

# Pre-emptive Dynamic Source Routing: A Repaired Backup Approach and Stability Based DSR with Multiple Routes

Susmit Maity, Soumen Saha, Sk. Sahnawaj, Barun Kumar Saha  
and C. T. Bhunia

Department of Computer Science and Informatics, Haldia Institute of Technology, India

DSR algorithm is applied to select the best path for communication between two nodes in a highly dynamic environment. As highly dynamic environment changes frequently, the probability of breakage of established path is high in such environment. Every time when established path breaks, a new route has to be discovered for communication between the nodes. In order to avoid path discovery every time, we propose the modification of the existing DSR algorithm. The modification proposed is based on a backup route (second best route). The backup routes will be provided by the destination node to the source node along with the best route during the process of path discovery. During the path maintenance process, in case any intermediate node identifies that the signal strength falls below a threshold indicating that the established route is about to break, the intermediate node sends a caution message to the source node. The source node switches the communication through the backup path, apprehending that the established route is about to break. As the communication through the backup route takes place, the previous established route is repaired, if possible. If repaired, the route will act as the new backup route. This process of toggling between backup route and established route reduces the call for path discovery to a large extent. The stability in consideration of failure of common link and nodes in the backup repaired algorithm has been investigated with new algorithm for stable route selection.

*Keywords:* MANET, DSR, Backup repaired, Stable route selection algorithm.

## 1. Introduction

A mobile ad-hoc network (MANET) is a self-configuring network of mobile routers (and associated hosts) connected by wireless links, the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's

wireless topology changes rapidly and unpredictably. Proactive MANET protocols are table-driven and will actively determine the layout of the network. Through a regular exchange of packets meant for network topology between the nodes of the network, a complete picture of the network is maintained at every node. Hence there is minimal delay in determining the route to be taken.

Reactive MANET protocols only find a route to the destination node when there is a need to send data. The source node will start by transmitting route requests throughout the network. The sender will then wait for the destination node or an intermediate node (that has a route to the destination) to respond with a list of intermediate nodes between the source and the destination. This is known as the global flood search, that in turn brings about a significant delay before the packet is transmitted.

Since each of the proactive and reactive routing protocols suits well in oppositely different scenarios, there is good reason to develop hybrid routing protocol that is a mix of both proactive and reactive routing protocols. The hybrid protocol is applied to find a balance between the proactive and the reactive protocols.

In Dynamic Source Routing, each source determines the route to be used in transmitting its packets to selected destinations. There are two main components, called Route Discovery and Route Maintenance. Route Discovery determines the optimum path for a transmission

between a given source and a destination. Route Maintenance ensures that the transmission path remains optimum and oscillation/loop-free as network conditions change, even if this requires changing the route during a transmission

Route Discovery is illustrated in Figure 1. Assume that node A (initiator) sends a Route Request Packet by flooding the network. If the node B has recently received another Route Request Packet from the same source, or is already listed in the Route Record, the node B discards the request. If node B is the target, it returns a Route Reply to the initiator along with the best path between the initiator and the target. When the initiator receives the Route Reply, it caches the route in its Route Cache for use in sending subsequent packets to the destination. Otherwise, node B is not the target, nor can it contribute to the best path from the initiator to the target and passes the Route Request Packet to its neighbor (except the initiator).

Under Route Maintenance in DSR every node is responsible for confirming that the next hop in the Source Route receives the packet. Also, each packet is only forwarded once by a node (hop-by-hop routing). If a packet is not received by a node, it is retransmitted up to some maximum number of times until a confirmation is received from the next hop. Only if retransmission fails, a Route Error message is sent to the initiator. Then that Source Route is discarded from its Route Cache.

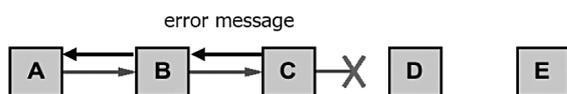


Figure 1. Route discovery illustration.

In Figure 1, if node C does not receive an acknowledgement from node D after a few requests, it returns a Route Error to the initiator A. As soon as node A (initiator) receives the Route Error message, it deletes the broken link-route from the cache. The initiator node A then repeats the Route Discovery Process.

