets along this selected route.
- *Active node*: a node is considered *active* if it originates or relays data packets or is a destination.
- *Activity*: The number of active paths through a node is defined as a metric measuring the activity of the node.
- *Cost*: Minimum traffic interference is proposed as the metric for best cost.

In wireless ad hoc networks, transmitters use radio signals for communication. Communication among mobile nodes is limited within a certain transmission. Within each such range, only one transmission channel is used, covering the entire available bandwidth. To transmit data, mobiles within the same range have to sense for other transmissions first and then gain access permit and transmit only if no other node is currently transmitting. Unlike wired networks, packet delay is not caused only from traffic load at the current node, but also by traffic load at neighboring nodes. We call this *traffic interference*. In the context of traffic interference, the best-cost route is regarded as a path, which encounters the minimum traffic load in transmission and minimum interference by neighboring nodes. To assess best cost, the term *node activity* is used as an indirect means to reflect traffic load at the node. Such activity information can be gained at the network layer, independent of the MAC layer. Traffic interference is defined as the sum of neighboring activity of the current node. During the routing stage, nodal activity and traffic interference are calculated at every intermediate node along path from source to destination. When the destination receives routing information, it chooses a path, which has minimum cost. We define the following:

- Activity A_i: Number of active paths through node i. The greater the value of activity is, the more traffic passing through node i would be.
- Traffic interference TI_i: $TI_i = \sum_{\forall j} A_j^i$, which is the sum of activity of neighboring nodes of node *i*, where j is a neighboring node of node i.
- Cost C_k: cost of route k. $C_k = \sum_{i \in k} (A_i + TI_i) = \sum_{i \in k} (A_i + \sum_{\forall j} A_j^i)$ where i is a node on path k other than source and destination. (Every path with identified source-destination pair includes same source and destination, so for simplicity, activities of source and destination are excluded.) j is a neighboring node of node i.

This is a generic cost function, which is based on the assumption that packets are of the same size and traffic is at a constant rate. Other alternative functions can be also used without impacting the generality of the proposed load-balancing routing protocol.

A pseudo code description of the algorithm for the source node is shown in Figure 1. Lines 1-2 represent the beginning of the forward stage, where a request to establish a path is initiated. The source node begins to forward this request to its neighbors. Lines 3-4 indicate the path has been found. Therefore, the source can begin transmitting data. Lines 5-6 describe the case where the source restarts the request if it does not receive acknowledgement from destination that a path has been established. When the source node receives an error notification indicating that destination cannot find alternate paths, it must restart route discovery (lines 7-8).

A pseudo code description of the algorithm for an intermediate node in any reachable path is shown in Figure 2. Lines 1-2 represent the forward stage of the scheme, where the node forwards setup message to its neighbors, avoiding already visited nodes. Lines 3-6 show the backward stage of the algorithm, in which acknowledgement is sent backward to upstream nodes if next link is not broken. Otherwise, an error notification is sent from the node along the path fragment to indicate the failure of the candidate path. Error notification is relayed downstream until it reaches the destination to pick an alternate path (lines 7-8). Upon receiving new partial route from destination, packets are redirected on this new partial route (lines 9-13).

A pseudo code description of the algorithm for the destination node is shown in Figure 3. Lines 1-2 represent the forward stage and the start of the backward stage. The route and cost information is stored at the destination routing table. If the route-select time period enforced at the destination node is reached, the path with the minimum cost is selected to begin the backward stage. Lines 3-9 represent the case when destination node receives error notification of link breakage. The destination node removes all the invalid paths associated with broken links. If an alternate path passing through the node detecting link breakage exists, the destination node selects this path to notify the error-detecting node. If no alternative path passing through the error-detecting node exists, the destination node selects another route with second best cost to notify the source. The source will use this new route to send data packets if it still has data to send. In the worst case, where the destination has no alternate paths, it notifies the source to restart route discovery.

