Proceedings of IEEE 2008 6th National Conference on Telecommunication Technologies and
IEEE 2008 2nd Malaysia Conference on Photonics, 26-27 August 2008, Putrajaya, Malaysia

LBAODV: A New Load Balancing Multipath Routing Algorithm for Mobile Ad hoc Networks

Amir Darehshoorzadeh
Computer Engineering Department, Eyvanakey Institute of Higher Education, Iran
darehshoori@eyc.ac.ir

Nastooh Taheri Javan
Computer Engineering Department, AmirKabir University of Technology Tehran, Iran
nastooh@ce.aut.ac.ir

Mehdi Dehghan
Computer Engineering Department, AmirKabir University of Technology Tehran, Iran
dehghan@ce.aut.ac.ir

Mohammad khalili
Computer Engineering Department, AmirKabir University of Technology Tehran, Iran
mkhalili@ce.aut.ac.ir

Abstract- An ad hoc network is comprised of mobile hosts without any wired infrastructure support. Multipath routing allows the establishment of multiple paths between a source and a destination. It distributes traffic among multiple paths instead of routing all the traffics along a single path. In this paper, we propose a new multipath routing protocol that uses all discovered paths simultaneously for transmitting data, by using this approach data packets are balanced over discovered paths and energy consumption is distributed across many nodes through network.

Keywords: Ad hoc, Multipath, Load balancing

I. Introduction

An ad hoc network is a dynamically reconfigurable wireless network with no fixed wired infrastructure. Each node can function both as a network host for transmitting and receiving data and as a network router for routing packets to the other nodes. Ad hoc networks have numerous practical applications such as military applications, emergency operations, and wireless sensor networks.

In such networks, nodes are typically distinguished by their limited power, processing, and memory resources as well as high degree of mobility. Due to the limited transmission range of wireless network nodes, multiple hops are usually needed for a node to exchange information with any other node in the network. Thus routing protocols play an important role in ad hoc network communications.

On-demand routing is the most popular routing approach in ad hoc networks. Instead of periodically exchanging routing messages in proactive routing protocols which brings in excessive routing overhead [1,4], on-demand routing algorithms discover routes only when a node needs to send data packet to a destination and does not have any route to it. Most of the existing on-demand routing protocols (for example, Dynamic Source Routing (DSR) and Ad hoc On-demand Distance Vector (AODV) build and rely on single path for each data session. So route recovery process is needed after each route failure, which causes to lose transmitted data packets, in such protocols. Multipath routing allows the establishment of multiple paths between a single source and single destination node. It is typically proposed in order to increase the reliability of data transmission (i.e., fault tolerance) or to provide load balancing [2, 5, 6 and 7]. In such protocols, traffic is not distributed into multiple paths; only one route is primarily used and alternate paths are utilized only when the primary route is broken.

AOMDV is an extension to the AODV protocol for computing multiple loop-free and link-disjoint paths. To keep track of multiple routes, the routing entries for each destination contain a list of the next-hops along with the corresponding hop counts. For each destination, a node maintains the advertised hop count, which is defined as the maximum hop count for all the paths. AOMDV can be used to find node-disjoint or link-disjoint routes.

Our goal here is to develop an *on-demand multipath distance vector* protocol as an extension to a well-studied single path routing protocol known as Ad hoc On-demand Distance Vector (AODV). We refer to the new protocol as Load Balancing Ad hoc On-demand Distance Vector (LBAODV) protocol. Primary design goal behind LBAODV is to provide multiple routes on which the source sends data simultaneously over them.

In AODV source sends data through shortest path, thus only a few nodes are involved for transmitting data and their remaining energy reduce dramatically, but in LBAODV since multiple routes are used, energy is distributed across all nodes on discovered routes.

The reminder of this paper is organized as follows. Section 2 describes AODV mechanism. Sections 3 describe our new proposed algorithm for searching multiple routes and load balancing data over discovered routes. Section 4 follows with the simulation results and concluding remarks are made in section 5.

