A Hybrid Protocol for Identification of a Maximal Set of Node Disjoint Paths in Mobile Ad hoc Networks

Ash Abbas
Department of Computer Engineering, Aligarh Muslim University, India

Abstract: *Identifying a maximal set of node-disjoint paths between a given source and a destination is a challenging task in mobile ad hoc networks. One cannot guarantee to identify the maximal set of node-disjoint paths in a single sequence of request-reply cycle. However, one can guarantee to identify a maximal set of node-disjoint paths in an incremental fashion using multiple route discoveries. In this paper, we present a protocol that adopts an approach that is a hybrid of the approaches taken by a protocol that tries to identify multiple node-disjoint paths in a single go and a protocol that identifies them incrementally. Our approach preserves the guarantee to discover a maximal set of node-disjoint paths between a given source and a destination. Further, we have shown that our approach is scalable and it requires less number of route discoveries than that required by an incremental protocol.*

Keywords: *Ad hoc networks, path diminution, maximal set, node-disjoint, multipath routing, multiple route discoveries, incremental protocols.*

Received April 6, 2007; accepted April 3, 2008

1. Introduction

An ad hoc network is a cooperative engagement of a collection of mobile devices without the required intervention of any centralized infrastructure or a centralized access point. The devices used to form such a network have limited transmission ranges, therefore, routes are often multihop. There are no separate routers, therefore, nodes in the network need to forward packets of one another towards their ultimate destinations. The devices are often powered through batteries, and as a result depletion of battery power may often cause failure of nodes as well as links. Further, node mobility makes the topology of the network highly dynamic. Therefore, routing is an important issue in ad hoc networks. Many researchers proposed routing protocols for mobile ad hoc networks, e.g., dynamic source routing [17], Ad hoc on-demand distance vector routing [22], *etc.,* These protocols provide a single path from a given source to a destination.

In such a network, providing a source with more than one paths can be quite useful, because if *a* path fails due to movement of an intermediate node, the communication may be continued through alternate paths. However, multiple paths can be better utilized if they satisfy some form of disjointness. On the basis of disjointness, multiple paths can be classified into the following categories: (i) node-disjoint (ii) link-disjoint, and (iii) neither node-disjoint nor link-disjoint. In node-disjoint paths, there are no common nodes except the source and destination. On the other hand, link-disjoint paths do not share any link but there can be common nodes. In other words, paths that satisfy node-disjointness also satisfy link-disjointness. Paths in the third category do not satisfy any kind of disjointness. We are mainly concerned with node-disjoint paths because using them one may address issues of fault-tolerance as well as load sharing. Also, if paths discovered between a pair of nodes are node-disjoint, frequency of route discovery is reduced [9], [8], [21], [26]. The throughput is likely to be improved if multiple node-disjoint paths are used simultaneously for data transfer between a given pair of nodes [23], [27]. Multiple node-disjoint paths can be useful in case of bursty traffic.

Identifying node-disjoint paths between a given pair of nodes is a challenging task in a mobile ad hoc network. Specifically, the identification of a maximal set of node-disjoint paths in a single route discovery cannot be guaranteed as this problem comes out to be an NP-complete problem [5]. However, one may provide guarantees using multiple route discoveries when the routes are discovered in an incremental fashion.

In this paper, we discuss a protocol to identify a maximal set of node-disjoint that adopts an approach which is hybrid of the approaches adopted by an incremental protocol and a multipath routing protocol that tries to discover as many paths as possible in a single route discovery.

The rest of this paper is organized as follows. Section 2 contains the problem formulation and major issues. In section 3, we discuss how the problem of finding node-disjoint paths is related to flow networks. In section 4, we briefly describe the approach adopted by an incremental protocol. In section 5, we discuss a protocol that is a hybrid of incremental and multipath approaches. In section 6, we analyze the number of RREQ transmission and the time to discover a maximal set of node-disjoint paths. Section 7 contains results and discussion. Finally, we conclude in section 8.

