A Load Balanced Aware Routing Protocol For Wireless Ad Hoc Networks

Vahid Nazari Talooki, Jonathan Rodriguez, Rasool Sadeghi

***Abstract*—Nodes in ad hoc networks can be unfairly burdened to support many packet-relaying functions, resulting in excessive loads on these hot spots. This load on nodes appears in two major aspects: traffic and power consumption. Unbalanced traffic may lead to more delay, packet dropping, and decreasing packet delivery ratio (PDR). Unbalanced energy consumption leads to node failure, network partitioning and decreases network lifetime and route reliability. Existing approaches try to improve the performance of routing protocols with respect to traffic balancing or energy consumption balancing, but most lead to drawbacks such as more delay, blocking, or dependence on global information from all nodes. In this paper we improve the well known Dynamic Source Routing (DSR) protocol to the so called Load Balanced DSR (LBDSR) protocol. We modify the RREQ (Route Request) and RREP (Route Reply) messages in DSR in order to maintain the remaining energy of intermediate nodes which forward RREQ and RREP. Route structure, available in the nodes cache, is modified so that the remaining energy of nodes can be calculated. LBDSR shows better traffic balancing and energy consumption balancing, end-to-end delay and route reliability metrics than DSR. Furthermore, LBDSR can also be customized to achieve better performance with respect to each of these metrics instead of being a trade-off between them.**

***Index Terms*— Ad hoc Networks, DSR, Load Balancing, Routing Protocols.**

I. INTRODUCTION

NODES in ad hoc networks can not only be a sender or receiver in a connection, but they can also be responsible for forwarding packets to neighboring nodes to implement the overall mechanism of routing. This approach will make ad hoc networks independent from a central point, but may lead to some particular mobile nodes being unfairly burdened to support many packet-relaying functions and consequently, loading on these hot spots. This load on nodes appears in two major aspects: traffic and power consumption. Load Balanced DSR (LBDSR) protocol tries to balance this load.

II. DESCRIPTION OF PROTOCOLS

A. Dynamic Source Routing (DSR)

DSR is a source routing protocol, and requires the sender to know the complete route to the destination. It is based on two main processes: (a) the route discovery process which is based on flooding and is used to dynamically discover new routes and maintain them in the nodes cache, (b) the route maintenance process, which periodically detects and notifies of network topology changes. Discovered routes will be cached in the relative nodes [1]. We use the latest version [2] of DSR for our study.

When a source node receives the first arriving RREP in response of RREQ for a destination, the source node uses the first arrived RREP and ignores all other arriving RREP. It is a logical assumption that the first arriving RREP has traversed the shortest route. Furthermore, by this approach, DSR is non-blocking because every participating mobile node does not have to wait indefinitely- or for a threshold which must be determined very carefully- for all RREP to arrive. This approach tries to reduce delay and to avoid blocking.

We modified this approach: if a source node has no route to a destination in its cache, it will initialize a RREQ for the first time and then can react with first arriving RREP. While DSR will still remain non-blocking, it doesn't ignore further arriving RREPs and will extract routes from them and cache these routes. By these new routes, the source node has not only more routes to the destination for future use (for example when current route fails and we need an alternate route for retransmission) but can also extract new routes to other destinations from these routes.

B. Previous improvements on DSR

IEEE 802.11 requires an RTS/CTS/Data/ACK exchange for all unicast packets. Hence data packets can be transmitted only through bidirectional routes.

Additionally, nodes are modified to work in promiscuous mode. In this mode each node overhears packets, even if it is not listed in the source route. By this approach nodes learn about route failures by tapping route error packets. Moreover, if a node overhears a packet which has its own address listed in the unprocessed portion of the source route, it implies that the node is set to receive the packet through a longer route. In that case, it can let the source of that packet know about the available shorter route by sending a gratuitous route reply.