3. Performance Evaluation of LBAR Routing

We have constructed a packet-level simulator that allows us to observe and measure the protocol's performance under a variety of conditions. The model is similar to that in [12]. Our simulations are run using ad hoc networks of 50 nodes under a nominal bit rate of 2 Mbps. Mobile terminals move with a speed that is uniformly distributed between 0 and 20 m/sec. In addition, mobility is varied by means of varying the pause/rest period. For every variation of the traffic sources, the experiments are run for a set of pause periods. The smaller the pause period, the higher the mobility, and, the greater the pause period, the lower the mobility. This implies that varying the length of the pause period is equivalent to varying the mobility model. Each and every mobile node alternately rests and moves to a new random location within the rectangular grid. Experiments were run for pause periods of 100, 300, 600 and 900 seconds in case of 50 nodes. It was also possible to vary mobility by changing the velocities by which the nodes are moving by within the closed coverage grid. Mobiles can communicate only within a constant range of 200m. A CSMA technique with collision avoidance (CSMA/CA) is used to transmit packets [13]. Packets not delivered in three attempts are dropped. The experiments use different number of sources with a moderate packet rate and changing pause times. We use 10, 20, 30, and 40 traffic sources and a packet rate of 4 packets/sec. Mobile nodes are free to move in a 1500m × 300m grid. The route-select timer for LBAR is set to 10ms.

```
For source node S
If S has new packets to send and no route is known to the targeted destination
    Then forward setup message to all available neighbor nodes of S
If S receives acknowledgement from destination that the route has been built
    Then start transmission
If S does not receive acknowledgement from destination within a route-discovery waiting period
    Then restart route discovery
If S receives error notification that destination does not have an alternate path
    Then restart route discovery
```

Figure 1: The algorithm for source node

```
For intermediate node i
If node i receives setup message
    Then node i forwards this message to all its unvisited neighbors and this message records every visited node to build a backward path
If node i receives acknowledgement from destination
    If node i's upstream neighbor on the backward path is reachable
        Then pass this acknowledgement upward to that upstream neighbor node and record necessary information to build active path
        Else send error notification to destination
If node i receives error notification of link breakage
    Then pass this error notification to its neighbor nodes to notify destination to choose a new route and tear down the active path
If next hop on the active path is still reachable
    Then send data packet to next hop
    Else If a new path has been established
        Then detour this packet along this new partial route to destination
        Else buffer packet and send error notification to destination to find an alternate path
```

Figure 2: The algorithm for intermediate node

```
For destination node D
If D receives setup message
    Then store the route and cost contained in the message. Select path with best cost and send Ack to source after route-select timer expires
If D receives error notification of link breakage
    Then remove paths containing broken links
    If an alternate path passing through the node detecting link breakage exists
        Then select this path and notify the node that detected the error
        Else If an alternate path through source exists
            Then select best-cost path and notify source
            Else send error notification to source
```

Figure 3: The algorithm for destination node

3.1 Performance metrics

Three key performance metrics are evaluated: (1) *Packet delivery fraction* – ratio of the data packets delivered to the destination to those generated by the CBR sources, which reflects the degree of reliability of the routing protocol; (2) *Average end-to-end delay* of data packets – this includes all possible delays caused by queuing for transmission at the node, buffering data for detouring, retransmission delays at the MAC, propagation delay and transmission time. It represents the quality of the routing protocol. (3) *Normalized routing load* – the number of routing packets transmitted per data packet delivered at the destination, which evaluates the efficiency of the routing protocol in terms of extra load introduced to the network.

3.2 Simulation Results

Figure 4 shows the packet delivery fractions for variations of the pause time for LBAR, AODV, and DSR. Note that the packet delivery fractions for LBAR, AODV, and DSR are very similar for both 10 and 20 sources. With 30 and 40 sources, however, LBAR outperforms AODV and DSR. In fact, LBAR achieves the highest packet delivery fraction for all pause time values. For 30 sources, LBAR achieves up to 20% higher packet delivery fractions than both AODV and DSR. This is mainly because of redundant route information that is stored in destination node to provide aid in routing, which eliminates the necessity of source reinitiating of route discovery. Similarly, LBAR has superior performance to both AODV and DSR in the case of 40 sources, in terms of the packet delivery fraction.

