Efficient Multipath Routing Protocol based on Path Survivability Factor

Hiba Afreen
Department of Computer Science & Engineering UTU, Dehradun, 248001, U.K. (India)

A K Daniel
Department of Computer Science & Engineering MMMUT, Gorakhpur, -273010, U.P. (India),

Pooja Chaturvedi
Department of Computer Science & Engineering MMMUT, Gorakhpur-273010, U.P. (India),

ABSTRACT
Mobile ad hoc network is a collection of mobile devices which can communicate through wireless links. The task of routing protocol is to send packets from source to destination. This is particularly hard in mobile ad hoc networks due to the mobility of the network elements and lack of centralized control, infrastructure, etc. In this paper, we propose a routing algorithm for the mobile ad hoc networks to discover an optimal route for transmitting data packets from source to destination. This protocol helps every node in MANET to choose next efficient successor node on the basis of channel parameters like noise, energy, bandwidth, number of hop, traffic load. The protocol improves the performance of a route by increasing network life time, reducing link failure and selecting best node for forwarding the data packet to next node. The protocol adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently.

Keywords
Alert Packet, Delay Time, Stream Array, Survivability Factor, Update Packet , ideal packet and Route Request Packet.

1. INTRODUCTION
A **mobile ad hoc network** (**MANET**), sometimes called a mobile mesh network, is a self-configuring network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Routing is the process of directing packets from source to destination. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which it forwards the packet. Source routing has been used in a number of contexts for routing in wired networks, using either statically or dynamically constructed routes. The protocol described here is explicitly designed for use in the wireless environment. When a host needs a route to another host, it dynamically determines the route based on cached information and thus results in a path discovery protocol. Our protocol uses no periodic routing in an ad hoc network. In this approach, the source routing is done dynamically using a procedure called path discovery. Whenever a node wants to send packet to another node, the sender initiates the path discovery process. Each node maintains a path cache to store the routes. As the nodes in an ad hoc network move from a place to another due to mobility of the network, some of the existing links may break and the routes from source to destination in the network are modified. This is done using a procedure called path maintenance. The proposed protocol used following stages for selecting the best route from source to destination which are: *a) path discovery b) path construction, and c) path maintenance.* The best possible selection of the path formation is obtained on the basis of factors like: *load, energy, noise, bandwidth, and number of hop counts* which are survival factors in our proposed protocol. They help a mobile ad-hoc network to choose successive node. The rest of the paper is organized as follows: In section 2, the related work is discussed. In section 3, the proposed protocol is described, which is followed by the simulation results and conclusion in the sections 4 and 5 respectively

2. RELATED WORK
Routing involves two basic activities: a) determining optimal routing paths and b) transporting information groups (typically called packets) through an internetwork. Routing is mainly classified into static routing and dynamic routing. Static routing refers to the routing strategy being stated manually or statically, in the router. In Static routing, the routing table is maintained by network administrator. Dynamic routing refers to the routing mechanism that is being knowledgeable by an interior or exterior routing protocol. Dynamic routing protocols are classified depending on how the routers communicate amongst each other and how the information kept in to their routing tables.

Most of the network protocols come under any one of the two categories: a) link state protocol and b) distance vector routing protocol. In link state protocols, a router provides the information about the topology of the network, but it doesn't provide the information about the destination (OSPF) [1]. In the distance vector routing protocol, routers send the information first, as to how far [I. distance], and in what direction [II. vector] the destination is (IGRP) [2]. Distance-vector routing protocols have less computational complexity and message overhead. Different types of routing protocol exist like Proactive, Reactive and Hybrid. Proactive routing protocols are table driven and their limitations are: Amount of data for maintenance and slow reaction in restructuring and failures. Reactive routing protocols find a route on demand by flooding the network with route request packets and their limitations are: High latency in route discovery. Hybrid routing protocols are combination of proactive and reactive routing. The routing is initially established with some proactive routes and then serves as the reactive routing protocol. The alternative to a periodic routing protocol is one that operates in an on-demand fashion.

Dynamic Source Routing Protocol (DSR)
The Dynamic Source Routing (DSR) [3] protocol is an "on-demand" source routing protocol. Mobile nodes are required to maintain route caches that contain active routes and are continually updated as new routes are erudite. The protocol

consists of two major mechanisms: a) route discovery and b) route maintenance.