Another improvement is that when a node forwarding a

Vahid Nazari Talooki , researcher at the Institute of Telecommunication (IT), Aveiro, Portugal, (e-mail: vahid@av.it.pt)
Jonathan Rodriguez, Senior Researcher at the Institute of Telecommunication (IT), Aveiro, Portugal, (e-mail: jonathan@av.it.pt)
Rasool Sadeghi , researcher at the Institute of Telecommunication (IT), Aveiro,Portugal,(phone:00351-962906851; fax: 0035-218418472; e-mail: rsadeghi@av.it.pt)

packet to its next hop discovers that the node is unreachable, it consults its cache to find an alternate route to the destination. If the node has another route to the destination, it changes the source route appropriately and forwards the packet according to this new route [3].

We use this latest version of DSR which is available in GloMoSim2.02 for our study.

C. *Ad Hoc On-demand Distance Vector (AODV)*

AODV is a combination of the DSR and the DSDV protocols. It uses the on-demand mechanism of route discovery and route maintenance from DSR and the hop-by-hop routing and sequence number from DSDV. Per each destination, AODV creates a routing table like DSDV, while DSR uses node cache to maintain routing information [1].

III. Load Balanced Dynamic Source Routing (LBDSR)

A. *Modifying control messages*

An energy field in the form of an array like the address array field to RREQ and RREP which contains the remaining power (battery energy) of each node that RREQ or RREP is forwarded by is added. Hence when a node forwards a RREQ or RREP, it will append its remaining power to the end of the energy array field.

B. *Modifying routes table in nodes caches*

A new field which maintains remaining energy of all intermediate nodes is added. The amount of this array field can be extracted from modified RREQ and RREP messages.

C. *Modifying the route selection process*

We define a *Route Priority Function* which determines the priority of each route. For a specific source and destination pair, the route which has the maximum priority will be selected as the candidate route.

Assume that for a specific (source, destination) pair of nodes in a network we have N routes like route i ordered by time (first route is the freshest route and Nth route is the oldest one) which are available in the source node's cache. All input parameters are normalized in range [0…1] to make them comparable. Route Priority Function has four parameters with respect to each route: *length, measure of freshness, traffic and remained energy level*.

Length, *L(i)* in eq.(1) indicates the priority of route i with respect to the length of the route:

$$L(i) = \frac{Actual\_Length\_Route(i)}{Max\_Length} \tag{1}$$

Actual_Length_Route(i) is the actual length of route i (i.e number of hops in route i) and *Max_Length* is the maximum length that a route can take in DSR routing protocols.

Measure of freshness, which relates the priority of route i with respect to the measure of freshness, is defined below:

$$F(i) = \frac{N - i + 1}{N} \tag{2}$$

Traffic, A route with a lower traffic cost has a higher priority in the routing process. The total traffic which can affect a route can be caused by traffic of the route's nodes. Traffic of each node is related to the number of routes through it. *TT(i)* which indicates total traffic of route i is defined in eq. (3):

$$TT(i) = \sum_{j=1}^{M_i} N_j \tag{3}$$

Where Nj is the number of routes through node j and M_i is the number of nodes in route i. Transmitters in ad hoc networks use radio signals for communication and this communication is limited within a certain transmission. Only one transmission channel is used within each range and covering the entire available bandwidth. To transmit data, mobiles which are in the same range must sense for other transmission at first, and then if no other node is currently transmitting, it can gain access permit and transmission. Hence packet delay is not caused only from traffic load at the current node, but can also be caused by traffic load at neighboring nodes [4]. Therefore *TT(i)* can be modified below:

$$TT(i) = \sum_{j=1}^{M_i} N_j + \sum_{k=1}^{G_j} R_j^k \tag{4}$$

Where G_j is the number of neighbors for node j and R^k_j is the number of routes through kth neighbor of node j. Since the maximum connection that a node can establish is limited, a node with a high traffic load can turn into a bottleneck for the route. So the next input parameter for the *RoutePriority* function can be defined as follows:

$$T(i) = \frac{TT(i)}{NE_i \times Max.Con} \times \frac{MT(i)}{Max.Con} = \frac{TT(i) \times MT(i)}{NE_i \times Max.Con^2} \tag{5}$$

Where NE_i is the number of neighbors of nodes in route i which has M_i nodes (repetitive neighbor is taken into account once), *Max.Con* is the maximum connection which a node can establish in a network (which is set to the same value for all nodes), and *MT(i)* is the maximum traffic load on a node between nodes of the route i (i.e maximum N_j in equation 5).