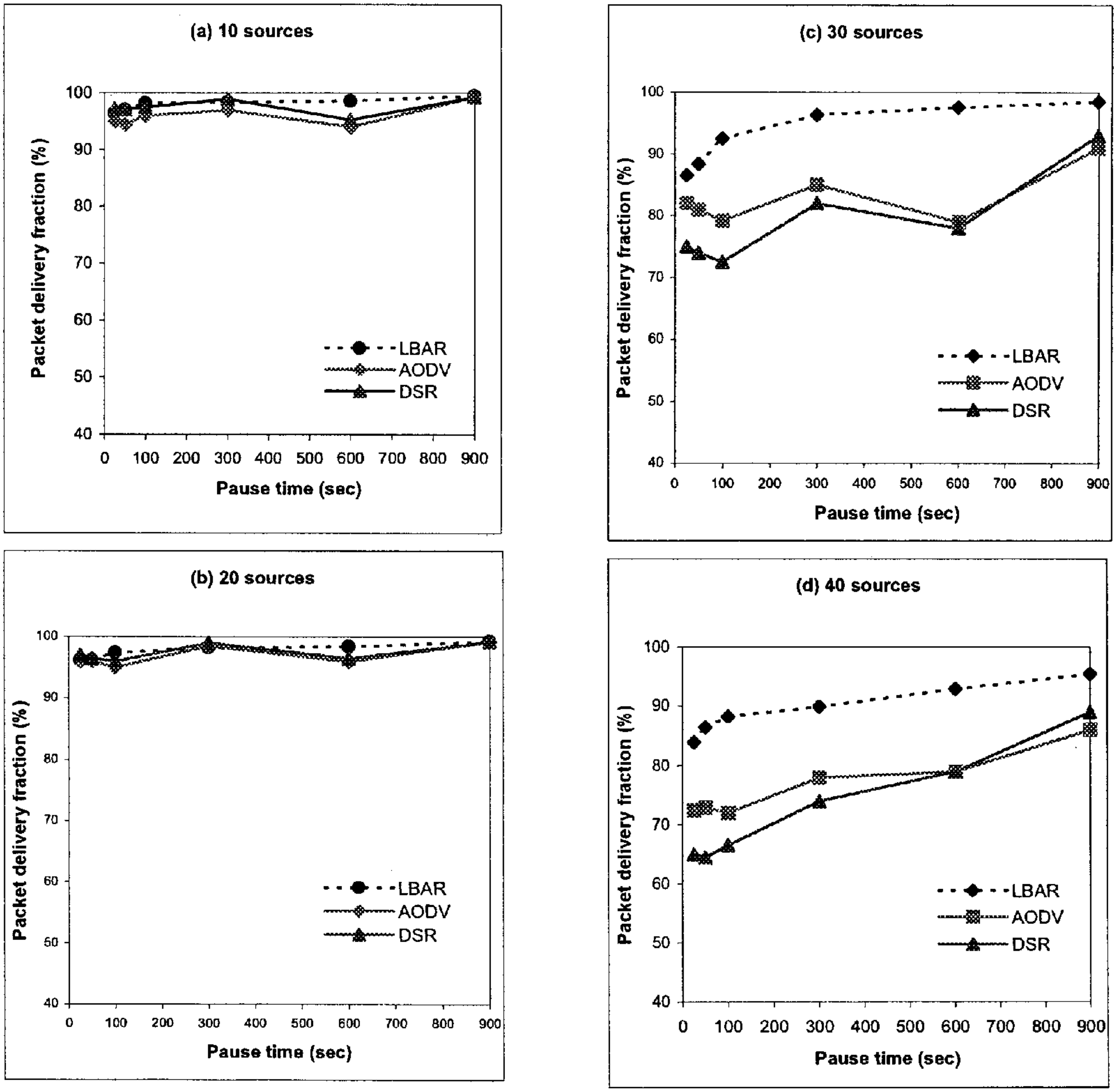

Figure 4: Packet delivery fraction

Also, LBAR has a better average end-to-end delay than both AODV and DSR (see Figure 5). For 30 and 40 sources, LBAR achieves significantly lower delay than AODV and DSR. Moreover, the delays decrease with lower mobility for LBAR in all four cases while it increases with 30 and 40 sources for both AODV and DSR. This is due to a high level of network congestion and multiple access interference in certain regions of the ad hoc network. Neither AODV nor DSR has any mechanism for load balancing, i.e., for choosing routes in such a way that the data traffic can be more evenly distributed in the network. This phenomenon is less visible with higher mobility where traffic automatically gets more evenly distributed due to source movements. In contrast, LBAR adopts a mechanism for load balancing, which tries to route packets along a less congested path to avoid overloading some nodes.

The routing load results, see Figure 6, show that the routing load of all three protocols increases with increasing the number of sources. This is because the increase in the number of source nodes causes a greater number of request messages flooding. LBAR demonstrates a higher routing load than both AODV and DSR. AODV and DSR only accept the first request message at every node, that is, if a node has already seen a request message for a particular packet, it will not accept a second message of the same packet. On the other hand, LBAR accepts request messages as long as they are not looping through the node. Destination nodes keep a record of different route information from request messages as backup for use during the path maintenance protocol. Therefore, LBAR will almost always have an alternative path to detour packets in case of link failure. This enables LBAR to achieve higher packet delivery fractions and lower average end-to-end delays.

4. Conclusion

In this paper, we have proposed a novel on-demand routing scheme, namely the Load-Balanced Ad hoc Routing (LBAR) protocol. In LBAR, routing information on different paths is forwarded through setup messages to the destination. The destination node selects the path with the minimum cost, which is measured by nodal activity. By weighing total nodal activity of a path, congested paths can be avoided, as packets are transmitted along the least-active