In Dynamic Source Routing protocol, every time a node failure is detected, the existing route gets disconnected; and the source node has to initiate the Route Discovery process. In

a highly dynamic topology the failure of node is frequent, which results in making Route Discovery process a time consuming one. In the proposed approach we intend to reduce the number of times the Route Discovery process is to be initiated thereby increasing the efficiency of the routing protocol. In the proposed approach we find a best route from a source to a destination in terms of number of hops and a second best route (backup route). The second best route is stored in the route cache as a backup route. When any node in a first best route fails, the source starts transmitting packets through the backup route. Meanwhile, the first best route (now disconnected) is repaired. When repaired, it is stored in the route cache as backup route. By the process of swamping between route under used and backup route, the number of times the route discovery process is initiated is reduced considerably.

## 2. Proposed Algorithm

Route Discovery:

- 1) When a source node, say A wants to send a data packet, it broadcasts the RREQ packet to its neighboring nodes.
- 2) When an intermediate node on the route to the destination receives the RREQ packet, it appends its address to the route record in RREQ and rebroadcasts the RREQ.
- 3) When the destination node, say D receives the first RREQ, it starts a timer and collects RREQ from its neighbors until a quantum of time has expired.
- 4) The destination node D finds the best route as well as the second best route between source and destination.
- 5) The destination node D sends back a RREP to the source node along with the best route (Primary route) and the second best route (Secondary route).
- 6) As the source node receives the RREP it starts transmitting packets through the best route (Primary Route) and keeps the second best route (Backup Route) in the Route Cache.

Route Maintenance:

Route Monitoring:

1) Each intermediate node in the route through which transmitting is going on starts monitoring their signal strength. If signal strength falls below a specified threshold, it will send a warning message to the source node. The warning message signals about the possible breakage of route.

2) On receiving the warning message from the intermediate node, the source node picks the backup route from the Route Cache and starts transmitting through the backup route.

Route Repairing:

1) As transmission goes on through the backup route (now primary route) the nodes in the previous primary route (now broken route) are examined in order to locate the particular node node(s) that had failed.

2) The DSR protocol is used to find out a route between the previous node and any of the next (succeeding) nodes that had failed.

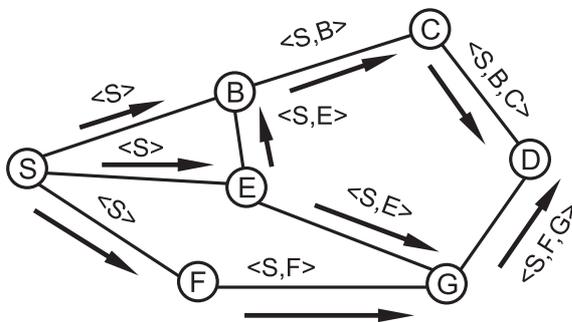


Figure 2. Example network for the illustration of proposed algorithm.

### Example:

When a node, say S in Figure 2 wants to send a message to the destination node, say D in Figure 2, it initiates route discovery by broadcasting the RREQ packet to its neighbors  $\langle B, E, F \rangle$ . The nodes  $\langle B, E, F \rangle$  on receiving the RREQ packet rebroadcast the packet to their neighbors by appending their id in the route record of the RREQ packet. Similarly, other intermediate nodes also forward the RREQ packets to their corresponding neighbors. When the destination node D receives two or more

RREQ packets from the same source, it finds out a best route and a second best route based on number of hops. In Figure 2, node D receives RREQ packets from routes  $\langle S, B, C, D \rangle$ ,  $\langle S, E, B, C, D \rangle$ ,  $\langle S, E, G, D \rangle$  and  $\langle S, F, G, D \rangle$  with number of hops of 3, 4, 3 and 3 respectively. Now, destination D may select any route in  $\langle S, B, C, D \rangle$ ,  $\langle S, E, G, D \rangle$  and  $\langle S, F, G, D \rangle$  as its primary route.

If destination D chooses  $\langle S, F, G, D \rangle$  as the primary route, the route  $\langle S, B, C, D \rangle$  will be the backup route. This is shown in Figure 3. The destination node D sends RREP packets to the source node S with the primary route and the backup route. When the source node receives the RREP packets, it starts communicating through the primary route  $\langle S, F, G, D \rangle$  keeping the backup route  $\langle S, B, C, D \rangle$  in the Route Cache.