2. Problem Formulation

There are protocols [9], [23], [28] that try to identify multiple node-disjoint paths between a given pair of nodes. In all these protocols, an intermediate node forwards a Route REQuest (RREQ) according to a stated RREQ forwarding policy. In most of these protocols (except for the protocol described in [28]), the destination is responsible to compute the resulting node-disjoint paths. Further, all of them try to discover multiple node-disjoint paths in a single sequence of RREQ/RREP transmissions through the network. Such a sequence is referred to "single route discovery".

The schemes discussed in [5], [23], [28] try to identify multiple node-disjoint paths between a given pair of nodes with high probability in a single route discovery. However, no such scheme can guarantee that it will always be able to identify a maximal set of node-disjoint paths in a reasonable amount of time.

Providing a guarantee for a protocol that discovers multiple node-disjoint paths between a given pair of nodes in a single route discovery is very difficult. This is due to the following reasons.

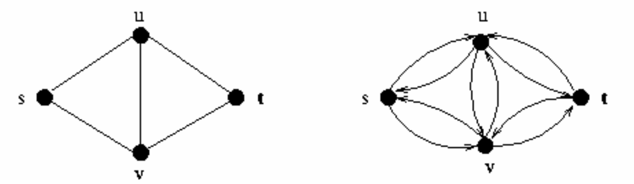
- It is difficult to devise an RREQ forwarding policy that forwards a limited number of copies of an RREQ such that all node-disjoint paths can be computed by the destination using traversed paths contained in different copies of the RREQ.
- If the number of copies of the RREQ that reach the destination is very large, the destination may or may not be able to determine a maximal set of node-disjoint paths in polynomial time.

Till date, there is neither an algorithm nor a protocol that claims to identify *a* maximal set of node-disjoint paths between a given source and a destination in *a* single route discovery in an ad hoc environment. In fact, it has been proved in [5] that computing a maximal set of node-disjoint paths, from a list of path traversed by different copies of an RREQ, either at the source or at the destination, is an NP-complete problem. Therefore, it is not possible to provide any guarantee about the identification of the maximal set of node-disjoint paths in a single route discovery[3]. However, some researchers have proposed some schemes to compute the maximal set of node-disjoint paths in multiple route discoveries and in an incremental fashion. A brief review of some of them is as follows.

A distributed algorithm to identify all node-disjoint paths between a given pair of nodes is proposed in [13]. The algorithm is a distributed version of a standard method of computing node-disjoint paths using a max-flow algorithm. The algorithm uses breadth first search and discovers one path at a time. Although, it has not been discussed that the algorithm is guaranteed to discover all node-disjoint paths, we anticipate that the algorithm is able to do so.

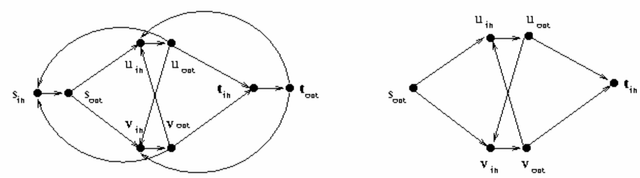
A graph theoretic framework to identify node-disjoint paths is proposed in [18], [19]. Based on the framework, the authors proposed a routing protocol called Multiple Node-Disjoint Path (MNDP). In the first route discovery, the protocol identifies a reference path using a single path routing such as Dynamic Source Routing (DSR) [17]. In the second route discovery, the protocol identifies an auxiliary path. The reference path and the auxiliary path are then inspected and reorganized to yield two node-disjoint paths. The authors have used MNDP to discover two node-disjoint paths. However, it can potentially be extended to find all node-disjoint paths that exist between a given pair of nodes, albeit in an incremental fashion. Further, the fact that MNDP is guaranteed to discover multiple node-disjoint paths is proved in [18] using concepts of flow networks.

In this paper, we present a protocol that employs a combination of the approaches presented in [17], [18] and that of MNDP [18]. We call the protocol Multiple Attempt Multipath Routing (MAMR) [21]. In MAMR, we try to discover as many paths as possible in the first route discovery using one of the schemes presented in [5]. Subsequent route discoveries identify paths in an incremental fashion using the approach of MNDP. We analyze the overheads in terms of the route request packet transmissions and the route discovery time.



(a) An undirected graph. (b) The corresponding directed graph.

Figure 1. A network.



(a) Node and edge replacements (b) s_{in} and t_{out} are deleted

Figure 2. Transformation.

