

# Stability Aware Routing in Mobile Ad-Hoc Networks using multiple Route

Aniket Mathuriya<sup>1</sup> & Pushpraj Pal<sup>2</sup> Lovedeep Grover<sup>3</sup>

Department of Computer Science & Engineering<sup>1</sup> Department of Electronics & Communication<sup>2,3</sup>  
SRMIET Ambala, Haryana, India

**Abstract:** Previous work on routing in MANETs has resulted in numerous routing protocols that aim at satisfying constraints such as minimum hop or low energy. Existing routing protocols often fail to discover stable routes between source and sink when route availability is transient, i.e., due to mobile devices switching their network cards into low-power sleep modes whenever no communication is taking place. In this thesis, we introduce a new approach stability aware source routing protocol that is capable of predicting the stability (i.e., expiration time) of multiple routes. Proposed protocol selects the route that minimizes hop count while staying available for the expected duration of packet transmission. The stability aware routing (SAR) resolve the problem of SADSAR protocol indicate a significant increase in route discovery success rate with comparable route establishment and maintenance overheads.

## KEYWORDS

Mobile Ad Hoc Networks, Dynamic Source Routing (DSR), Dynamic Power Management (DPM), Stability Aware Routing

## 1. INTRODUCTION

A Mobile Ad-Hoc Network (MANET) is a collection of mobile nodes (such as laptops, PDAs) forming as arbitrary networks without the support of any fixed infrastructure such as base station or access point. In MANET, each node functions as a router and forwards packets for other peer nodes. There is no fixed topology due to the node mobility, which results in interference, multipath propagation and path loss. Mobile nodes are constraints to battery power, computation capacity, bandwidth, and wireless channel leading to number of challenges while design routing procedures. Determining viable routing paths and delivering messages in a decentralized environment where network topology fluctuates has always attracted the attention of researchers to design new and new mechanism to solve these problems. While the shortest path (based on a given cost function) from a source to a destination in a static network is usually the optimal route, this idea is not easily extended to MANET.

MANET Routing protocols are classified into three categories: table-driven, on-demand and hybrid routing protocols. The table-driven

routing protocols (DSDV, FSR, CGSR, WRP, GSR) determine the path to destination before it is needed. On-demand routing protocol (AODV, DSR, TORA, ABR) determines the route to destination when required. Hybrid routing protocols used both approaches.

In a mobile ad hoc network, nodes are often powered by batteries. The power level of a battery is finite and limits the lifetime of a node. Every message sent and every computation performed drains the battery. One solution for power conservation in mobile ad hoc network is power awareness routing. This means that routing decisions made by the routing protocol should be based on the power-status of the nodes. Nodes with low batteries will be less preferably for forwarding packets than nodes with full batteries, thus increasing the life of the nodes. A routing protocol should try to minimize control traffic, such as periodic update messages to improve the lifetime of the nodes and network. However, not every routing protocol is suitable for implementing power awareness routing and different approaches on power awareness routing can be followed.

The highly dynamic character of a mobile ad-hoc network (MANET) poses significant challenges on network communications and resource management. Previous work on routing in MANETs has resulted in numerous routing protocols that aim at satisfying constraints such as minimum hop/distance or low energy. Existing routing protocols often fail to discover stable routes between source and sink when route availability is transient, e.g., due to mobile devices switching their network cards into low-power sleep modes whenever no communication is taking place.

In this paper, the work proposed a stability aware routing protocol (SAR) that is capable of predicting the stability (i.e., expiration time) of multiple routes. SAR then selects the route that minimizes hop count while staying available for the expected duration of packet transmission. The proposed protocol increases the successful packet transmissions with comparable route establishment and maintenance overheads. A new approach in Stability Aware Routing (SAR) is based on the prediction of future sleep times of mobile nodes (i.e., the times when mobile nodes' DPM techniques will turn off their network cards). The goal of this approach is to introduce DPM-awareness in routing decisions and thereby to increase the number of successful packet transmissions.