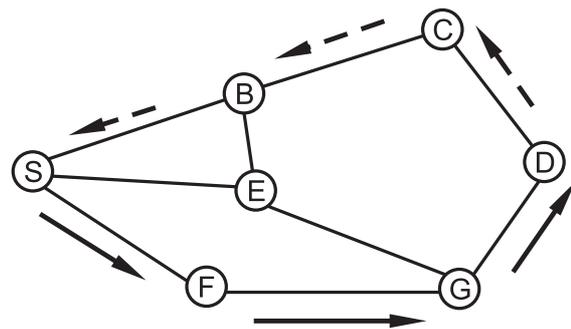


Figure 3. Primary and back up routes.

Primary Route  $\longrightarrow \langle S, F, G, D \rangle$

Backup Route  $- - \longrightarrow \langle S, B, C, D \rangle$

During transmission of packets, for example if the node F apprehends the signal strength falling below the threshold, it sends a warning message to the source node S. The source node S immediately switches the transmission of packets through the backup route  $\langle S, B, C, D \rangle$  and sends the previous primary route  $\langle S, F, G, D \rangle$  for repairing. As transmission goes on through the backup, the node S tries to find a route by searching a succeeding node of the failed node F by the DSR protocol. As a route from S to G is discovered, say as  $\langle S, E, G \rangle$ , the route  $\langle S, E, G, D \rangle$  is treated as the repaired route and is stored in the Route Cache of the source node S as the backup route. The process of repairing the broken route and toggling between the used primary route and the backup route continues whenever a node failure in a route is detected.

This minimizes the number of times the Route Discovery process is initiated in conventional DSR process. A comparison of the proposed algorithm with the conventional DSR is shown in Table 1.

Characteristics	DSR	Backup repaired DSR
Time required for getting route for transmission	High	Low
Route Caching	Yes	Yes
No of Routes	Single	Double
RREQ Flooding	High	Medium
Energy Saving	No	Yes
Mobility	Medium	High
Reliability	Low	High

Table 1. DSR vs repaired backup DSR.

### 3. Results

#### 3.1. Conventional DSR

Let  $n$  be the number of nodes contributing to the primary route (Best Route). Let  $p$  be the maximum probability of failure of a node in a network. Each node in the primary route has a probability of link failure and thereby breakage of the route. Since breakage at any one of  $n$  nodes causes the primary route to break, the probability of breakage of primary route  $P(pr)$  is given by:

$$P(pr) = {}^n c_1 p^1 (1-p)^{n-1}. \quad (1)$$

#### 3.2. Repaired backup approach DSR

In repaired backup approach when the primary route fails, the transmission continues on the backup route. So the probability for breakage of the primary route is same as conventional DSR given by equ.1.

Let  $m$  number of nodes contribute to the backup route (second best route). The probability of breakage of backup route  $P(bk)$  is then given as

$$P(bk) = {}^m c_1 p^1 (1-p)^{m-1}. \quad (2)$$

Since the primary route and backup route are independent of each other (as they do not have any

intersection of nodes), the probability of breakage of both the primary route and the backup route,  $P$  is given as

$$P = P(pr)XP(bk). \quad (3)$$

The net gain in probability of breakage in Repaired Backup Approach over conventional DSR using equ (1-2) is

$$\text{Net Gain}(g) = P(pr) - P. \quad (4)$$

The numerical results based on equ (4) for different sets of  $m$  and  $n$  are shown in Table 2 and in Figures (4-5).

$p$	Net Gain ( $n = 2,$ $m = 10$ )	Net Gain ( $n = 4,$ $m = 8$ )	Net Gain ( $n = 4,$ $m = 12$ )	Net Gain ( $n = 8,$ $m = 4$ )
0.2	0.23410 0654	0.2721 6105	0.32515 7507	0.19810 5367
0.4	0.46065 0824	0.3146 4132	0.33958 1632	0.05862 0838
0.6	0.47924 5025	0.1523 9204	0.15355 3614	0.0066 5636
0.8	0.31999 8689	0.025 5979	0.02559 9995	7.98228 E-05

Table 2. Net Gain of the proposed algorithm.

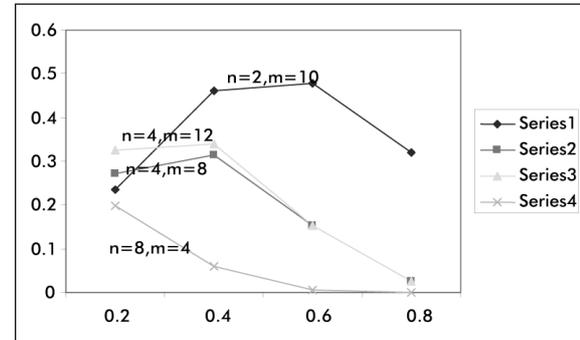


Figure 4. Net Gain for different values of  $n$  and  $m$ .