path. As a consequence, traffic over the ad hoc network tends to be evenly distributed in the long term. In addition, in order to keep up with frequent topology change, LBAR provides quick response to link failure by patching up the broken routes in use, thus guaranteeing reliability of data transmission. In LBAR, route information stored at the destination node is used to select alternate paths whenever possible. The performance of the proposed LBAR protocol has been studied through a simulation study. Simulation results have clearly shown the advantages of LBAR over DSR [8-9] and AODV [10] in terms of packet delivery fraction and average end-to-end delay.

References

[1] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Computer and Communication*, Oct. 1994, pp.234-244.

[2] C. -C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel" Proc. *IEEE SINGAPORE CONFERENCE – IEEE SICON'97*, Apr.1997, pp.197-211.

[3] Tsu-Wei Chen and Mario Gerla, "Global State Routing: A New Routing Scheme for Ad hoc Wireless Networks" *IEEE International Conference on Communications (ICC), 1998.*

[4] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *ACM Mobile Networks and Application Journal,* Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[5] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks", *IEEE Journal on Selected Areas in Communications,* Special Issue on Ad hoc Networks, Aug. 1999, pp.1369-79.

[6] C-K. Toh, "A Novel Distributed Routing Protocol to Support Ad Hoc Mobile Computing" Proc. *1996 IEEE 15th Annual International Phoenix Conference Computer and Communication*, Mar. 1996, pp. 480-86.

[7] C-K. Toh, "Long-lived Ad Hoc Routing Based on the Concept of Associativity" *Internet Engineering Task Force -- IETF Draft*, Mar. 1999.

[8] David B. Johnson, Davis A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks" *IETF Draft*, Oct. 1999.

[9] David B. Johnson, Davis A. Maltz, "Dynamic Source Routing in Ad Hoc Networks", *Mobile Computing,* T. Imielinski and H. Korth, Eds. Kulwer, 1996, pp. 152-81.

[10] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, "Ad Hoc On-demand Distance Vector Routing", *IETF Draft*, Oct. 1999.

[11] S. R. Das, C. E. Perkins and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *IEEE INFOCOM*, March 2000.

[12] IETF MANET Working Group Charter http://www.ietf.org/html.charters/manet-charter.html.