II. Ad Hoc on Demand Distance Vector Routing

The AODV [3, 8, 9] routing protocol is an on-demand routing protocol. When a node wants to send data, it first checks its routing table if an entry for this destination node does not exists; the source node has to initialize a route discovery. This process is initiated by creating a RREQ message, including the hop count to the destination, the IP address of the source and the destination, the sequence numbers of both of them, as well as the broadcast ID of the RREQ. All nodes which receive the RREQ first checked by comparing the identifier of the message with identifiers of messages already received. If it is not the first time the node sees the message, it discards silently the message. If this is not the case the node processes the RREQ by updating its routing table with the reverse route. If a node is the destination node or has already an active route to the destination in its routing table with sequence number of the destination host which is higher than the one in the RREQ, it creates a RREP message and unicasts it to the source node. Otherwise it increments the RREQ's hop count and then rebroadcasts the message to its neighbors.

When the source node receives no RREP as a response on its RREQ a new request is initialized with a higher TTL, wait value and a new ID. It retries to send a RREQ for a fixed number of times after which, when not receiving a response, it declares that the destination host is unreachable.

For example in fig. 1, *S* wants to send data to the *D* and no route information is known, therefore *S* sends RREQ and nodes *A*, *C* and *B* that are in its transmission range, receive RREQ and rebroadcast it until RREQ receives to the *D*.

In fig. 1, *D* replies back to the source through *G*. When the source receives the RREP, it records the route to the destination and can begin sending data. If multiple RREPs are received by the source, the route with the shortest hop count is chosen. According to the fig. 1, established route is *S-C-G-D*. If a route is not used for some period of time, a node cannot be sure whether the route is still valid or not; consequently, the node removes the route from its routing table.

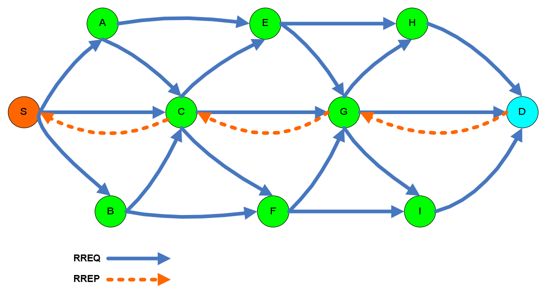

Figure 1. Mechanism of AODV routing protocol

Route maintenance is done by means of route error (RERR) packets. When an intermediate node detects a link breakage, it generates a RERR packet. The RERR propagates towards all traffic sources having a route via the failed link, and erases all broken routes on the way. A source upon receiving the RERR initiates a new route discovery if it still needs the route.

III. Load Balancing Ad hoc On-demand Distance Vector Routing Protocol (LBAODV)

Similar to AODV, LBAODV is an on-demand routing protocol that consists of three main phases:

a) *Path discovery process*

When a node has data to send and no route information is known, it initiates path discovery process by sending route request packet (RREQ) to its neighbors. The RREQ packet identifies the host, referred to as the *target* of the route discovery, for which the route is requested. In addition to the address of the original initiator of the request and the target of the request, each route request packet contains a *route record*, in which is accumulated a record of hops taken by the route request packet as it is propagated through the network during this route discovery. Each RREQ packet also contains a unique *request id*, set by the initiator. To prevent the possibility of forming routing loops, each intermediate node that receives RREQ, propagates it if their address is not already included in RREQ's *Route Record* filed and appends its address to the RREQ's *Route Record* before rebroadcasting it. If a node receives a RREQ with a new *request id* it stores the hop count of this request in the *NumHopCount* variable, appends its address to the *Route Record* of RREQ, increases the hop count of RREQ and rebroadcast it. To prevent flooding network with too many RREQs, nodes only rebroadcast it if the hop count of received RREQs is less (or equal to) than *NumHopCount.* Rebroadcasting the RREQs is continued until they reach to the destination. By using this method for propagation the RREQs, many RREQs from different routes will be received to the destination.