3. Flow Networks and Disjoint Paths

An ad hoc network can be represented by an undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of bidirectional communication links. If each undirected edge is represented by two directed edges in opposite direction, the network graph becomes a directed graph as shown in Figure 1.

In order to find node-disjoint paths between the source node s and the destination node t, an undirected graph can be transformed into a unit capacity flow network (see [11] and [14]). The steps are as follows.

- Each node u (including s and t) is replaced by two subnodes u_{in} and u_{out} such that there is a directed link from u_{in} to u_{out}.
- Each undirected edge *(u,v)* is replaced by two directed edges $u_{out} \rightarrow v_{in}$ and $v_{out} \rightarrow u_{in}$.
- Delete s_{in} with all edges incoming to it. Delete t_{out} and all edges outgoing from it.
- Assign unit capacity to all links.

For the network shown in Figure 1, the transformation is shown in Figure 2. The problem of finding the maximum number of node-disjoint paths in the original network is equivalent to the problem of finding the maximum flow in the unit capacity flow network so constructed. Below, we describe a method to compute node-disjoint paths.

3.1. A Centralized Method

To find all node-disjoint paths between a given pair of nodes, one can find an augmenting path iteratively and assign a flow along it. The augmenting path found in an iteration may not be node-disjoint with the paths of the previous iteration. Therefore, after each iteration the augmenting path found has to be merged or reorganized with the paths of the previous iteration. This process can be repeated until no more augmenting path can be found. Based on the max-flow min-cut theorem [16], this method is guaranteed to identify all node-disjoint paths between a given pair of nodes.

Comment: the above method requires that the topology information is available at a centralized node. In an ad hoc network, a node knows the topology only partially. Specifically a node knows of nodes in its neighbourhood. Therefore, in an ad hoc network, a method that does not rely on the global topology information is needed. In what follows, we discuss the modifications to the above method so that the protocol does not need the global topology information.

3.2. A Distributed Method

A method that is distributed and is based on the local topology information has been proposed in [18]. The authors, therein, has transformed it to the extent of a protocol called MNDP. The MNDP protocol is basically based on flow networks and is proved to provide guarantees[4] to discover *all* node-disjoint paths, albeit in an incremental fashion, between a given pair of nodes.

Note that MNDP requires a single path routing protocol to identify an initial reference path in the first route discovery. In what follows, we present a version of MNDP that does not need a single path routing protocol.

4. A Version of MNDP

In this section, we describe a version of MNDP, an incremental routing protocol. The protocol MNDP is an on-demand routing protocol. Recall that an on-demand protocol has two major steps: route discovery and route maintenance. These steps are described as follows.

Table 1. Second route discovery (*ForbiddenPathSet = {p₁, p₂}* where $P_1: < s, 1, 12, 7, d >$ and $P_2: < s, 2, 11, 10, 9, d >$).

Node	Condition	Action	*CurrentPath*
s	*source*	-	-
3	$3 \notin \{P_1, P_2\}$	broadcast	$< s, 3 >$
11	$11 \in P_2$ *	unicast	$< s, 3, 11 >$
2	$2 \in P_2$ □	broadcast	$< s, 3, 11, 2 >$
12	$12 \in P_1$ *	unicast	$< s, 3, 11, 2, 12 >$
1	$1 \in P_1$ □	broadcast	$< s, 3, 11, 2, 12, 1 >$
13	$13 \notin \{P_1, P_2\}$	broadcast	$< s, 3, 11, 2, 12, 1, 13 >$
15	$15 \notin \{P_1, P_2\}$	broadcast	$< s, 3, 11, 2, 12, 1, 13, 15 >$
d	*destination*	-	$< s, 3, 11, 2, 12, 1, 13, 15, d >$
* neither predecessor nor successor			
□ successor			

4.1. Route Discovery

In each route discovery there are four major steps:

- Initiation by the source.
- Processing of RREQ at intermediate nodes.
- Reply by the destination.
- Reorganization at the source.

(a) Initiation by the source to initiate a route discovery, the source broadcasts an RREQ. An RREQ contains the following information in its header:

<SourceAddress, DestinationAddress, SourceSeqNo, ForbiddenPathSet, CurrentPath>.