Route Discovery is done by the source if it doesn't find any route for the destination in its route cache. In that case, it broadcasts a RREQ [Route Request] packet to all the neighbors, and then every neighbour node that receives the RREQ packet, broadcasts RREQ to their neighbour, and it continues till the destination is found. When destination receives the RREQ packet, it replies source with a RREP [Route Reply] packet along the reverse of the route path recorded in RREQ. *Route maintenance* is done by the use of RERR [Route Error] packets and acknowledgments. RERR packet is sent by a node to the source when a link breaks between the sender and receiver nodes. When a RERR packet is received, the flawed hop is removed from the node's route cache and from all the routes [4][6]

LIMITATIONS OF DSR:

- The RERR packets are sent by receiver node to the sender, and it continues backwards to the source node, informing it about the problem, and this activity results in the increase of the packet delay time
- The source needs to append the ids of all nodes along the path to the destination, and due to this, the overhead in every data packet sent, is increased
- As mobility increases, more links are broken which causes a significant increase in the drop packet fraction.

3. PROPOSED PROTOCOL

3.1 Proposed Network Model

DSR is selected as the baseline routing protocol because it is an "On-Demand" Source Routing Protocol. The proposed study focuses on factors which influence the path construction. In the proposed model, these factors are: load, energy, noise, bandwidth, and hop counts. Use of an alert packet (AP) and update packet (UDP) is also proposed in path maintenance.

- **Alert Packet** The Alert packet is a small sized, fixed length packet, like an acknowledgement packet. The size of Alert packet will not increase as it visits the node. It is sent from destination to source at regular intervals, on the same route that has been with the source, indicating that the current working path is still valid [3].
- **Update Packet** The Update Packet is also a small sized packet, and is used to update back the source in case of a link failure, detected by any of the path's intermediate nodes. It contains information about the failure link and alternative amended path selected by the intermediate node.
- **Delay Time** Delay time will depend upon traffic and distance between two farthest pair. If the network is congestion free, then the time taken by a packet to travel between two nodes is at least 2 times the packet to travel in one direction
- **Traffic** Traffic is the average congestion that can exist in longest path of the network for timely delivery of alert packet in the worst case, in other words in the case of a highly congested network
- **Distance between two farthest nodes**
  The average time taken by the alert packet to reach from destination to source will always be less than or equal to the average time taken by the alert packet to traverse the longest distance possible in the same network

3.2 Proposed Algorithm

The proposed algorithm consists of three phases: *a) path /route discovery, b) path/route construction and c) path/route maintenance*. The algorithm also uses initial phase of DSR algorithm for route construction and for deciding all possible paths between given source and destination.

3.2.1 Path Discovery

The specific goal of this approach is to select a route based on survivability parameters like *max Energy, max Bandwidth, min Load and min Hop Count, and min Noise* among the entire route requests arrived. Such an approach will result in shorter, best and effective routing that also ensures longevity of the network lifetime. Node which has to send packet to another node, checks its path cache first. Path Cache is a kind of buffer that contain information about paths emanating from that node as a source. If there is no existing route for the required destination node, then the source node broadcasts a route request with a sequence id and destination address. The sequence id identifies a particular route request. Each node maintains route request table, which maintains the information about all the route requests the node has received before. The source node makes a note of the route request it is has sent in the route request table. RRP [Route Request Packet] travels that route, adds node id of all the visited nodes until the destination node is found. When a node receives the route request, stream array is checked. Stream array (S) of Size equal to number of adjacent neighbours to that node. And if the node is the destination of the route request, it sends a route reply back to the source node by simply tracing the path from the RRP. If the node is an intermediate node, it checks its path cache. If it has a route to the destination in its cache, the node sends a route reply to the source node by tracing the path from its cache and the path from the RRP. If the intermediate node does not have any obtainable route to the destination in its path cache, it rebroadcasts the RRP. After receiving the first route reply, source sends all the data packets intended to the destination using the path map retrieved from the route reply. Additional route reply that source node gets will act as an alternate to the current operational path [7].

Let m be the node that broadcasts RRP and n be one of the adjacent neighbours of m. Whenever node m broadcasts a RRP it updates only those elements of its stream array that have null value present in it by inserting 1 in S_{mn} where n belongs to one of the neighbours of m. Correspondingly, neighbours that will receive RRP will update their stream array by inserting 0 in S_{nm} where n receives packet from m. Now n will also broadcast this received RRP to all its adjacent neighbours and updates only those elements of its stream array which have null value present in it. After receiving RRP, neighbours of n will respond in similar manner as described above.