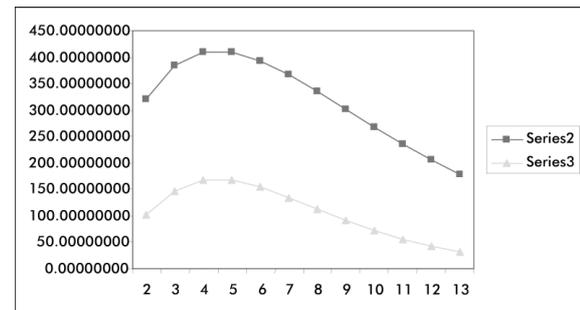


Figure 5. Probability to Net Gain for different values of  $n$  and  $m$  ( $P = 0.2$ ).

From Figures (4-5) it is concluded that

- i) The performance of repaired backup algorithm is always better than that of the conventional DSR.
- ii) Net Gain is greater for lower range of  $P$ ; and this is due to the fact that with lower  $P$ , the probability of breakage decreases.
- iii) Net Gain becomes maximum at some selected values of  $n$  and  $m$ . This is explained later.
- iv) There is no gain at higher values of  $n$  and  $m$  as then the probability of breakage for both the conventional and the repaired algorithms becomes equal.

We can find the condition at which gain,  $g$  becomes maximum. By putting  $\frac{dg}{dm} = 0$ , the maximum gain is obtained:

- i) when  $n$  and  $p$  are constant at

$$m = -\frac{1}{\ln(1-P)} \quad (5)$$

and

- ii) when  $m$  and  $p$  are constant at

$$n = -\frac{1}{\ln(1-P)}. \quad (6)$$

It is verified that the maximum gain as shown in Figures (4-5) are in conformity with equs (5-6). In fact, as  $P$  increases, the maximum value of  $m$  or  $n$  for which the gain becomes maximum will follow a curve as shown in Figure 6.

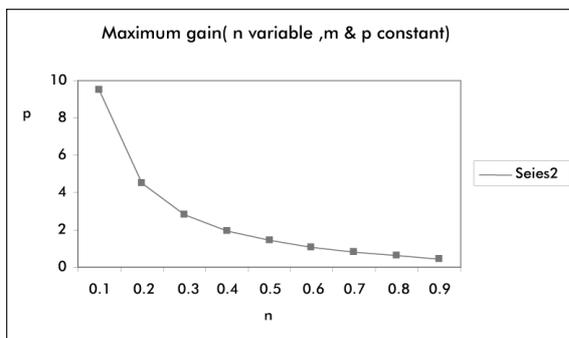


Figure 6.  $n(m)$  for maximum gain when  $m(n)$  and  $P$  are constants.

#### 4. New Algorithm

In our above stated work, stability of the route was not analyzed. Neither was the consideration of common links or node in a route addressed. In considering these issues, we investigated a few new algorithms.

##### Route Selection

The stability of a route used for routing depends on the mobility of the intermediate nodes constituting the route from the source node to the destination node. High mobility of the intermediate nodes implies that the nodes will stay at their current positions for a relatively small duration of time and therefore there is a high probability that the route will break at some intermediate position “very soon”. On the other hand, if intermediate nodes are less mobile, they are likely to stay at their current positions for a relatively greater interval of time and that would make the route more stable. Thus, we see that mobility of nodes directly affects the stability of a path.

Below we define certain terms and introduce some notations to be used further.

- **Node Stability:** It is defined as the probability of a node being stable. Let  $p_i$  denote the probability that node  $i$  is unstable. We measure this probability as

$$p_i = \frac{\text{Total motion time of node } i}{\text{Total motion time of node } i + \text{Total pause time of } i} \leq 1.$$