## 2. RELATED WORK

### 2.1 Dynamic Power Management

MANETs has gained huge popularity over the last decade. Mobile nodes in wireless ad-hoc networks often put their wireless network cards to sleep when they are not transmitting or receiving data. In most MANETs, wireless traffic is infrequent and recent work [1] shows that wireless network cards should be turned inactive for 50% or less of the entire lifetime to obtain a balance between optimal power-saving and sustained network connectivity. But sleep modes can lead to loss of network connectivity and hence lower the packet delivery ratio. This paper focuses on routing in MANETs with transient route availabilities, i.e., route establishment takes into consideration the expiration time and therefore the stability of a potential route. This new approach is based on the prediction of future sleep times of mobile nodes (i.e., the times when mobile nodes' DPM techniques will turn off their network cards). The goal of this approach is to introduce DPM-awareness in routing decisions and thereby to increase the number of successful packet transmissions. The concept of stability awareness can be added to any routing protocol. A variety of stability predictors can be used, including hints given by applications and/or the DPM mechanism.

### 2.2 Network Sleep Time

In [2], Chia et al. propose that devices which are not currently active in any data communication may enter a sleep state, but can be powered up remotely through a signal using a simple circuit based on RF technology. Radio devices select different time-out values (sleep patterns), to enter various sleep states depending on their battery status and quality of service. In [3], Singh et al. employ a MAC layer protocol for PAMAS (Power Aware Multiple Access protocol with Signaling) in which nodes overhearing transmissions between two other nodes turn themselves off and wake up after an interval of time equal to the total transmission time as indicated in the RTS/CTS message exchange between the sender-receiver pairs. They deploy metrics such as minimize energy consumed per packet or minimize time to network partition, and verify these metrics with their proposed MAC layer protocol. In proposed protocol, the sleep and awake schedule is determined from prediction of link expiration based on the queue contents of the packet scheduler and the network interface device timeout value. The DPM schedule is Somewhat conservative since it ignores the possibility of more packets being added before the timeout expires.

## 3. PROBLEM FORMULATION

DPM supports energy conservation by making mobile nodes put their wireless network cards to sleep when no data communication is taking

place. A consequence of this technique is that mobile nodes will be unreachable for large periods of time. Therefore, we need to know the accurate network 'up' and 'down' times (DPM schedule) in order to introduce DPM awareness in routing decisions. Currently the protocol predicts the DPM schedule for mobile nodes from the queue contents of the packet scheduler and the network device timeout value. Toward that end, the protocol computes the minimum time to transmit all packets currently residing in the packet scheduling queue and adds the device-specific timeout value, each data packet's transmission time is calculated as follows[4]:

$$T_{data} = T_{rts} + T_{cts} + T_{ack} + T_{difs} + 3T_{sifs} + \frac{(P + Q)}{BW_{channel}}$$

Where  $T_{data}$  - transmission time of each data packet

$T_{rts}$  - Time for transmitting RTS

$T_{cts}$  - Time for transmitting CTS

$T_{ack}$  - Time for transmitting ACK

$T_{difs}$  - DCF inter frame space defined in the IEEE 802.11 protocol standard

$3T_{sifs}$  - Short inter frame space defined in the IEEE 802.11 protocol standard

$P$  - Size of the data packet

$Q$  - IP and MAC packet header length

$BW_{channel}$  - Channel capacity

We define the route uptime factor (RUF) as a metric which indicates the earliest up time when the link between any pair of adjacent nodes on a route is going to be interrupted due to one (or both) of the nodes being put to sleep. Now we derive RUF as follows: If we assume nodes as vertices and the links between the nodes as edges connecting them, then let  $G(V,E)$  be the graph representing the network topology where  $V$  is the set of vertices and  $E$  is the set of edges. Let

$$R_{ij} = (V_i, V_{i+1}, V_{i+2}, \dots, \dots, V_k, V_{k+1}, \dots, V_{j-1}, V_j) \quad (1)$$