Algorithm for Path Discovery

Table 1: Pseudo Code for Path Discovery

Begin {Find the next adjacent node based on the present node value}
Step 1: node m sends RRP to all neighbor nodes
If the value of Stream array S_{mn} = null
Then insert
$S_{mn} \leftarrow 1$
Step2: Node n received RRP and then it updates its Stream array $S_{nm} \leftarrow 0$
Step3: If n is destination node
Then
Send RRP to source node
Else
If n has any path to D in route catch
Then rebroadcast RRP
Step4: Else m=n go to step 1
Else node n discards the RRP as $S_{mn} \neq$ null
Stop.

3.2.2 Path Construction

The next step of the protocol is to determine the path through which packets is to be forwarded. The logical path from source to destination is constructed on the basis of the survivability factor

3.2.2.1 Parameters affecting the Selection of path

The route reply packet contains five parametric values for path determination

- **Number of Hop Traveled (H):** When destination sends the route reply packet to source node it counts the number of intermediate nodes traversed as Hop count.
- **Bandwidth (B):** The RRP travels from destination to source as it visits and join all nodes and links in pathway. As it travels it stores the available bandwidth of the link. The minimum value among all traversed links is the obtainable bandwidth for data broadcast
- **Traffic Load (T):** The outgoing and incoming traffic at any node may be calculated by the size of output and input buffer of that node. The input buffer may act as output buffer and vice versa.

  Traffic Load (T) = Outgoing traffic at that node – Incoming traffic at that node

- **Energy (E):** Energy of nodes is one of the important factors
  for route discovery. Maintaining an optimized lifetime of a routing path in a network is a very challenging task because the power or energy of the nodes depends on the size, model, property, and capacity. Selection of a node with low energy level reduces the stability of the communication path as that node may run out of energy causing the breakdown of the communication channel [5]

- **Noise (N):** All real measurements in any network are disturbed by noise. Several types of noise as thermal noise, induced noise, impulse noise, may corrupt the signal. In MANET, it is always needed to choose a channel with lower noise that result in reduction of number of dropped packets to increase the quality of service (**QoS**).[7].

3.2.2.2 Survivability Factor (SF)

The survivability factor of each path is calculated on the basis of Bandwidth (B), Traffic Load (T), Hops (H), Energy (E), Noise (N) parameter. These parameter effect the survivability of the link as following:

SF=F (B, T, H, E, N)

Survivability factor is directly proportional to the bandwidth and energy at each node and is inversely proportional to the noise of channel, traffic load, hop count i.e. no. of edge in a route between source and destination**.** Survivability factor in terms of bandwidth, energy of each node, noise at channel, traffic load and hop count can be defined as follows:

$$SF = K\left(\frac{B,E}{N,T,H}\right)$$

Where K is system design constant, K ≥ 1.
Path is much more survivable if it has max energy, max bandwidth, min load, min noise and min hop count amongst the entire route. On the basis of SF we shall prioritize the paths. After the determination of the first path, node will start sending the packets and for all subsequent route replies, it shall calculate SF and prioritize them [10]. The directed tree will be constructed for a given network which contains all the possible paths between the source and the destination**.** The constructed tree for the considered ad hoc network is shown in the fig. 1.

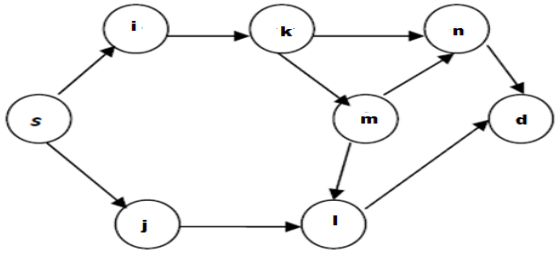

Fig 1: Tree for the Network

Following set of rule should be considered for choosing an optimal route:

Rule 1: If the routes are of equivalent energy
Then
Route with maximum avail bandwidth will be selected
Rule 2: If the routes are of equivalent energy and equivalent bandwidth:
Then
Route with minimum traffic load will be selected.
Rule 3: If the routes are of equivalent Energy, equivalent Bandwidth and equivalent Load also
Then
Route with minimum noise will be selected
Rule 4: If the routes are of equivalent Energy, equivalent Bandwidth equivalent Load and equivalent noise also
Then
Route with minimum Hop Count will be selected

Rule 5: If the routes are not of equivalent Energy:
Then
a) Route with maximum Energy should be taken
b) Route with maximum bandwidth should be taken
c) Route with minimum Load should be taken.
d) Route with minimum noise should be taken
e) Route with minimum hop count should be taken

The preference of order for selecting optimal route is as follows

Energy > Bandwidth > Load > Noise > Hop Count

The following paths were found from the source S to the destination D (fig: 1).