Pause time indicates the time duration for which the node remains stationary. The node stability is given by

$$q_i = 1 - p_i.$$

- **Link Stability:** It is the probability that the link  $i - j$  formed by nodes  $i$  and  $j$  is stable. Let  $q_i$  and  $q_j$  respectively denote the stability of nodes  $i$  and  $j$ . Then the link stability is given by

$$q_{ij} = q_i * q_j = (1 - p_i) * (1 - p_j).$$

- **Path Stability:** It is the probability of a path being stable. A path is stable if all the intermediate links are stable. Let  $\varphi_k$  denote the

stability of the path  $k$  and  $q_{kij}$  be the stability of the link  $ij$  along the path  $k$ . Then we have

$$\varphi_k = q_{k12} * q_{k23} * \dots * q_{k(n-1)n} = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{kij}.$$

- **Parallel Routes:** Two routes  $R_i$  and  $R_j$  are said to be parallel and can be used simultaneously if they have no common intermediate node(s) i.e. if

$$R_i = S - 1 - 2 - \dots - D,$$

$$R_j = S - 1' - 2' - \dots - D,$$

then

$$R_i \cap R_j = \{S, D\}.$$

We use the factor  $\varphi_k$  to determine the stability of a route during the Route Discovery phase of DSR. When the RREP message is traveling from the destination node to the source node, every intermediate node sends their stabilities ( $q_i$ ) along with the RREP message. The source node on receiving all  $q_i$  and  $q_j$  values along the path  $k$  computes the stability of the path. Alternatively, each node can cumulatively find out the stability of the path. This can be achieved by adding a PATH\_STABILITY header field to the RREP message. PATH\_STABILITY is initially set to 1 by the destination node. The subsequent nodes along the path update the PATH\_STABILITY header by multiplying with their own stability values  $q_i$ . This will reduce the computational burden of the source node.

After receiving all the RREPs, the source node then proceeds to determine the best path to the destination. For this purpose every node waits for a certain period of time waiting for multiple RREPs to arrive. Route length and stabilities can be known from these RREPs. Suppose the route  $k$  has  $n$  hops. Then length of the route is

$$L_k = n$$

and stability of the route is

$$\varphi_k = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{kij}.$$

We need not consider the stabilities of the source and the destination node because only the intermediate,  $n - 1$  nodes contribute to the route stability. The stable route is then the route having minimum number of hops as well as minimum instability among all received routes i.e.

$$\min_k (L_k, 1 - \varphi_k).$$

### Multiple Routes

Use of multiple routes simultaneously, instead of a single route at a time, would help to improve the ongoing communication between the two ends. In our algorithm, if multiple routes are found to have the same minimum value of  $(L_k, 1 - \varphi_k)$ , that would prove to be advantageous for the source node. The source node will use each of these routes alternatively to send packets to the destination node. Use of multiple routes reduces the dependency on a single route, which results in more stable communication. This is because, if a single route fails, we need to again initiate the Route Discovery process. However, if multiple routes are used, when one route fails, another route can be used. Only when all the routes fail, the Route Discovery is to be done to search a new route.

We note that the use of multiple routes is different from the backup route theory of DSR. In the backup route approach, the source node uses the primary route for communication and keeps a backup (secondary) route in its route cache. Whenever the primary route fails, the backup route is used. The problem with this approach is that, while the source is still using the primary route, the backup route might fail and the source would remain unaware of that. If after some time the primary route fails and the source node switches to the backup route, it discovers that the backup route has been already broken. But if multiple routes are used in parallel, the source node will be informed of the route failure immediately whenever it occurs. Thus, the source node will never attempt to use a stale route.

Even if multiple routes having the same minimum value of  $(L_k, 1 - \varphi_k)$  are not found, routes having close values to the minimum value can be used. Below, we show that if multiple routes are used, the probability of communication breakage decreases.

## 5. Related Analysis

To show that the use of multiple routes provide increased stability to the communication process, we make the following assumptions:

1. Number of intermediate nodes in a route between the source and destination nodes is always fixed, say  $n$ .
2. Every node has the same mobility and, therefore, every link has the same probability of breakage.
3. The lifecycle of each node consists of alternate periods of motion and rest. Each node moves for a fixed duration of time, say  $m$ , randomly and then remains in rest for a fixed duration of time  $p$ , which is the pause time. The probability of a node being stable at time  $T$  is then given by  
Stability = Total pause time up to time  $T/T$ .  
The instability of the node is  
Instability =  $1 - \text{Stability}$ .