In eq. (5), *MT*(i) is an agent for drawback of the node which is turned to a bottleneck for the route because of its high traffic load. But *MT(i)* is a part of *TT(i)* and, consequently is taken into account twice. Hence eq. (5) can be modified below:

$$T(i) = \frac{(TT(i) - MT(i)) \times MT(i)}{NE_i \times Max.Con^2} \tag{6}$$

Remaining energy level, The remaining battery power of a route's nodes can be an important metric in the selection of that route. Thus a route with a maximum remaining battery power in its nodes can be an ideal route with respect to energy level. Moreover, despite the fact that sometimes a route may have a high remaining energy level, it may have some nodes with very low energy levels too. There is a high probability of failure in these hot spots which makes these nodes undesirable.

In eq. (7), The *E(i)* indicates the priority of route i with respect to the remaining energy level:

$$E(i)=\frac{RE(i)}{M_i \times InitialEnergy}\times\frac{MRE(i)}{InitialEnergy} \quad (7)$$

$$=\frac{RE(i)\times MRE(i)}{M_i \times InitialEnergy^2}$$

Where M_i is the number of nodes in the route, *InitialEnergy* is the node's energy at the beginning of simulation (which is set to the same value for all nodes), *RE(i)* is the total of the remaining energy in the nodes of route i, and *MRE(i)* is the minimum of the remaining energy between all nodes of route i. In the above equation, *MRE* is an agent for the drawback of the node which is turned into a bottleneck for the route because of its low energy level. But *MRE(i)* is a part of *RE*(i) and is taken into account twice. Hence equation 7 can be modified as shown below:

$$E(i)=\frac{(RE(i)-MRE(i))\times MRE(i)}{M_i \times InitialEnergy^2} \quad (8)$$

As time goes by, the power consumption of each node will increase. Hence for some routes which were discovered a long time ago, the remaining energy array field of these routes may not be sufficiently fresh and sometimes can result in an invalid decision made based on these values. An improvement can be applied to this approach. If a node of a route in the routing table appears intermediate node, it can have diverse values for the remaining energy in each of these routes, which are discovered at different times. The minimum value of this different remaining energy for a node is fresher and can be a valid value. More importantly, as time passes, the consumed energy of the node increases and more consumed energy leads to less remaining energy, this is why the minimum value is fresher. With this proposed approach, we can update the value of the remaining energy of each node in our cache prior to the route selection process. This allows us to compute the total remaining energy of each route's node as the remaining energy level of that route, and hence facilitate efficient decisions.

***RoutePriority* Function,** When searching for a routing path, the *RoutePriority* function is computed for all available routes to the destination, which are present in the source node's cache, and the route with maximum *RoutePriority* will be selected. Hence the *RoutePriority* function is given by:

$$Route\Pr iority(i)=\frac{K_F \times F(i)+K_E \times E(i)}{K_T \times T(i)+K_L \times L(i)} \quad (9)$$

Where K_F, K_E, K_T, and K_L are coefficients of freshness, energy, traffic, and length of route respectively. Desirable values of these coefficients obtained by simulation make a quadruple for length, freshness, traffic, and energy parameter.

Other alternative functions can be also used without impacting the generality of the proposed approach. Based on a variety of workloads and scenarios for simulation, the coefficient of parameters can be determined to be a trade-off for improving end-to-end delay, traffic balancing, and power consumption balancing all together.

Note that we can achieve higher LBDSR performance with regard to end-to-end delay ($LBDSR_d$), Traffic ($LBDSR_t$), or Energy consumption ($LBDSR_e$) by adjusting the weighting coefficient K_F, K_E, K_T, and K_L respectively in (9) .