For example in fig 2, *S* sends RREQ and *A, B* and *C* receive this RREQ with *HopCount* sets to 1. According to our proposed algorithm when node *B* rebroadcasts RREQ and *C* receives it, since the hop count of received RREQ is greater than the hop count of first RREQ, this RREQ will be dropped. By using this method 7 RREQs from different paths will be received by the destination.

Initiating RREQ by the source and handling it by intermediate nodes are shown as pseudo code format in fig. 2 and fig. 3 respectively.

When a destination receives RREQs, reverses the route in the *route record* from the received RREQs, and uses this route to send back the route reply (RREP) packet to the source. As the RREPs travel back to the source each node along the path sets up a forward pointer to the node from which the RREP came, *NextHop,* and updates its timeout information for route entries to the source. When a node receives multiple RREPs from a node, it increments the number of route reply, *CountReply,* that received from this node in its *route table* field whitch means how many routes from this next hop to the destination are exist. This process is continued until the RREPs reach the source. These two phases create multiple routes from source to the destination. As shown in fig. 2, *C* has three next hops for *D* as a destination, also *S* receives 7 RREPs from next hops and since *C* sends four RREPs to *S* the *countReply* of *C* is sets to 4.

b) *Sending Data*

When the source receives RREPs, it can transmit data packets through the discovered routes. Our protocol uses hop-by-hop method for forwarding data. Each node that receives data packets sends them to the next hops according to their *CountReply* values. Each next hop that has greater *CountReply* receives more data than the next hops that have less *CountReply*. This process causes that all of the discovered routes is used and data packets distributed across all of the paths simultaneously. For example *S* sends $^4/_7$ of data packets to *C* that means these packets are distributed to the four different routes later, also $^2/_7$ of data packets transmitted to *B* and $^1/_7$ of them to *A*. Each intermediate node that receives data packets sends them to the next hops according to their *CountReply* in their *Route Table*.

c) *Route Maintenance*

If a route is not used for some period of time, a node cannot be sure whether the route is still valid; consequently, the node removes the route from its routing table.

If data is flowing and a link break is detected, a Route Error (RERR) packet is sent to the source of the data in a hop-by-hop fashion. As the *RERR* propagates towards the source, each intermediate

DestAddr	NextHop	CountReply
D	***C***	***4***
D	***B***	***2***
D	***A***	***1***

Route Table Of ***S***

DestAddr	NextHop	CountReply
D	***I***	***2***
D	***E***	***1***
D	***F***	***1***

Route Table of ***C***

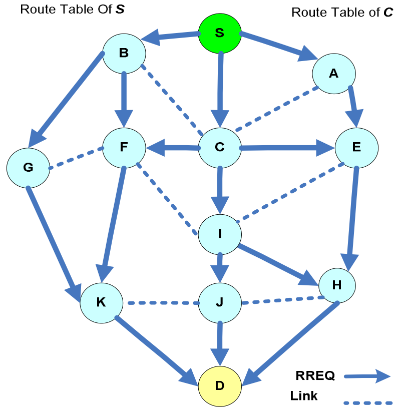

Figure 2. Propagation RREQs

```
Initiating Route Request packet:
If (Source doesn't have any route to D)
{
    rreqPkt = Create RREQ Packet;
    rreqPkt.requestId = get a Unique Id;
    rreqPkt.SourceAddress = Its Address;
    rreqPkt.TargetAddress = D Address;
    rreqPkt.hopCount = 1;
    Append Its Address to the Route Record of rreqPkt;
    Send rreqPkt to its neighbors;
}
Else
     Call Transmit_Data function;
```

Figure 3. Initiating RREQ by the source

```
Handle Route Request Packet:
If (this node is destination)
   Send Back Route Reply to the Source;
Else
{
If (Its address is not included in rreqPkt's Route Record field)
{
   If (rreqPkt.hopCount <= the first HopCount received for
                                          this rreqPkt.requestId)
   {
       rreqPkt.hopCount++;
       Append Its Address to the Route Record of rreqPkt;
       Rebroadcast rreqPkt;
   }
   Else
       Drop rreqPkt;
}
Else
   Drop rreqPkt; }
```

Figure 4. Handlling RREQ by intermediate nodes

node decrements *CountReply* by 1 which means one of the routes from this next hop to the destination is broken. When *CountReply* of each next hop in *Route Table* reaches to 0 this next hop is deleted from route table. If no entry for a destination exists in *Route Table* of source, it invalidates the route and reinitiates route discovery process if necessary.