### 5.1. Calculations

Consider a source node S and a destination node D, and a route  $R_0$  from S to D (that has  $n$  intermediate nodes). Below, we determine the probability of communication breakage when the single route  $R_0$  is used:

Probability that  $R_0$  will not fail ( $\varphi_0$ ) = Probability that all of the intermediate links will be stable

$$\Rightarrow \varphi_0 = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{0ij} = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{0i} * q_{0j}.$$

Therefore, the probability that communication will break = Probability that  $R_0$  will fail

$$\Rightarrow P_1 = 1 - \varphi_0 = 1 - \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{0ij} = 1 - \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{0i} * q_{0j}.$$

Now we consider the case when two parallel routes, say  $R_1$  and  $R_2$ , are available between S and D. According to our assumptions,  $R_1$  and

$R_2$  have  $n$  intermediate nodes each. The probability of communication breakage in this case is determined below:

Probability that  $R_1$  will not fail,

$$\varphi_1 = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{1ij} = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{1i} * q_{1j}.$$

Probability that  $R_2$  will not fail,

$$\varphi_2 = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{2ij} = \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{2i} * q_{2j}.$$

Hence, the probability that the communication will break = Probability that both  $R_1$  and  $R_2$  will fail

$$\begin{aligned} \Rightarrow P_2 &= (1 - \varphi_1) * (1 - \varphi_2) \\ &= \left(1 - \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{1ij}\right) * \left(1 - \prod_{\substack{i=1, \\ j=i+1}}^{n-1} q_{2ij}\right). \end{aligned}$$

For simplicity, we assume that the stability of each link is  $q$ . So, we have

$$q_{kij} = q, \forall i, j, k.$$

Hence, we have the probabilities

$$P_1 = 1 - \prod_{i=1}^{n-1} q = 1 - q^{n-1}$$

and

$$\begin{aligned} P_2 &= \left(1 - \prod_{i=1}^{n-1} q\right) * \left(1 - \prod_{i=1}^{n-1} q\right) \\ &= (1 - q^{n-1}) * (1 - q^{n-1}) \\ &= (1 - q^{n-1})^2 = (P_1)^2 \leq P_1, \end{aligned}$$

since  $1 - q^{n-1} \leq 1$ .

In general, for  $k$  such parallel routes we would have,

$$P_k = (P_1)^k.$$

So, the probability of communication breakage between S and D reduces exponentially if parallel routes are used. In other words, communication becomes more stable when multiple routes are used.

We note that if there were some intermediate nodes common to the routes  $R_1$  and  $R_2$ , then the existence of one route would be dependent, to some extent, on the other route. In fact, if the number of common intermediate nodes increases, the two parallel routes tend to become a single route. Hence, parallel routes should not have any common intermediate nodes.

## 5.2. Results and Discussions

Figure 7 shows the plot of  $P_k$  ( $k$  = no. of parallel routes) with respect to time. The different parameters of the plot are listed in Table 3.

No. of parallel routes ( $k$ )	1-5
No. of intermediate nodes ( $n$ )	5
Motion time	10s
Pause time	2s after every 10s
Total time	200s

Table 3. Plot Parameters

Figure 7 shows that when multiple parallel routes are used, the communication between the source node and the destination node reduces exponentially. That is, greater the number of parallel routes, the more stable the communication becomes. This is due to the fact that when multiple routes are used, dependency on a single route is reduced. Therefore, even if a single route fails, we have other routes in hand to use for transmitting packets. If a very long time period is considered, the fluctuation in the probability values stops and reaches a saturation level.

## 6. Conclusion

Reactive routing protocol initiates discovery only when the route breaks. It needs significant overhead for detecting the disconnection and the reconstruction of a new route. Repaired backup-based DSR detects early the link breakage, but may prove to be fatal if false warnings are generated. So the efficiency of the algorithm depends much on the selection of the value of the threshold ( $T$ ). An inappropriate selection of the values of threshold ( $T$ ) may reduce the efficiency of the algorithm. Future research may be

carried out over the calculation of the optimum value of the threshold ( $T$ ). Since the environment is highly dynamic by nature, there is a high probability that the backup route will fail by the time the primary route breaks. Route selection algorithm provides a stable and alternative (flexible and reliable) route selection compared to simple repaired backup scheme.