IV. METRICS

Comparisons will be perform based on five metrics: I) Average end_to_end delay, II) Normalized Routing Load (NRL) which is the number of routing packets transmitted per data packet delivered successfully at the destination, III) Failure degree which is the number of nodes which have failed because of low battery power, IV) Traffic balancing and V) Energy consumption balancing. For Traffic balancing metric, first we compute Traffic Load, TL(i) for each node i in the network.

$$TL(i)=\frac{\sum_{j=1}^{M_i} Packet\_size_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M_i} Packet\_size_{ij}} \quad (10)$$

Where $Packet\_size_{ij}$ is the size of packet j forwarded by node i, M_i is the number of packets that node i have forwarded, and N is the number of the network nodes. Then we can compute the deviation of these load values which will be our metric for the traffic load balancing capability of the protocol. The smaller the deviation means better traffic load balancing. For energy consumption balancing, first we compute Energy Load, EL(i) for each node i in the network.

Deviation of these loads will be our metric for the energy consumption balancing capability of the protocol. The smaller the deviation, the better the energy consumption balancing.

$$EL(i)=\frac{ConsumedEnergy(i)}{TotalConsumedEnergy} \quad (11)$$

V. METHODOLOGY

Nodes in the simulation move according to a model that we call "random waypoint" model and all settings are similar to [5]. The movement scenario files we used for each simulation are categorized into two different groups based on two parameters: paused time, maximum node mobility and simulation time. We used the GloMoSim 2.02 simulator [6], which is a scalable simulation environment for wireless based on Parsec [7]. GloMoSim can support a wide range of ad hoc routing protocols, as well as a realistic physical layer like DSR and AODV.

VI. SIMULATION RESULTS

A comparison between DSR and AODV can be found in many papers [1] so we will focus on LBDS performance in this section.

A. *Average end_to_end delay*

In LBDSR all routes to a destination which are available in the source node's cache must be evaluated by running the *RoutePriority* function for all of them, which leads to an increase in route discovery process time (although this process happens once only in the beginning of a continuous transition). Despite the fact that route discovery latency is higher in LBDSR compared to DSR, LBDSR considers the traffic of routes and has superiority to DSR.

When LBDSR is customized to LBDSR$_d$ by giving a higher coefficient to traffic and the length of the route (which can decrease transfer time), it shows a better performance with respect to end-to-end delay, obviously up to near 20% fewer deviation than DSR by varying pause time or mobility (Fig. 1, 2).

B. *Normalized Routing Load*

In LBDSR, RREQ and RREP control messages have an extra field which is an array of the remaining energy of nodes in the current path. Hence routing in LBDSR causes more traffic loads compared to DSR. Although this extra load is small it can lead to greater NRL.

On other hand, nodes with a high traffic load drop RREQ due to their full queue buffer, which result in more RREQ being produced. In the network, LBDSR can better handle this event by providing enhanced traffic load balancing. Additionally, sometimes a link in a route which ends in an intermediate node fails because of node battery failure. LBDSR can decrease node battery failure by facilitating enhanced energy consumption balancing.

Considering the aspects in some special scenarios (based on the movement and communication model of the nodes within the network), LBDSR can decrease NRL, while at other times it shows more NRL. This situation leads to fluctuation in the LBDSR curve with respect to NRL (Fig. 3, 4).

C. *Traffic balancing*

Increasing pause time leads to fewer network topology changes and so the selected paths by the protocols will be more stable. Therefore, protocols have a greater tendency to use particular nodes (through these paths). This, in turn, leads to more loads on these nodes and consequently, more deviation of the network's load (Figure 5).

With a higher mobility ratio, more links will be disconnected and the stability of the paths will decrease. This pattern results in decreasing the delivery ratio and as a result, fewer sample points. This in turn, leads to a decrease in the deviation of the network's load (Figure 6).

As shown in figures 5 and 6, LBDSR has a higher performance than DSR with regard to traffic balancing (in other words LBDSR has a lower deviation in traffic load), and the deviation of traffic load is 15% and 10% lower than DSR by varying the pause time and mobility, respectively .When we customize LBDSR by giving a higher weight to the traffic of routes and obtain LBDSR$_t$, a higher performance can be achieved and LBDSR$_t$ shows a nearly 25% lower deviation than DSR by varying pause time or mobility.