1. s→i→k→n→d
2. s→i→k→m→n→d
3. s→i→k→m→l→d
4. s→j→l→d

The paths are chosen in order of decreasing survivability factor. The path with high survivability factor is chosen first Hence, s→j→l→d is the selected optimal path.

3.2.3 Path Maintenance

Whenever a node m sends a data packet to the next hop, it waits for the acknowledgement from the receiving node n. Node m may not receive the acknowledgement packet due to the failure of link between m and n or due to the high congestion at the link. If the node m detects that link to the next hop (i. e. n) is erroneous; it sends the data packet to previous node in that path (i.e. k) and checks for the availability of any path from that node to destination in its path cache[8][9].In this situation, two cases may arise:

- If it finds any path in the path cache, it changes the route map of the data packet and sends an UPD packet to the source *s* informing about the particular link failure and amended path. After receiving the UPD packet the source deletes all paths that contain failed link. It considers the new found path as the current working path
- If path is not found it simply sends the UPD packet to the source informing about the particular link failure.

The following two cases arise at the source node

- If this UPD packet reaches the source before the expiry of the delay time, source deletes all those paths that contain the failed link. The source chooses the alternate path having the highest priority from its path cache.
- If delay time expires before the arrival of UPD packet source ***s*** chooses alternate path that have highest priority from its path cache and sends the data packet through it. When the UPD packet arrives at source node, it checks whether failed link is contained in the new chosen path or not and simultaneously deletes all paths that contain failed link. If the failed link is found, the source annuls the current transaction in chosen path and selects the path having the highest priority (highest value of survivability factor) as the new working path and sends data packet through it. If the failed link is not found in the newly selected path then the source continues to send data packets through the newly selected path. If source doesn't have any alternate path then it again starts the path discovery process[10].

Algorithm for Path Maintenance

Table 2: Pseudo Code for Path Maintenance

Begin
{Find the node and link of the effective path}
If {link failed between m→n}
Then
Node m send packet to k where k is previous node of m &
check for substitute path from route cache.
If (substitute path found)
Then
Change the route path map and send update packet to source s
for informing and deleting all path containing failed links
Else
Send update packet to source node
If
UPD packet reaches before delay time expires
Then choose new path
Else
Source s deletes all paths having failed link
Then s choose highest priority path from remaining one.
End

4. SIMULATION RESULT

The performance of the protocol is evaluated through the results obtained by extensive simulations performed with C++ ns-2 simulator with mobility framework. A node sends a packet to set RTS (Request-to- Send) flag of its neighbors and the intended receiver sets CTS (Clear-to-Send) flag of its neighbors. Nodes whose RTS or CTS flag is set cannot transmit data, except the sender. Control packets have higher priority over data packets. Propagation delay is assumed to be negligible and it is assumed that packets always arrive without any bit error. The source node generates packets at a constant rate. We have evaluated the performance in terms of throughput obtained by a densely populated network. We have considered a network of 5 to 40 nodes with an increasing number of neighbors from 5 to 10 nodes. Each node has a traffic flow with infinite demands towards one of its neighbors.

Figures below shows the effect of energy and noise, along with node bandwidth, on throughput. This is carried out with varying number of nodes and simulation time (*it-iteration*) respectively

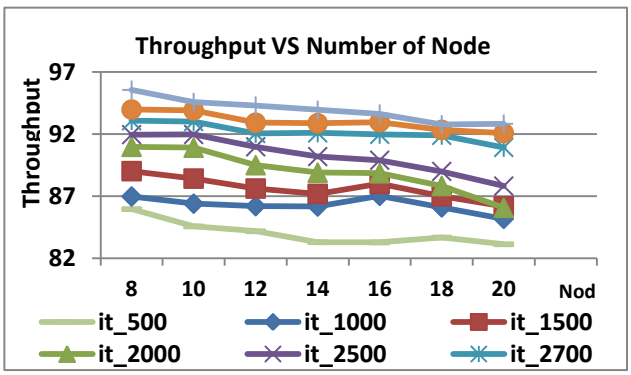

Fig 2: Effect on Throughput, on Ideal Nodes[#]

Above fig shows the effect on throughput (*on ideal nodes*[#]) as numbers of nodes are increased slowly. The work considers bandwidth of the nodes ('bandwidth' being the essential attribute of nodes for computing the path) as the node value, in determining the path, and hence throughput. The same work for all the nodes is repeated across different simulation time (it-iteration). We see that as the simulation time increases, **throughput (or data delivery)** also increases, but there is also a gradual fall in throughput as the number of nodes increase i.e. the drop packet increases with the increase in number of nodes.