be the route from source node  $V_i$  to  $V_j$  through intermediate nodes  $V_k, V_{k+1}$ , etc. Let  $\Omega_{ij}$  be the set of all possible alive routes between  $V_i$  and  $V_j$ . The DPM sleeping schedule  $S_{ij}$  for the route  $R_{ij}$  is defined as

$$S_{ij} = (t_i^{up}, t_{i+1}^{up}, t_{i+2}^{up}, \dots, t_k^{up}, t_{k+1}^{up}, \dots, t_{j-1}^{up}, t_j^{up}) \quad (2)$$

Where  $t_i^{up}$  is the earliest up time for node  $V_i$ . We define the link uptime vector or  $L_{ij}$  for the route  $R_{ij}$  as

$$L_{ij} = (t_i^{uptime}, t_{i+1}^{uptime}, t_{i+2}^{uptime}, \dots, t_k^{uptime}, t_{k+1}^{uptime}, \dots, t_{j-1}^{uptime}) \quad (3)$$

Where  $t_i^{uptime}$  is the uptime of the link  $E_{i, i+1}$  connecting nodes  $V_i$  and  $V_{i+1}$  and is defined by  $\text{average}(t_i^{up}, t_{i+1}^{up})$ , since uptime of a link is determined by how long the

$$t_i^{uptime} = \text{average}(t_i^{up}, t_{i+1}^{up}) \quad (4)$$

link will be alive before breaking down due to one of its end nodes going to sleep and thus essentially is expressed by the minimum of the DPM sleeping schedule of the end nodes. The route uptime factor  $RUF_{ij}$

for route Rij can be expressed as the minimum of the link uptime vector Lij along the route since it will indicate how long the route will be alive before breaking down due to the break in any of its constituent links:

$$RUF_{ij} = \min (t_i^{uptime}, V_i \in V, t_i^{uptime} \in Lij) \quad (5)$$

given the next earliest time to sleep ti off for each node Vi ∈ V, in the graph G(V,E), accumulate the set of all possible routes Ωij between nodes Vi and Vj with the corresponding route uptime factors RUF ij for each Rij ∈ Ωij and find the min-hop route Rij from the set of all stable routes Ωij . If there are more than one routes with the same min-hop length, then the one with the minimum route uptime factor value is selected. since it has the highest predicted lifetime. The route uptime factor contains the all link uptime value between two node which is satisfied the transmission threshold value. The each node stores the link uptime vector which contains the all stable link between two nodes from source to destination.

#### 4. PROPOSED WORK

##### 4.1 The propose protocol will follow the following steps to identify the stable path:

1. Source Node initiates the Data Transmission Request.
2. Check for the destination node in route cache. If found then forward the data packet to Destination.
3. If destination node is node found in route cache then broadcast the route request Packet(RREQ) to their neighbor node
4. Calculate the link uptime value  $t_i^{uptime}$  between the node Vi to  $V_{i+1}$   

$$t_i^{uptime} = \text{average}(t_i^{up}, t_{i+1}^{up})$$
5. if  $t_i^{uptime} > th$  (Threshold) then add in to the link uptime vector  

$$Lij = (t_1^{uptime}, t_{i+1}^{uptime}, t_{i+2}^{uptime}, \dots, t_k^{uptime}, t_{k+1}^{uptime}, \dots, t_{i-1}^{uptime})$$

And forward the packet to the next node.
6. if the intermediate node receive more than two RREQ packet then  
 If  $T_k^{uptime\_previous\ RREQ} > T_k^{uptime\_latest\ RREQ}$  then forward previous RREQ and discard latest RREQ otherwise forward latest RREQ
7. Repeat step 3 to step 6 till destination is found.
8. if destination found then store all route in to the routing table in increasing order of Route Uptime Factor  $RUF_{ij} = \min(t_i^{uptime}, V_i \in V, t_i^{uptime} \in Lij)$  and send RREP to the first entry of routing table