D. *Energy consumption balancing*

A scenario similar to the previous one happens with respect to energy consumption balancing, which leads to greater deviation with a higher pause time, and reduced deviation due to the effects of mobility.

Figure 7 and 8 show the performance of the protocols with regard to average energy consumption balancing by varying pause time and mobility respectively.

TABLE I
MOVEMENT MODEL I PARAMETERS, CHARACTERIZED BY PAUSED TIME

Parameters	Values
topology area	1500m x 300m
maximum of nodes	10 m/s
paused time	0..200
number of nodes	50
simulation time	200s

TABLE II
MOVEMENT MODEL II PARAMETERS, CHARACTERIZED BY MAXIMUM NODE MOBILITY

Parameters	Values
topology area	1500m x 300m
maximum mobility of nodes	0 m/s…40 m/s
paused time	50 s
number of nodes	50
simulation time	200s

TABLE III
MOVEMENT MODEL III PARAMETERS, CHARACTERIZED BY *SIMULATION TIME*

Parameters	Values
topology area	1500m x 300m
maximum mobility of nodes	10 m/s
paused time	50 s
number of nodes	50
simulation time	300s…2800s

TABLE IV
PARAMETERS OF COMMUNICATION MODEL.

Parameters	Values
Traffic sources	CBR
Data packets size	512 bytes
Sending rate	8 packets/second
Maximum connection	10

As shown in figures 7 and 8, LBDSR has a higher performance than DSR regarding energy consumption balancing and shows lower deviation in energy consumption. Deviation of energy consumption is 15% and 10% lower than DSR by varying pause time and mobility respectively. By customizing LBDSR into LBDSR$_e$ by giving a higher weight to the remaining energy of routes, a higher performance can be achieved. Figure 7 and 8 shows LBDSR$_e$ has 25% less deviation than DSR by varying pause time or mobility.

E. *Node's failure degree*

A route which is being used can become invalid due to two major aspects: i) link breakage and ii) node's failure. An energy hungry node can conserve its battery power by not forwarding data packets on behalf of others. This approach can decrease the node's failure degree, and consequently increase the route's reliability. A protocol which can provide a greater balance in energy consumption can decrease the node's failure in the network. As a consequence, we perform our simulation not only for LBDSR, but also for LBDSR$_e$ (which is a version of LBDSR customized with respect to balancing in energy consumption).

Results in this section are obtained by performing simulation with regard to movement model III. In this model simulation time varies from 300 up to 2400.

Figure 9 shows protocols performance with respect to the node's failure degree. The probability of failure in a node will increase if simulation takes a long time, hence all curves in this figure have an ascending shape. As shown in Figure 9, LBDSR and especially LBDSR$_e$, show better performance because of their lower node failure degree.

VII. CONCLUSION AND FUTURE WORKS

We modified the control packets, routing tables, and the route selection method resulting in enhanced LBDSR performance with respect to traffic load balancing, energy consumption balancing, average end-to-end delay, and route's reliability metrics. LBDSR shows a 15% improvement in balancing (traffic and energy consumption), 10% reduction in average end-to-end delay, and a decrease in the network node failure rate resulting in better route reliability. Additionally, LBDSR can be customized to focus on one of these metrics. In this case LBDSR shows an improvement of up to 25% with respect to these metrics.

The proposed approach can be applied to existing routing protocols, especially for reactive protocols such as AODV to provide better load balancing in ad hoc networks.

ACKNOWLEDGMENT

This work was carried out in the scope of the PEACE project that is supported by the European Commission in the framework of FP7 ICT-SEC-2007 with contract number 225654.