#Ideal nodes i.e. nodes considered without factors of energy or noise

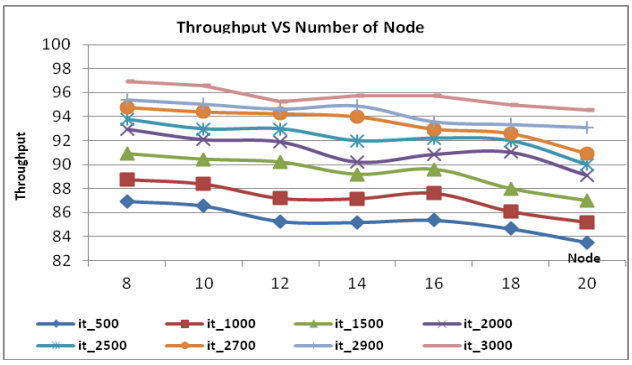

Fig 3: Effect on Throughput after considering Energy of nodes

Fig 3 shows the effect on throughput after considering the energy of individual nodes along with bandwidth, and repeating the previous work [as in Fig 2]. Varying energy values [high to low] of nodes were considered, as the real-time nodes will have variable energies. We observe that the results of throughput take a very marginal dip when compared to the ideal case earlier.
Energy of node proves to be a direct proportionate factor in Survivability. If more bullish values of energy are taken, results will definitely be better.
The number of nodes and simulation time has the same effect as earlier.

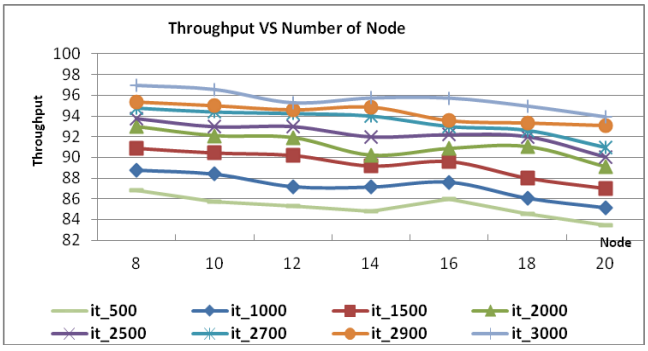

Fig 4: Effect on Throughput after considering Noise at nodes

It shows the effect on throughput after considering the noise at individual nodes along with bandwidth, and repeating the previous work [as in Fig 3]. Varying noise values [low to high] of nodes were considered, considering real-time nodes will definitely not be free of noise. While nodes having feeble noise can be ignored, and such nodes can be considered '*almost ideal*', noise above a certain '*feeble threshold*' has to be considered practically. After setting up such threshold, and doing the experiment, it is observed that the results of throughput take a substantial dip when compared to the ideal case earlier. Noise of node proves to be an indirect proportionate factor in Survivability. The number of nodes and simulation time has the same effect as earlier.

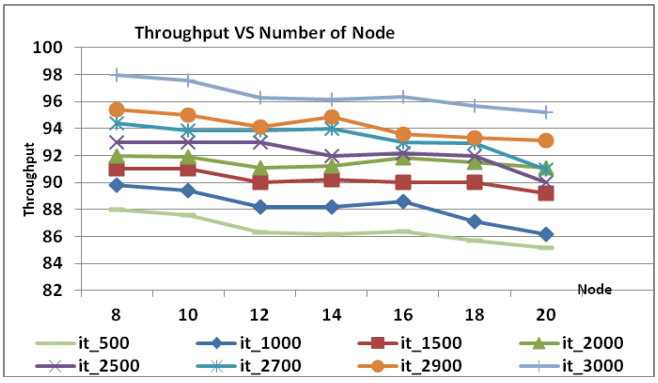

Fig 5: Cumulative Effect on Throughput after considering Energy and Noise of nodes

More practical scenarios exist by cumulating parameters or "*simultaneous consideration of parameters"* i.e. considering both Energy and Noise in one work along with bandwidth. This is exactly what Fig 5 shows. The cumulative effect's results are slightly different when energy or noise is taken individually.