##### 4.2 Algorithm for the Node already visited

```

Algorithm Node_Visited(NA Node Address)
{
  n = Count(Add)
  For(j=0;j<n;j++)
  {
    if (NA == Add[j]);
      return 0;
  }
  return 1;
}
    
```

Fig. 4.1 Algorithm for the Node already visited

4.3 Algorithm for forwarding the data packet from source to destination is as follows:

```

Algorithm SAR(DA Destination Address, DP Data Packet, Route_Cache)
// TTL=16
{
    n = Count(Address)
    For(j=0; i=Route_cache_size; i++)
    {
        If (Route_Cache[i]==DA)
        {
            forward DP;
            break;
        }
        Else
        {
            X = Find_Route(Tkup,DA,Threshold,LUV);
            If X=RREP
            {
                RUF=min(LUV);
                add X of min(RUF) in Route_Cache;
            }
            Else
            {
                Drop DP;
                Forward RERR;
                Break;
            }
        }
    }
}
    
```

Fig. 4.2 Algorithm for forwarding data packet from source to destination

```

Algorithm Find_Route(Tkup,DA,Threshold,LUV)
{
    If(DA==NA)
    {
        Return 0;
    }
    Else If (TTL<=n)
    {
        Tkuptime=avg(Tkup,Tk+lup);
        Prev. Tkuptime=Tkup;

        If(Tkuptime>Threshold)
        {
            add Tkuptime to LUV;
            forward RREQ;
            if (Node_Visited==0)
            {
                if(Prev.Tkuptime > Tkuptime)
                    Forward (Tkuptime=Avg(Prev.Tkuptime,Tkuptime));
                else Forward (Tkuptime=Avg(Tkuptime,Tk+luptime));
            }
            Find_route(Tkuptime,DA,Threshold,LUV);
        }
        else drop the DP;
    }
    else drop the DP;
}
    
```

Fig. 4.3 Algorithm for find stable routes between source to destination

4.2 Route discovery phase in stability aware routing

When a source node needs to send a data packet to a target node, it first searches its routing cache for any entry using the target node address as the key. An entry in the routing cache contains a list of stable routes to the target node. If a routing cache entry is found, then the source node picks a route. If no entry is found for the target node, then the source node initiates a route discovery for the target. The proposed protocol adds four new entry types to the RREQ packet format of standard DSR

.Link Uptime Vector ( $t_i^{uptime}$ ;  $i \in (1, \dots, N - \varnothing 1)$ ) for the route,

1. Partial route  $R_{ij}$   
 $= (V_i, V_{i+1}, V_{i+2}, \dots, V_k, V_{k+1}, \dots, V_{j-1}, V_j)$
2. Earliest Up-time of Last-upstream node ( $t_i^{up}$ )
3. Threshold value ( $Th = T_{data}$  transmission time of each data packet)

The protocol allows intermediate nodes to forward multiple RREQ packets with the same <source address, request id> pair if the packets contain distinct source routes. During the RREQ lookup at intermediate nodes, the 4-tuple <source address, request id, last upstream node address, partial route length> is checked with each entry in the recently seen requests list for possible match. If no match is found, then the RREQ contains a distinct source route and is eligible to be forwarded if the contained source route is predicted to be stable. If the intermediate node receives the more than two RREQ packets. It does not forward these all RREQ packet to their neighbor node. The intermediate node performs the comparison between the received RREQ and the previous RREQ packet. If uptime of previous RREQ is greater than to the Received RREQ then discard the received RREQ and forward the previous RREQ uptime.

Figure 4.8 shown the route Discovery process in the given example the source node 1 Broadcast the RREQ packet to their neighbor node and try to find the stable route between source node 1 to destination node 5. The Fig. 4.5 has been shown the Route Request packet header of SAR.