The ForbiddenPathSet is a set of paths discovered before the beginning of a new route discovery. The source checks its RouteCache for a set of routes to the destination. If number of routes in RouteCache is less than a desired number, the source copies all paths to the destination in ForbiddenPathSet of the RREQ. It then initiates a new route discovery with an empty CurrentPath.

(b) Processing of the RREQ at an intermediate node Processing of an RREQ at an intermediate node is similar to that of MNDP described in [18]. Following this way, the RREQ reaches the destination.

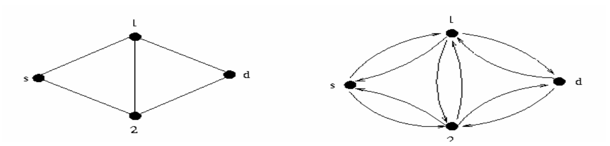
(c) Processing of RREQ at the destination upon receiving the first RREQ, the destination sends an RREP towards the source along the CurrentPath. The destination discards other copies of the RREQ. Note that CurrentPath may not be node-disjoint with the paths in RouteCache. When source receives the RREP, it reorganizes previous set of node-disjoint paths and CurrentPath as follows.

(d) Reorganization at the source Denote the set of node-disjoint paths in RouteCache and CurrentPath by a directed graph $\mathcal{H}=(\mathcal{V}_{\mathcal{H}},\mathcal{E}_{\mathcal{H}})$. If for any edge $(u,v)\in\mathcal{E}_{\mathcal{H}}$ if (v,u) is also in $\mathcal{E}_{\mathcal{H}}$, remove (u,v) as well as (v,u) from $\mathcal{E}_{\mathcal{H}}$. The resulting graph gives new set of node-disjoint paths. The source stores them in its RouteCache.

We wish to see how the above protocol is able to discover node-disjoint paths where other protocols fail. This is illustrated in the following examples.

Example 1: Figure 3 (a) shows a small network with bidirectional links. Figure 3 (b) shows the same network in which each bidirectional link is replaced by two unidirectional links in opposite directions. Let the path discovered when the protocol is executed the first time be <s, 1, 2, d> as shown in Figure 4 (a). The source places it in the ForbiddenPath and initiates a new route discovery. Major steps are as follows.

- Node 1 discards the RREQ from node s because s is its predecessor on ForbiddenPath.
- Node 2 is on the ForbiddenPath but s from which it received a copy of the RREQ is neither its predecessor nor successor on ForbiddenPath, it appends its own address on the CurrentPath and unicasts it to its predecessor (node 1) on the ForbiddenPath.

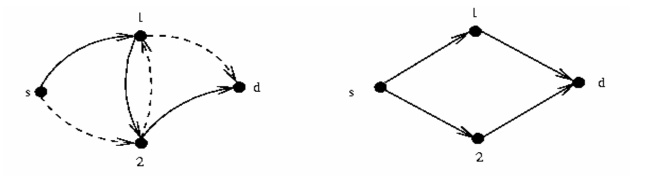


(a) A small network with 1 bidirectional links. (b) Another representation.

Figure 3. Example.

(a) ForbiddenPath = <s,1,2,d>. (b) CurrentPath = <s,2,1,d>.

Figure 4. Example 1.

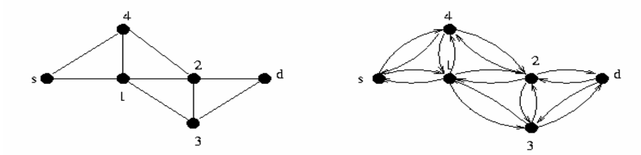


(a) Reorganization. (b) Final set of node-disjoint paths.

Figure 5. Example 1.

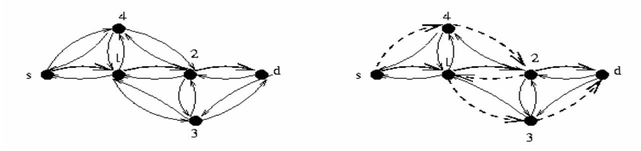
- Node 1 is on the ForbiddenPath and it has now received the copy of the RREQ from its successor. Therefore, node 1 broadcasts it to its neighbors. As a result, the CurrentPath becomes <s, 2, 1, d> as shown in Figure 4 (b).
- After reorganization (as shown in Figure 5(a)), there are two node-disjoint paths: <*s*,1,*d*> and <*s*,2,*d*> (Figure 5(b)).