[13] IEEE Standards Department. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE standard 802.11, 1997.

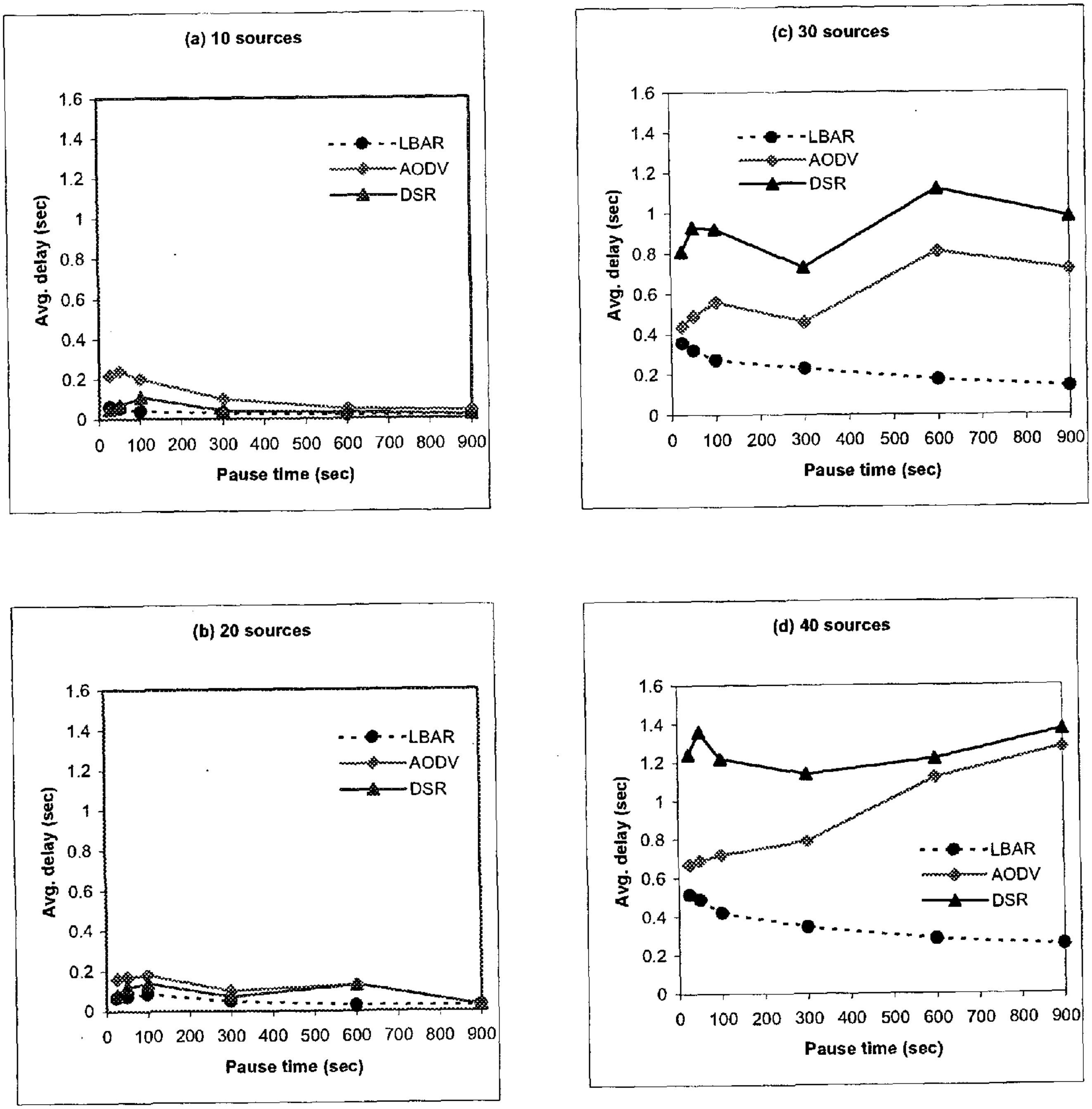