REFERENCES

[1] Nazari,V. and Ziarati, K., "Performance Comparison of Routing Protocols for Mobile Ad Hoc Networks," IEEE Conf., Asia-Pacific Conference on Communications, Busan, Korea, 2006 pp. 1-5.
[2] D.B. Johnson, D.A. Maltz, Y.-C. Hu, The dynamic source routing protocol for mobile ad hoc networks (DSR).Available from: <http://www.ietf.org/internet-drafts/draftietf-manet-dsr-09.txt>.
[3] G. Chakrabarti, S.Kulkarni," Load Balancing and resource reservation in mobile ad hoc networks", Ad Hoc Networks, Volume 4, Issue 2, 1 March 2006, Pages 186-203.
[4] K. Woo and B. Lee, "Non blocking localized routing algorithm for balanced energy consumption in mobile ad hoc networks," Proc. IEEE MASCOTS'01, 2001, pp.15–18.
[5] C.E. Perkins, E.M. Belding-Royer, S.R. Das, "Ad hoc on demand distance vector (AODV) routing. Available from:routing.
[6] Available from: <http://pcl.cs.ucla.edu/projects/glomosim/ >
[7] Available from: <http://pcl.cs.ucla.edu/projects/parsec/>

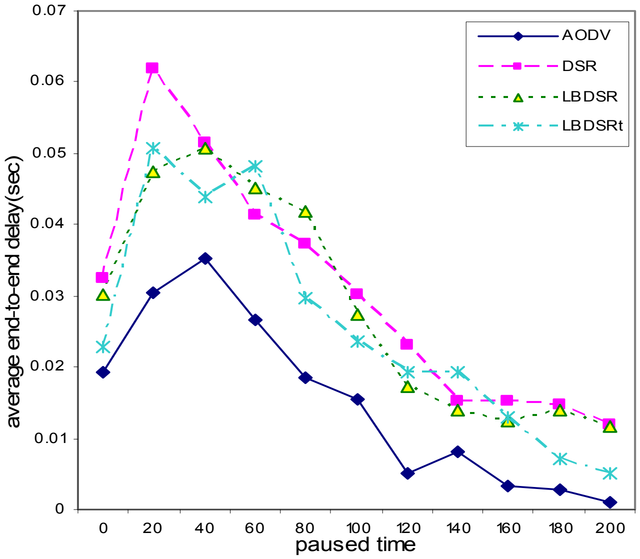

Fig. 1. Delay vs. paused time

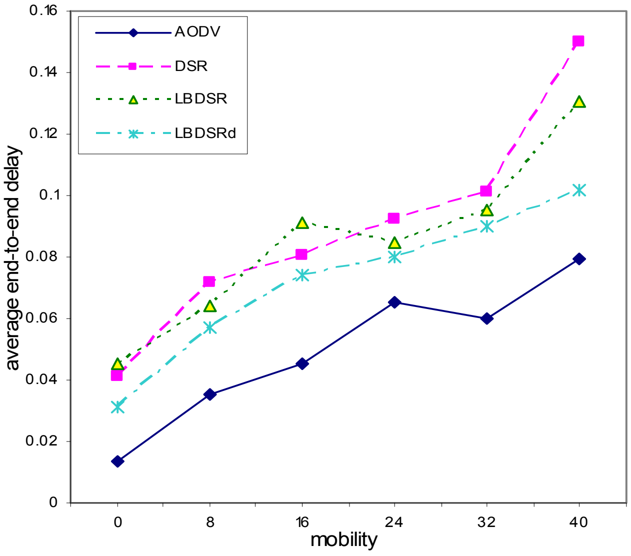

Fig. 2. Delay vs. mobility