No other path can be found in the next route discovery. Therefore, the route discovery is terminated.



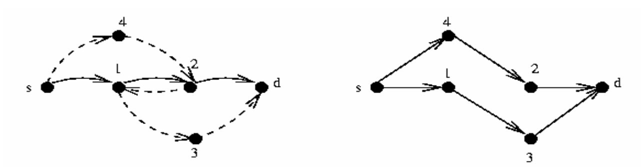
(a) Another network with bidirectional links. (b) Equivalent representation.

Figure 6. Example 2.



(a) ForbiddenPath = <s,1,2,d>. (b) CurrentPath = <s,4,2,1,d>.

Figure 7. Example 2.



(a) Reorganization (b) Final set of node-disjoint paths.

Figure 8. Example 2.

Example 2: Figure 6 shows another network. Let the path discovered in the first route discovery be <s, 1, 2, d>. This serves as ForbiddenPath for next route discovery whose major steps are as follows.

- Node 1 discards the RREQ from node s because s is its predecessor on ForbiddenPath.
- Node 4 receives a copy of the RREQ from node s. Since node 4 is not on ForbiddenPath, it appends its own address to the CurrentPath and broadcasts the RREQ to its neighbours.

- Node 2 receives a copy of the RREQ from node 4 which is neither its successor nor its predecessor on the ForbiddenPath. Therefore, node 2 appends its own address on the CurrentPath and unicasts the RREQ to its predecessor (node 1) on the ForbiddenPath.
- Node 1 has now received the RREQ from its successor and it is on the ForbiddenPath. Therefore, node 1 appends its own address on the CurrentPath and broadcasts the RREQ to its neighbours.
- Node 3 is not on the ForbiddenPath. Therefore, it appends its own address on the CurrentPath and broadcasts the RREQ to its neighbours.
- The CurrentPath is <s ,4 ,2 ,1 ,3 ,d> as shown in Figure 7(b). After reorganization, there are two node-disjoint paths: <s,4,2,d> and <*s*,1,3,*d*>.

No other path can be found in the next route discovery. Therefore, the route discovery is terminated.
In what follows, we briefly discuss the route maintenance phase.

4.2. Route Maintenance

If a link along a path fails, a node that senses the link failure downstream sends a Route ERRor (RERR) message upstream. Upon receiving an RERR message, a source may initiate a new route discovery if it requires a path over and above those paths that have not yet failed. If there is no path to the destination and the source have packets to send, the source has to initiate a new route discovery.

In what follows, we discuss a protocol that adopts an approach that is a hybrid of the approaches taken by a protocol that tries to discover multiple paths in a single route discovery and a protocol that tries to discover them incrementally.

5. A Hybrid Protocol

The MNDP protocol tries to discover paths one at a time and in an incremental fashion. In other words, after the end of each route discovery, it tries to increment the set of paths by exactly one path, if any. Therefore, if there exist *k* node disjoint paths between a given pair of nodes, exactly *k* route discoveries are required to discover them. On the other hand, there are protocols (such as those described in [5], [23], [28]) that try to discover multiple node-disjoint paths in a single route discovery.

To take the advantage of both these approaches, we propose a protocol that adopts an approach that is hybrid of the approaches taken by MNDP and that described in [5]. In what follows, we describe a hybrid approach.

Algorithm 1: Processing of RREQ at the source node in MAMR

```
if IncrementedFlag = T then
    Generate an RREQ such that RREQ.ForbiddenPathSet =
   RouteCache.PathSet and initiate a route discovery
   if RREP is received from the destination then
         Reorganize RREQ.ForbiddenPathSet with
         RREP.CurrentPath to get new set
       of node-disjoint paths NewPathSet
      if |RouteCache.PathSet|<|NewPathSet| then
  |RouteCache.PathSet|=|NewPathSet| and
         IncrementedFlag
         Go to Step 1
    else
         IncrementedFlag = F
         End of route discovery
   endif
 endif
 endif
```