RREQ packet header of SAR has the four new entry in the standard RREQ packet header of DSR. Fig.4.4 has been shown standard Route Request packet header of DSR. SAR predicts the route stability using a link by link stability prediction. Each intermediate node received RREQ and predicts the stability of the link between itself and the last upstream node. All previous links in the source route are assumed to be stable, otherwise the previous upstream nodes would not have forwarded the RREQ packet.

Type	Reserved	Hop Count
Broadcast ID		
Destination IP		
Destination Sequence Number		
Source IP address		
Source Sequence number		

Fig.4.4 Route Request Message Format in DSR

Thus the stability of the current link ensures the stability of the entire source route. For each received RREQ, intermediate node  $V_{k+1}$  calculates the uptime of the link between itself and the last upstream node recorded in the RREQ and appends it to the Link Uptime Vector in the RREQ. If the uptime is less than Threshold, then the link will not be stable for the entire period of exchanges of the RREQ, the following RREP and then the data packet. Hence the intermediate node discard the RREQ

Fig.4.5 Route Request (RREQ) packet in SAR

4.6 Flow chart of route discovery at source node

4.7 Flow chart of route discovery at Intermediate node

4.9 Example of route discovery in SAR

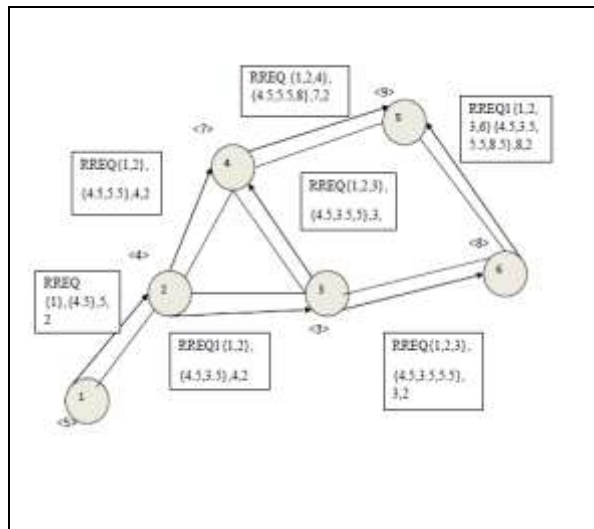


Fig. 4.8 Route discovery in SAR

In this example the source node 1 broadcast the route request packet (RREQ) to find the stable route for transmitting the data packet to the destination node 5. The RREQ packets contain the information of partial route and link uptime vector for each route. The link uptime vector store the value of each link uptime value which is the average of uptime value of both nodes. In this

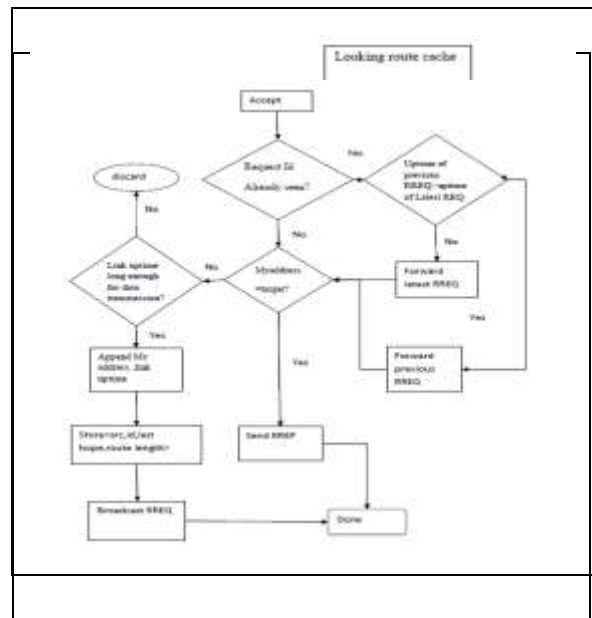
figure each node has an uptime value. In this example the threshold value is

Source Address	Source Sequence #	Destination Address
Partial Route		Link Uptime Vector
Earliest Up-time of Last Upstream Node		Threshold value

2.the node 4 receives the two RREQ packets from node 3 and 2. Node 4 discard the RREQ of node 3 because the uptime value of node 3 is less then to the uptime value of node 2 .