Figure 5: Average end-to-end delay

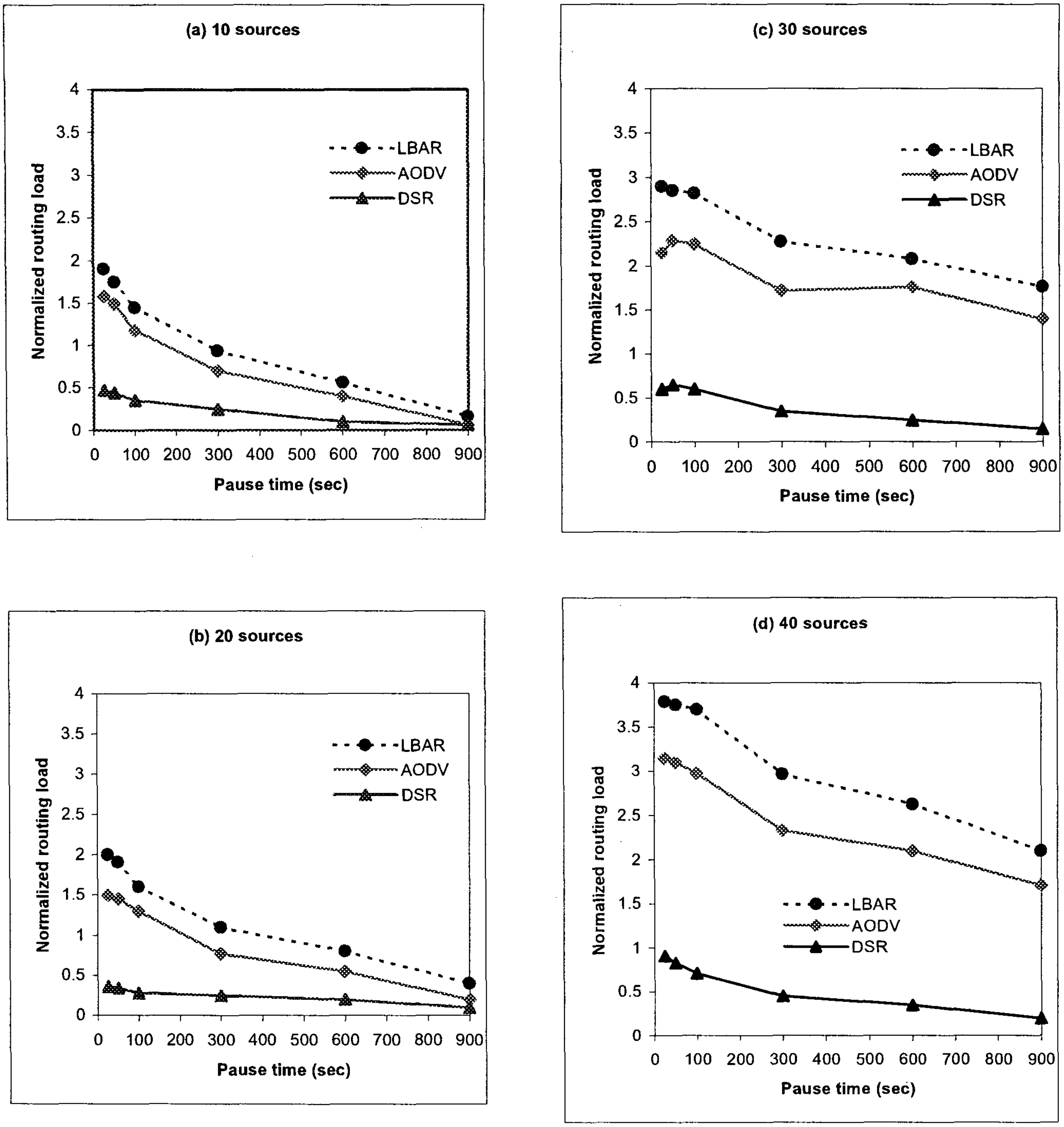

Figure 6: Normalized routing loading