Algorithm 2: Processing of RREQ at the destination node in MAMR

```
 if RREQ.ForbiddenPathSet = ϕ then
       Compute node-disjoint paths from RREQ.PathTraversed
       Send multiple RREPs one along each path
 else
      Send an RREP along RREQ.CurrentPath
endif
```

Algorithm 3: Processing of RREQ at an intermediate node in MAMR

```
 if RREQ.ForbiddenPathSet = ϕ then
       Process RREQ as in OFC (see [18])
else
       Process RREQ as in MNDP
 endif
```

5.1. Hybrid Approach

The hybrid approach is as follows. In the first route discovery, we discover as many paths as possible. To do so, we use a forwarding policy known as Only First Copy (OFC). In OFC, an intermediate node forwards only the first copy of an RREQ and discards other copies of the same RREQ. The destination computes a maximal set of node-disjoint paths and sends multiple RREPs, one along each path. Note that if we employ OFC, the reorganization step is not required in the first route discovery because the ForbiddenPathSet is empty. In each subsequent route discovery, forwarding of an RREQ is the same as in MNDP described in section 4. In other words, each subsequent route discovery discovers only one path if it exists between the given pair of nodes. The source reorganizes the CurrentPath together with the set of paths in ForbiddenPathSet to yield a set of node-disjoint paths discovered in the current route discovery. The set of node-disjoint paths after reorganization in the last route discovery yields the final set of node-disjoint paths. We call this protocol, which adopts an approach that is a combination of an incremental approach and an approach that is adopted by a multipath routing protocol, as MAMR. In case of MAMR, processing of an RREQ at the source is shown in Algorithm 1, and that at the destination node is shown in Algorithm 2.

In a hybrid protocol such as MAMR, an intermediate node needs to differentiate between the first route discovery and a subsequent route discovery so as to take different actions accordingly. To ensure whether an RREQ belongs to the first route discovery or not, a node checks ForbiddenPathSet. If ForbiddenPathSet is empty then the current route discovery is the first route discovery. Otherwise, it is a subsequent route discovery. Processing of the RREQ at an intermediate node is shown in algorithm 3.

If we discover multiple node-disjoint paths in the first route discovery using OFC [5], the number of route discoveries is reduced. Specifically, the number of route discoveries required is $k - k_1 + 1$ where k_1 the number of paths is discovered in the first route discovery using OFC. Note that in OFC, the destination has to compute a maximal set of node-disjoint paths using the heuristic described in [5]. However, the fact that the set of paths may not be a maximal set would not affect the guarantee provided by the protocol. The paths that are not identified in the first route discovery will certainly be identified in subsequent route discoveries. A suboptimal path set identified in the first route discovery can only increase number of subsequent route discoveries. In fact, by using the principle of mathematical induction, one can argue that there will be no effect of using any of the scheme in the first route discovery, and/or paths and the number of paths returned in it, on the guarantee of identifying a maximal set of node-disjoint paths.

We discuss an example where multiple node-disjoint paths are discovered in the first route discovery. In each subsequent route discovery, one of the remaining paths is identified.

Example 3: consider a network shown in Figure 9. In the first route discovery, two node-disjoint paths are discovered between the source and the destination. These are <*s*,1,12,17,*d*> and <*s*,2,11,10,9,*d*> as shown in Figure 10 The source places them in ForbiddenPathSet and initiates the second route discovery. Table 2 shows the processing of RREQ at different nodes in the second route discovery.

Table 2. Third route discovery (*ForbiddenPathSet* = { p_1, p_2, p_3} where P_1 : < *s*, 1, 13, 15, *d* >, P_2 : < *s*, 2, 12, 17, *d* >, and p_3: < *s*, 3, 11, 10, 9, *d* >).