4.10 Route reply phase in SAR

When the RREQ reaches the target. the route is predicted to be stable, the target node sends an RREP packet back to the source along the reverse path recorded in the RREQ. In this proposed protocol the routing table is maintained at the destination node. At the destination node the all stable route store in the routing table in increasing order of the Route uptime factor .In the



route reply phase the destination node only send the one RREP packet which contain the first entry of the routing table Hence it leads to reduce the total transmission time between source to destination. Figure 4.12 shown the route reply process in the given example The proposed protocol adds three new entry types to the standard DSR RREP packet format:

1. source route  
 $R_{ij} = (V_i, V_{i+1}, V_{i+2}, \dots, V_k, V_{k+1}, \dots, V_{j-1}, V_j)$
2. Link uptime vector  $(t_i^{uptime}; i \in (1, \dots, N - \text{Hop} - 1))$

3. Earliest up time. The minimum of all the Link Uptime Vector elements.
4. Estimated Transmission Time

Fig.4.9 Route Reply Message Format in DSR  
Fig4.10 Route Reply (RREP) packet Format in SAR

**4.11 Route selection phase at the destination node**

At the destination node the all stable route store in the routing table in increasing order of the Route uptime factor .In the route reply phase the destination node only send the one RREP packet which contain the first entry of the routing table Hence it leads to reduce the total transmission time between source to destination

Fig.4.11 Routing table Entry in SAR

The example 4.12 is based on the processor of route replay in the proposed protocol

In The figure 4.12 the destination node receive the two RREQ packet . at the destination node the all route store in the routing table . the route store in the increasing order of Route Uptime factor.

The RUF of first route{1,2,4} is 4.5 and RUF of route{1,2,3,6} is 3.5. hence the destination node first send reply packet (RREP) to the route {1,2,3,6}.

**4.12 Example of Route Reply in SAR**

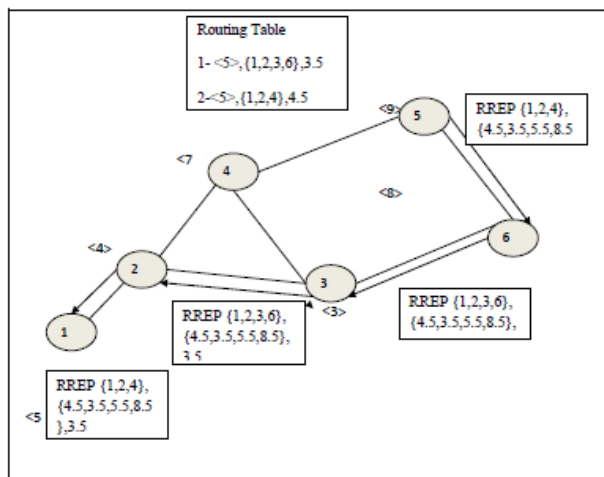


Fig. 4.12 Route Reply in SAR

**5. EXPERIMENTAL RESULTS**

In order to validate the proposed protocol and show its efficiency we present simulations using MATLAB. MATLAB is a very popular network simulation tool. MATLAB is an interactive software package which was developed to perform numerical calculations on vectors and matrices. Initially, it was simply a Matrix Laboratory. However, today it is much more powerful:

- It can do quite sophisticated graphics in two and three dimensions.

- It contains a high-level programming language (a \baby C") which makes it quite easy to code complicated algorithms involving vectors and matrices.

Type	Reserved	Hop count	L
Destination IP Address			
Destination Sequence Number			
Lifetime			

Source Route		Link Uptime Vector	
Earliest Up Time	Destination Address	Source Route	Estimated Transmission Time
			Time

- It can numerically solve nonlinear initial-value ordinary differential equations.
- It can numerically solve nonlinear boundary-value ordinary differential equations.
- It contains a wide variety of toolboxes which allow it to perform a wide range of applications from science and engineering. Since users can write their own toolboxes, the breadth of applications is quite amazing.