For example in fig 2, if the link between *I* and *J* breaks *I* sends a *RERR* to the *C*, when *C* receives this packet it changes the *CountReply* of next hop *I* in *Route Table* to 1 and forwards this packet to *S*, also when *S* receives this packet changes the *CountReply* of *C* to 3.

IV. Performance Evaluation

a) Simulation Environment

The simulation code was implemented within the Global Mobile Simulation (GloMoSim) library [10]. In the simulation, we modeled a network of 50 mobile hosts placed randomly within a $1000\times1000m^2$ area. Radio propagation range for each node was 200 meters and channel capacity was 2Mbit/sec. Each simulation runs for 600 seconds of simulation time. The IEEE 802.11 Distributed Coordination Function was used as the medium access control protocol. We used Constant Bit Rate as our traffic and Random-Way point as mobility model. The size of data payload was 512 bytes.

b) Performance metrics

To evaluate the performance of LBAODV, we simulated and compared the following schemes:

- AODV
- AOMDV
- LBAODV

We evaluated these schemes as a function of mobility speed of nodes and the traffic of sessions. The number of sessions was set to 1 and mobility speed of nodes was varied from 0 to 75 Km/hr with pause-time that is equal to 3 seconds. To evaluate the performance of our protocol, each source sent data as a rate of 100Kbit/s to 600Kbit/s. we evaluated the following metrics for each session:

- *Packet Delivery Ratio*: The number of data packets received by destinations over the number of data packets sent by the source.
- *Routing Overhead:* The number of all packets (data and control packets) transmitted divided by the number of data packet delivered to the destination.
- *Average of End-To-End Delay*: The end-to-end delay of a packet is defined as the time a packet takes to travel from the source to the destination.
- *Energy Consumption*: This parameter is measured for each node.

c) Simulation Results

c.1) Packet Delivery Ratio

The packet delivery ratios as a function of mobility speed and throughput are shown in figure 5 and 6. In figure 5 data rate is set to 300Kbit/s. We can observe that as speed increases because of links break the packet delivery ratios decrease in AODV, AOMDV and LBAODV protocols. Since in LBAODV, all discovered routes are used simultaneously for transmitting data, large amount of data packets will receive to the destination in compare of AODV and AOMDV that use only one route for transmitting data. In figure 6 mobility speed is set to 30Km/hr. In AODV as throughput increases the packet delivery ratio is decreased dramatically that in 600Kbit/s the packet delivery ratio in AODV is about 55% but since LBAODV uses multiple paths simultaneously the packet delivery ratio is about 80%.

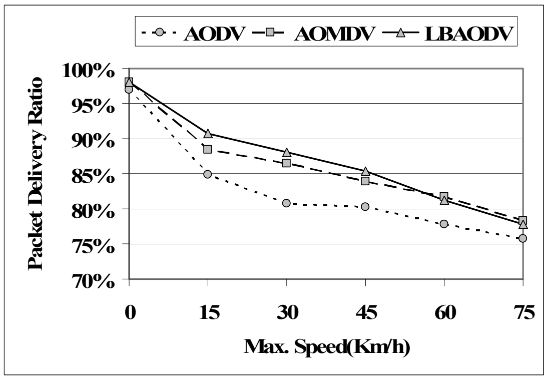

Figure 5. Packet Delivery Ratio Vs Mobility (Data Rate=300Kbit/s)