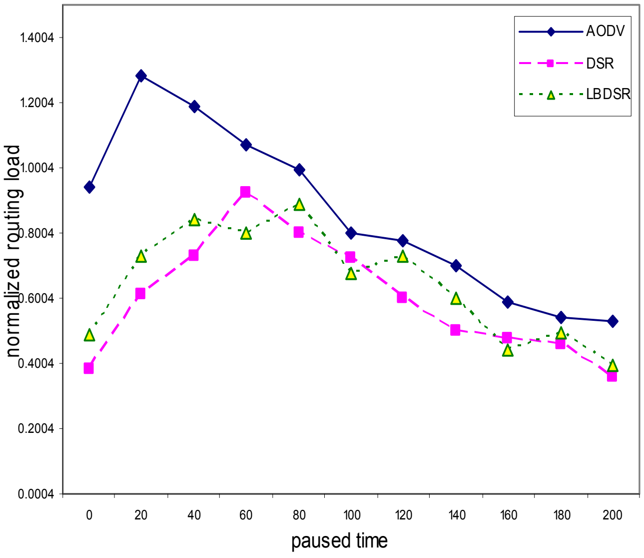

Fig. 3. NRL vs. pause time

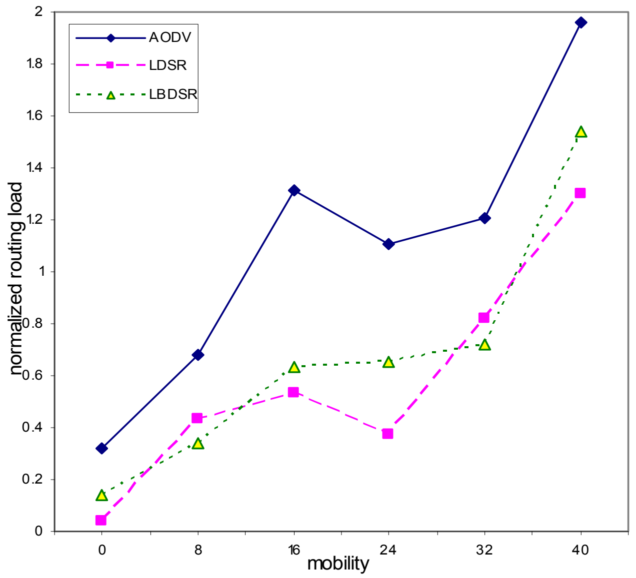

Fig. 4. NRL vs. mobility

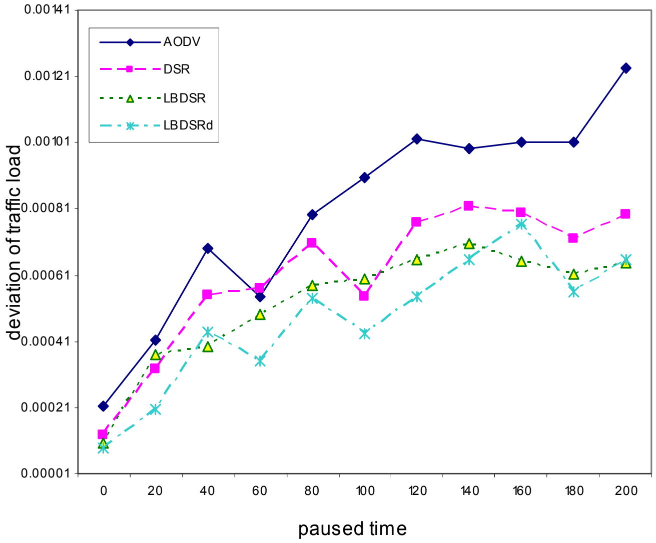

Fig. 5. Traffic balancing vs. paused time

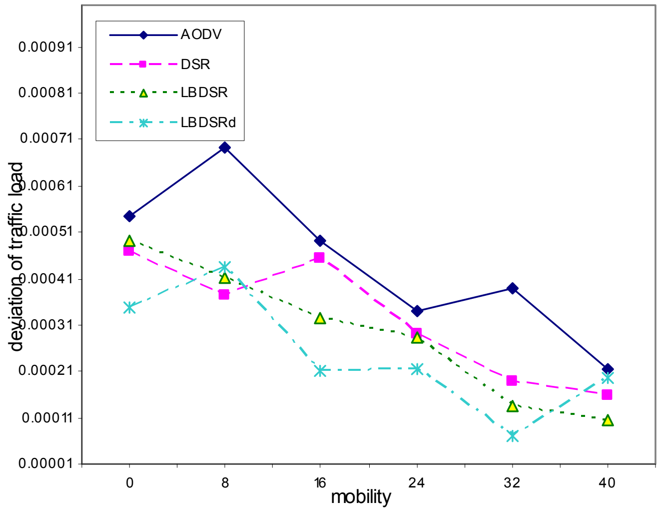

Fig. 6. Traffic balancing vs. mobility

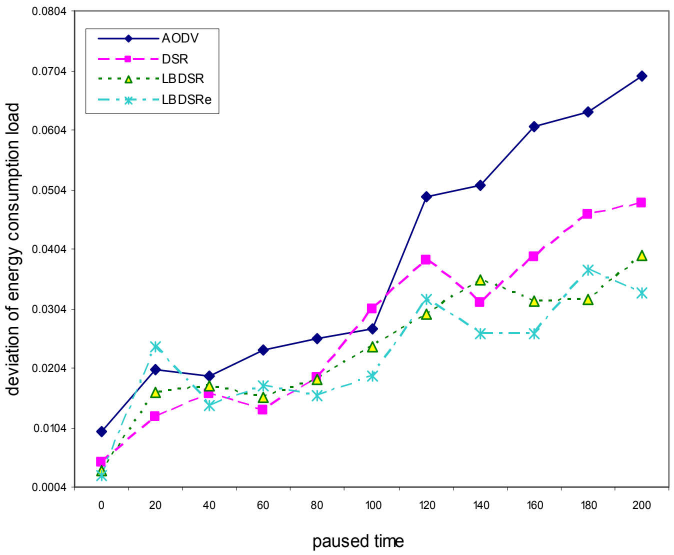

Fig. 7. Energy consumption balancing vs. paused time

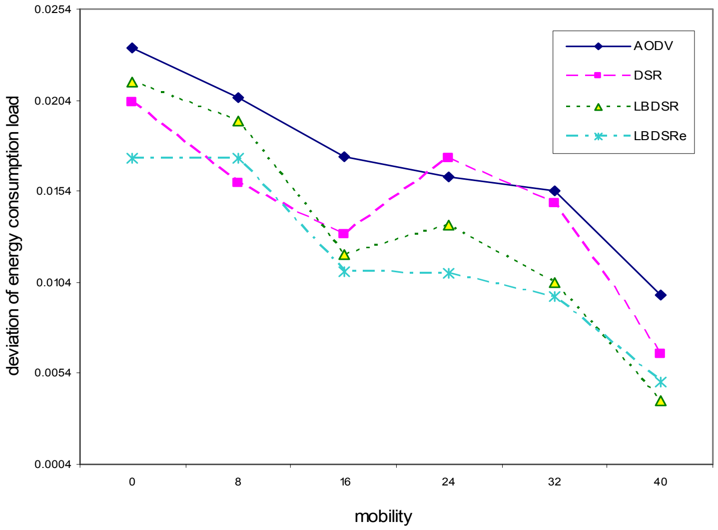

Fig. 8. Energy consumption balancing vs. mobility

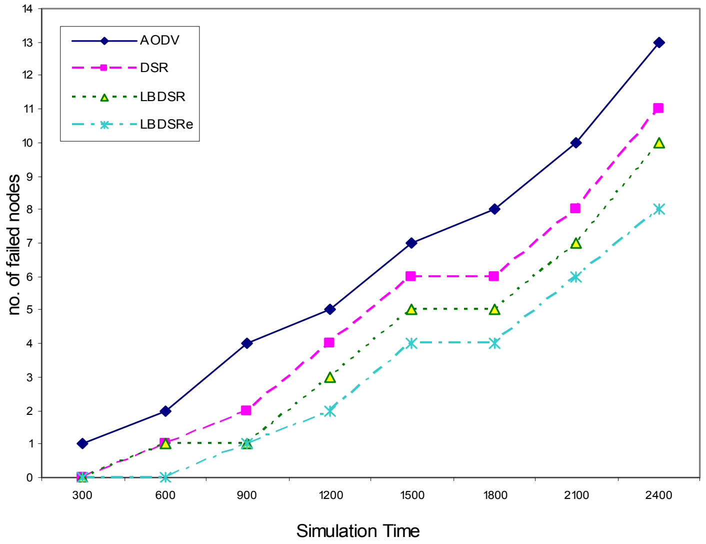

Fig. 9. Number of failed nodes vs. simulation time