Mathematics is the basic building block of science and engineering, and MATLAB makes it easy to handle many of the computations involved. You should not think of MATLAB as another complication programming language, but as a powerful calculator that gives you fingertip access to exploring interesting problems in science, engineering, and mathematics. And this access is available by using only a small number of commands and function because MATLAB's basic data element is a matrix (or an array).

**5.1. Simulation Setup**

- The simulated network area is 25pixelX25pixel with 10 to 25 nodes.
- Network simulates a random interval repeatedly for each transmitting node. Each node stays up for the input Uptime percentage of the simulated random interval and notifies the routing module about its next earliest time to sleep each time when it up.
- Each simulation was run for 5 trials, with a full range of uptime percentages from 0% to 100% with an interval of 10%. Each trial was run for 120seconds. Each transmitting source node attempts to send one data packet to the sink node.
- Nodes intermittently turn off their network cards for power-saving in absence of

Significant communication, which leads to link failures and low network connectivity.

- Simulated network use the threshold value 0.07 for the each link stability
- The node transmission range is 5 meter

**5.2 Results**

We evaluate the performance of SAR in terms metric as follows:

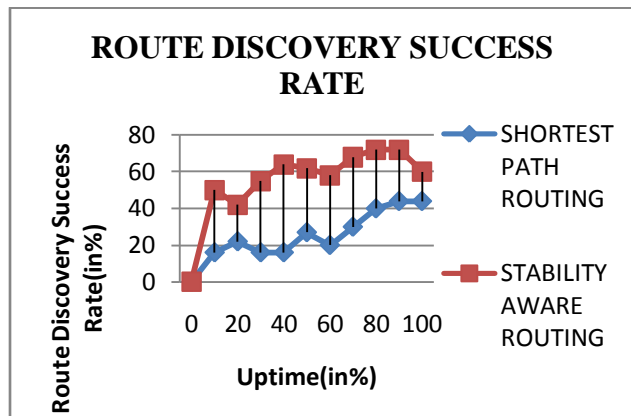


Fig. 5.1 Route discovery success rate

in Fig 5.1 the graph measure the route discovery success rate for both of the protocols. Route discovery success rate are the ratio of number of successful source nodes getting RREP packets back to the number of source nodes sending a RREQ packet. This graph gives an impressive performance improvement of SAR over standard shortest path Routing. It is expected because SAR uses both the min-hop and stability metrics in route discovery and finds more routes. At Uptime percentage of 100%,

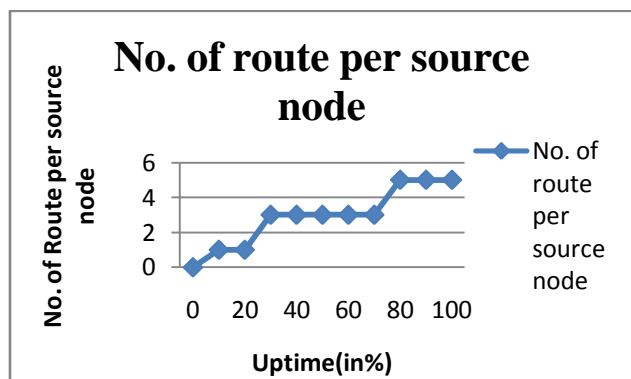


Fig. 5.2 Number of routes per source nodes

In Fig. 5.2 the average number of routes per node has been depicted. It shows how SAR finds more routes per node with increasing Uptime percentage.

**6. CONCLUSION**

The proposed protocol “Stability Aware Routing” introduces DPM awareness into the routing decisions and finds multiple stable routes to the target node. Intermediate nodes can broadcast multiple RREQ packets with the same sequence number. After broadcasting the first RREQ packet, only those RREQ packets having same sequence number will be broadcasted which have more stable route. Destination node will wait for some time after receiving the first RREQ packet.