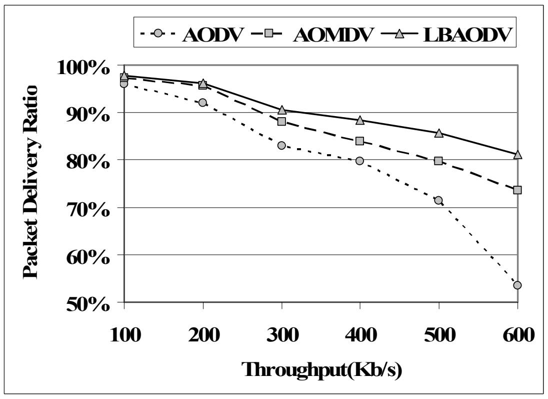

Figure 6. Packet Delivery Ratio Vs Data Rate (Max Speed=30Km/hr)

C.2) Number of total packets transmitted per data packet delivered

Figures 7 and 8 show the routing overhead as a function of mobility speed and throughput. In figure 7 data rate is set to 300Kbit/s and in figure 8 mobility speed is set to 30Km/hr. As shown in figure 7 as speed increases, because of link break and route reconstruction the routing overhead in AODV, AOMDV and LBAODV are increased. Since LBAODV transmits many RREQs and RREPs in compare to AODV and MAODV thus its routing overhead is higher than AODV and AOMDV.

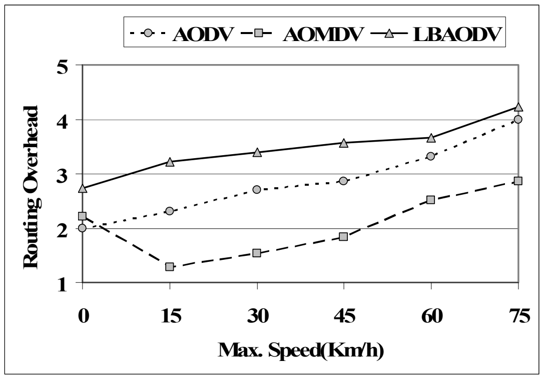

Figure 7. Number of total packets transmitted per data packet delivered Vs. Mobility (Data Rate=300Kbit/s)

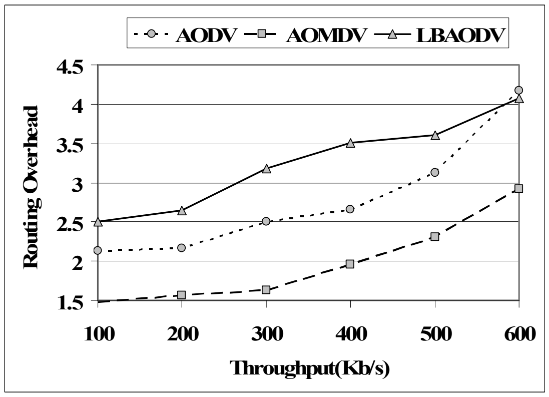

Figure 8. Number of total packets transmitted per data packet delivered Vs. Data Rate (Max Speed=30Km/hr)

C.3) Average of End-To-End Delay

Figures 9 and 10 show the average end to end delay as a function of mobility and data rate. In figure 9 data rate is set to 300Kbit/s and in figure 10 mobility speed is set to 30Km/hr. since LBAODV uses all discovered routes simultaneously its end-to-end delay is less than AODV, also LBAODV may find some longer path than AOMDV, thus its end-to-end delay is greater than AOMDV.

C.3) Energy Consumption

In figure 11, 12 and 13 energy consumption are shown as a function of mobility speed and data rate. In LBAODV all discovered routes are used simultaneously thus many of nodes participate in forwarding data and their consumed energies are increased simultaneously, but in AODV and AOMDV only one route is used for forwarding data thus only a few nodes are involved in transmitting data. In fact, LBAODV distributes energy consumption across many nodes thus the life time of network is increased but in AODV and AOMDV the energy consumption is focused on a few nodes so the network life time is less than in LBAODV.