It is practically possible that a node exists with less noise, but still lesser energy, and such a node is not going to contribute much to the throughput as against only less noise consideration [as in Fig 4] wherein energy is assumed to be ideal, and this node becomes the most contributing node thus increasing the throughput. Similarly a node with high energy but also high noise is not going to contribute much.
The work can be extended to include more node influencing parameters simultaneously and individually, to make a more robust and practical DSR, and our future work would be dedicated towards it.

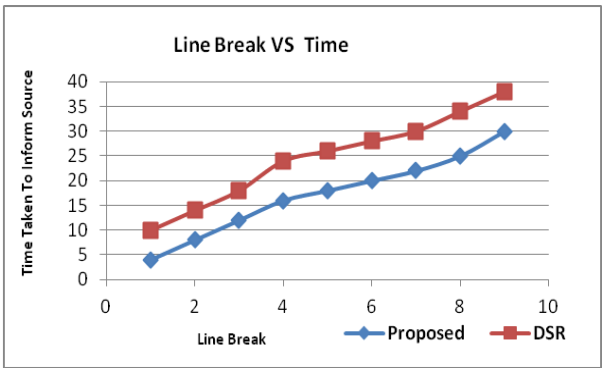

Fig 6: Comparison of DSR with our approach

Shows the results of AP and UDP. It shows that our proposed protocol of AP and UDP takes less time source node when link fails between sender node to the destination node.

5. CONLUSION

The proposed protocol is based on DSR protocol. The proposed mechanism involving AP and UDP improves the performance of a route by reducing the link failure, increasing life time, minimizing overhead, thus reducing time to inform source node about line break.

The real-time and practically possible outcomes can be obtained on the basis of parameters like traffic load, energy, noise, bandwidth and number of hops. These parameters prove decisive factor in selection of a mobile ad-hoc network (MANET) to choose successive node.

Due to multiple paths, the proposed approach identifies the accurate location of the link failure. The paths are chosen in order of decreasing survivability factor. The path with high survivability factor is chosen first in case of link failure. The reliability of the chosen path is more, thus improves the performance.

6. ACKNOWLEDGMENTS

I would like to thank Mr. A.K. Daniel, Associate Professor, Department of Computer Science & Engineering MMMUT – Gorakhpur, India for his valuable support and guidance without which this work wasn't possible. Special thanks to Ms. Pooja Chaturvedi, Research scholar, Department of Computer Science & Engineering MMMUT, Gorakhpur for her consistent assistance and suggestions throughout this effort.

7. REFERENCES

[1] "Mobile Ad hoc Networks", Mobile Network and Applications, Vol. 6, No. 3/June 2001.

[2] Chi-Chun Lo, Bin-Wen Chuang, "A Novel Approach of Backup Path Reservation for Survivable High-Speed Networks", IEEE Communications Magazine, March 2003, pp 146-152.

[3] Vincent D. Parka and M. Scott Corsonb, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Copyright 1997 IEEE. Published in the Proceedings of INFOCOM'97, April 7-11, 1997 in Kobe, Japan.

[4] Thomas Stidsen, Bjorn Petersen, Kasper Bonne Rasmussen, "Optimal Routing with Single Backup Path Protection", IEEE Communications Magazine, 40(1):34–41, 2002.

[5] C.E.Perkins, E.M.Royer, S.R.Das, and M.K.Marina, "Performance comparison of two on demand routing protocols for ad hoc networks", IEEE Personal Communications, vol.8, pp.16–28, Feb. 2001.

[6] David B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts", in Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94) , pages 158–163, December 1994.

[7] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad HocWireless Networks. In Mobile Computing", edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[8] David B. Johnson, David A. Maltz, Yih - Chun Hu, and Jorjeta G. Jetcheva. "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet-Draft, draft- ietf-manet-dsr-07.txt, February 2002.

[9] Josh Broch, David B. Johnson, and David A. Maltz., The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet-Draft, draft-ietf-manet-dsr-03.txt, October 1999, earlier revisions published June 1999, December 1998, and March 1998.

[10] Carla F. Chiasserini, Pavan Nuggehalli and Vikram Srinivasan, Energy-Efficient Communication Protocols," Proceedings of 39th Design Automation Post-Conference (DAC) 2002, June 2002.