Best optimal path will be selected from the received paths. SAR has been compared to shortest path routing which does not consider power-saving but optimizes routing for shortest delay and the proposed protocol provides a significant increase in successful packet transmissions with comparable route establishment and maintenance overheads. SAR solves the problem of SADSR Routing problems and reduces the packet overhead on each node and reduces the traffic of packet transmission. SAR is based on the prediction of future sleep times of mobile nodes (i.e., the times when mobile nodes' DPM techniques will turn off their network cards).

SAR is aggressive in sense that it proactively discards any RREQ which is predicts to be non-stable and thus might lead to a scenario where the source node fails to find any stable path to the sink node

**6.2 FUTURE WORK**

Future works would study how SAR performs with respect to other protocol. The concept of stability awareness can be added to any routing protocol. A variety of stability predictors can be used, including hints given by applications and/or the DPM mechanism.

SAR combines the two metrics stability and minimum threshold power. Future work would also study the combination of other metrics such as real-time or minimum-energy communication and how to obtain a balance of these multiple constraints to make the new protocol yield good performance.

**7. REFERENCES**

[1]. J. Monteiro, A. Goldman, A. Ferreira, "Performance Evaluation of Dynamic Networks using an Evolving Graph Combinatorial Model", *IEEE International Conference on Wire-less and Mobile Computing, Networking and Communications*, 2006.

[2]. Carla F. Chiasserini, Ramesh R. Rao, "A Distributed Power Management Policy for Wireless Ad Hoc Networks", *Proceedings of IEEE Wireless Communication and Networking Conference*, 2000.

- [3]. Suresh Singh, Mike Woo, C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", *Proceedings of MobiCom 98 Conference, Dallas, 1998*.
- [4]. D.S. Thenmozhi and M. Rajaram "Contention Aware Multi-hop Stable Routing to Provide Quality of Service Based on Multiple Constraints in Mobile Ad Hoc Networks" *European Journal of Scientific Research ISSN 1450-216X Vol.48 No.4 (2011), pp.567579 © EuroJournals Publishing, Inc. 2011*  
<http://www.eurojournals.com/ejsr.htm>
- [5]. Pramita Mitra, Christian Poellabauer, Shivajit Mohapatra "On Improving Dynamic Source Routing for Intermittently Available Nodes in MANETs" November 8, 2008
- [6]. Joseph Polastre, Robert Szewczyk, Cory Sharp, and David Culler. The mote revolution: Low power wireless sensor network devices. In *Proceedings of Hot Chips 16: A Symposium on High Performance Chips, Stanford, USA*, pages 56{76. IEEE Computer Society, 2004.
- [7]. M. W. S. Singh and C. S. Raghavendra. "Power-Aware Routing in Mobile Ad Hoc Networks". *International Conference on Mobile Computing and Networking, Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190, 1998.
- [8]. B. H. A. Nasipuri, R. Burleson and J. Roberts. "Performance of Hybrid Routing Protocols for Mobile Ad Hoc Networks". *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN2001)*, pages 296–302, 2001.
- [9]. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Mobile Computing*, ed. T. Imielinski and H. Korth, Kluwer Academic Publishers, pp. 153-181, 1996.
- [10]. V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of INFOCOM '97, April 1997*.
- [11]. C-K. Toh, "A Novel Distributed Routing Protocol To Support Ad-Hoc Mobile Computing," *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communication*, pp. 480-486, March 1996.
- [12]. Power-Aware Routing in Mobile Ad Hoc Networks – Suresh Singh, Mike Woo, C.S. Raghavendra
- [13]. S. Singh and C. S. Raghavendra, "PAMAS-Power Aware Multi-Access protocol with Signaling for Ad Hoc Networks," *ACM Commun. Rev.*, July 1998.
- [14]. K. Scott and N. Bambos, "Routing and Channel Assignment for Low Power Transmission in PCS," *Proc. ICUPC '96*, vol. 2, pp. 498-502, Oct. 1996.