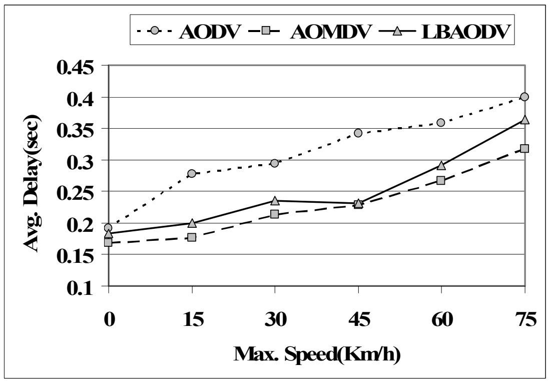

Figure 9. Average End-to-End Delay Vs. Mobility (Data Rate=300Kbit/s)

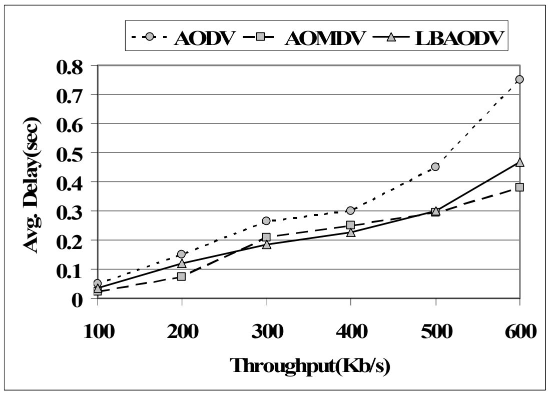

Figure 10. average End-to-End Delay Vs. Data Rate (Max Speed=30Km/hr)

V. Conclusion and future work

We introduced a new multipath routing that uses discovered paths simultaneously; this technique applied to AODV and evaluated via several simulations scenarios. Simulation results show that our protocol have better packet delivery ratio in compare of AODV and AOMDV, also the energy consumption is distributed across many nodes that cases the network life time in LBAODV is better than AODV and AOMDV. We are going to examine our work over other routing protocols such as DSR and evaluate its performance with different scenarios.

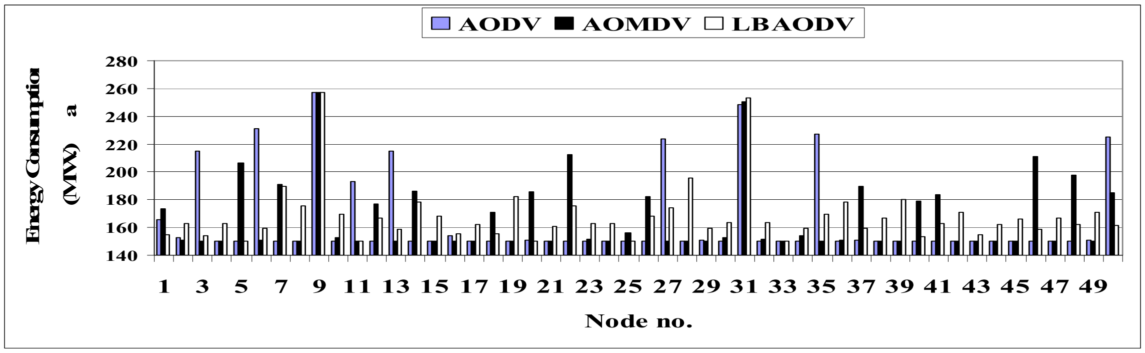

Figure 11. Power Consumption
(Max Speed=15Km/hr,Data Rate=200Kbit/s)