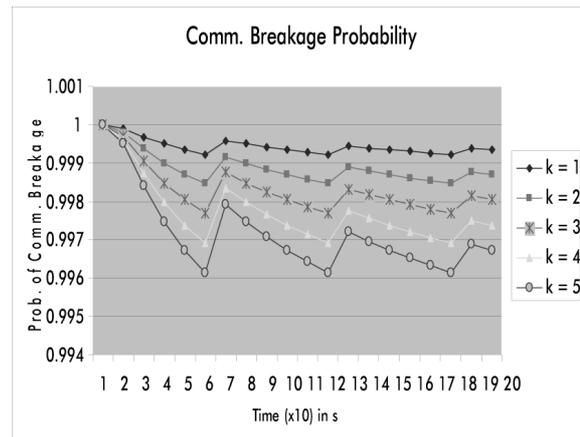


Figure 7. Probability of communication breakage decreases when parallel routes are used

## References

- [1] D. B. JOHNSON, D. A. MALTZ, Dynamic Source Routing in Ad-hoc Wireless Networks. *Mobile Computing* T. IMIELINSKI, H. KORTH, EDS. KULWER, (1996), pp. 151–181.
- [2] H. HASSANEIN, H. DU, C. YEH, Robust Route Establishment in High-mobility MANETs. *Proceedings ICENCO*, (2004).
- [3] E. M. ROYER, CHAT-KEONG TOH, A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Commn.*, **6**(2) (1999), pp. 46–55.
- [4] AD-HOC ROUTING PROTOCOLS (CLASSIFICATION): [http://wiki.uni.lu/secan-lab/Ad-hoc+Protocols+\(\\$28\)Classification\(\\$29\).html](http://wiki.uni.lu/secan-lab/Ad-hoc+Protocols+($28)Classification($29).html)

Received: June, 2007  
Accepted: November, 2007

Contact addresses:

Susmit Maity, Soumen Saha, Sk. Sahnawaj,  
Barun Kumar Saha and C T Bhunia  
Department of Computer Science and Informatics  
Haldia Institute of Technology  
PO: HIT  
Pin-721657  
India  
e-mail: ctbhunia@vsnl.com

---

SUSMIT MAITY received MCA from Burdwan University and M.Tech in information technology from the University of Calcutta in 2005. Presently, he is working as senior research fellow at Bengal Engineering & Science University, India. Earlier, he worked as senior lecturer at Haldia Institute of Technology.

---

---

SOUMEN SAHA is currently a lecturer of information technology at Haldia Institute of Technology. Prior to joining Haldia Institute of Technology, he worked at Bengal Institute of Technology & Management as a lecturer. He earned M.Tech and B.Tech in information technology from the University of Calcutta in 2005 and 2003 respectively.

---

---

SK. SAHNAWAJ HOSSAIN is currently a senior Sr. lecturer of master of computer application at Haldia Institute of Technology. He did MCA at Institute of Technology, Rourkela, India in 2001 and M.Sc. (App Math) from Burdwan University in 1996.

---

---

BARUN DAS received B. Tech. in information technology from Haldia Institute of Technology in 2007. Presently, he is working as a software engineer at Wipro.

---

---

CHANDAN T. BHUNIA earned his B.Tech in radio physics and electronics in 1983 from the Calcutta University, and then joined DVC of Govt. of India as telecommunication engineer. He got M.Tech in radio physics and electronics in 1985, and then joined North Bengal University as a lecturer of computer science & application in 1988, and became Assistant Professor of electronics & communication engineering at the North Eastern Regional Institute of Science & Technology (NERIST) of Govt of India in 1990. He got his Ph.D in computer science & engineering from the Jadavpur University. He became full Professor in 1997 at NERIST where he was HOD of ECE & CSE for about 6 yrs and Dean (Academics/Post Graduate Studies) for about 1.5 years. He then switched to private engineering colleges from 1999 to 2003 as HOD, Deputy Director and Director. Lastly, he was a full Professor of computer science & engineering of the Indian School of Mines (Deemed University) of Govt of India. He is currently senior Professor and Dy Director (Acad) of Haldia Institute of Technology. He has extensively visited foreign countries, namely China, Italy, Singapore, UK and Bangladesh on several assignments including BOYSCAST Fellowship and ICTP senior associateship. He has published around 200 research papers and technical articles/reports in national/international journals/magazines/seminars. He is the author of the books a) "Introduction to Knowledge Management" published by Everest Publishing House, Pune in 2003 and b) "Information Technology, Network and Internet", published by the New Age International Publishers, New Delhi. He is Fellow of the IETE and the IE(I), and a senior member of the IEEE & CSI. He is currently a member of International Advisory Board of Science Echoes, USA and an editor of the Journal of Mobile Communication.

---