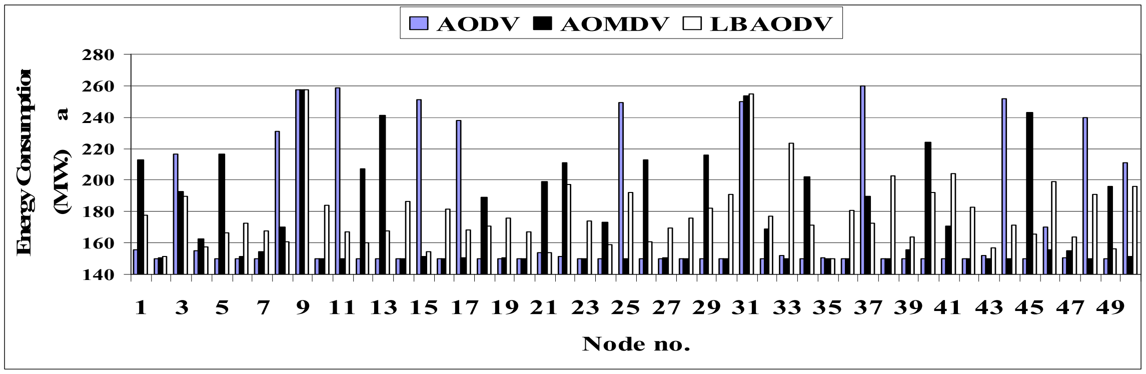

Figure 12. Power Consumption
(Max Speed=15Km/hr,Data Rate=400Kbit/s)

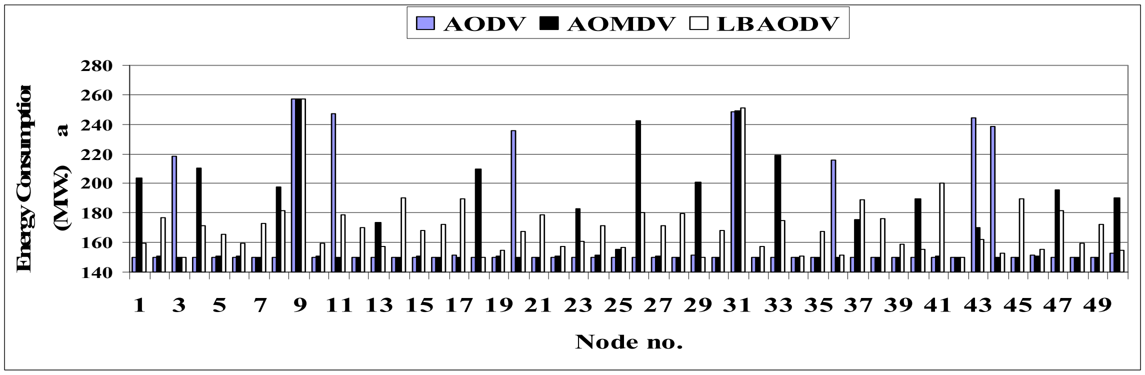

Figure 13. Power Consumption
(Max Speed=30Km/hr,Data Rate=200Kbit/s)

References

[1] 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,” Proceedings of ACM/IEEE MOBICOM ’98, Dallas, TX, Oct.1998, pp.85-97.

[2] Cho, H.K., Kim, E.S., Kang, D.W.”A load-balancing routing considering power conservation in wireless ad hoc networks.”, Proc. of 16th International Workshop on Database and Expert Systems Applications. (2005) 128–132

[3] Park, S., Shin, J., Baek, S., Kim, S.C.,”AODV-Based Routing Protocol Considering Energy and Traffic in Ad Hoc Networks.” In Proc. of International Conference on Wireless Networks. (2003) 356–361.

[4] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad hoc Networks,” Proceedings of ACM/IEEE MOBICOM'99, Seattle, WA, Aug. 1999, pp. 195-206.

[5] S.-J. Lee and M. Gerla, “AODV-BR: Backup Routing in Ad hoc Networks,” Proceedings of IEEE WCNC 2000, Chicago, IL, Sep. 2000.

[6] A. Nasipuri and S.R. Das, “On-Demand Multipath Routing for Mobile Ad Hoc Networks,” Proceedings of IEEE ICCCN'99, Boston, MA, Oct.1999, pp. 64-70.

[7] V.D. Park and M.S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” Proceedings of IEEE INFOCOM' 97, Kobe, Japan, Apr. 1997, pp. 1405-1413.

[8] C. E. Perkins, E. M. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing”, RFC 3561, July 2003.

[9] C. E. Perkins and E. M. Royer, “The Ad hoc On-Demand Distance Vector Protocol”, In C. E. Perkins, editor, Ad hoc Networking, pages 173.219. Addison-Wesley, 2000.

[10] UCLA Parallel Computing Laboratory, GloMoSim: A scalable simulation environment for wireless and wired network system.