Node	Condition	Action	CurrentPath
s	*source*	-	-
4	$4 \notin \{P_1, P_2, P_3\}$	broadcast	< *s*, 4 >
5	$5 \notin \{P_1, P_2, P_3\}$	broadcast	< *s*, 4, 5 >
6	$6 \notin \{P_1, P_2, P_3\}$	broadcast	< *s*, 4, 5, 6 >
10	$10 \in P_3$ *	unicast	< *s*, 4, 5, 6, 10 >
11	$11 \in P_3$ □	broadcast	< *s*, 4, 5, 6, 10, 11 >
17	$17 \in P_2$ *	unicast	< *s*, 4, 5, 6, 10, 11, 17 >
12	$12 \in P_2$ □	broadcast	< *s*, 4, 5, 6, 10, 11, 17, 12 >
15	$15 \in P_1$ *	unicast	< *s*, 4, 5, 6, 10, 11, 17, 12, 15 >
13	$13 \notin \{P_1, P_2, P_3\}$	broadcast	< *s*, 4, 5, 6, 10, 11, 17, 12, 15, 13 >
14	$14 \notin \{P_1, P_2, P_3\}$	broadcast	< *s*, 4, 5, 6, 10, 11, 17, 12, 15, 13, 14 >
16	$16 \notin \{P_1, P_2, P_3\}$	broadcast	< *s*, 4, 5, 6, 10, 11, 17, 12, 15, 13, 14, 16 >
d	*destination*	-	< *s*, 4, 5, 6, 10, 11, 17, 12, 15, 13, 14, 16, *d* >

* neither predecessor nor successor
□ successor

A copy of the RREQ with CurrentPath <s,3,11,2,12, 1,13,15,d> reaches the destination before any other copy. The destination sends an RREP against this copy of the RREQ. The source reorganizes the CurrentPath and the set of paths in ForbiddenPathSet as shown in Figures 12 and 13 show the set of node-disjoint paths after reorganization in the second route discovery. These paths are: <s,1,13,15,d>, <s,2,12,17,d> and <s,3,11,10,9,d>.

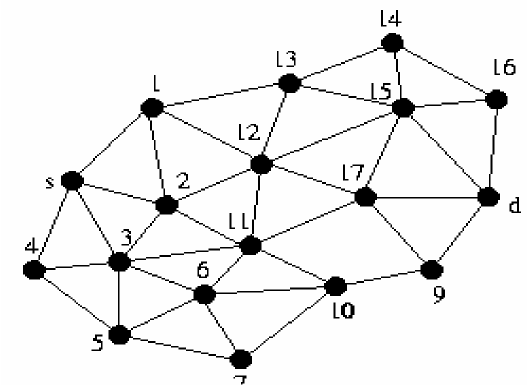


Figure 9. A network with bidirectional links to illustrate the hybrid approach.

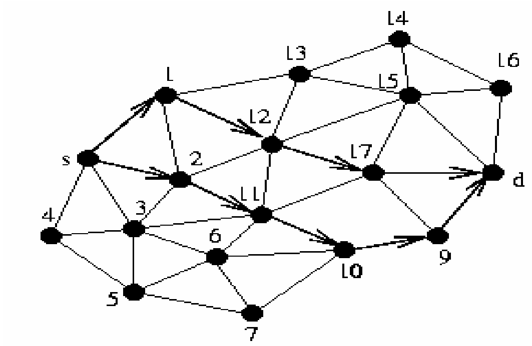


Figure 10. Set of node-disjoint paths discovered in first route discovery.

For third route discovery, the set of node-disjoint paths discovered after second route discovery serves as ForbiddenPathSet. Processing of RREQ in third route discovery is shown in Table 3. The path discovered is <s,4,5,6,10,11,17,12,15,13,14,16, d>, as shown in Figure 14. After reorganization (shown in Figure 15), the set of node-disjoint paths is: <s,1,13,14,16,d>, <s,2,12,15,d>, <s,3,11,17,d>, <s,5,6,10,9,d>. No other path can be discovered.

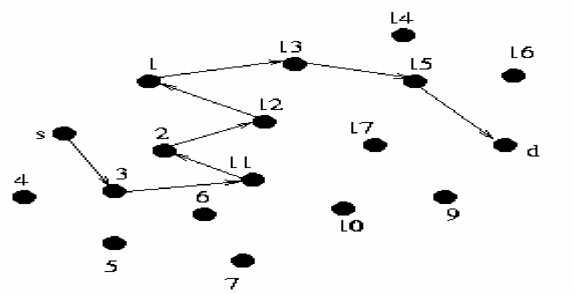


Figure 11. CurrentPath discovered in second route discovery.

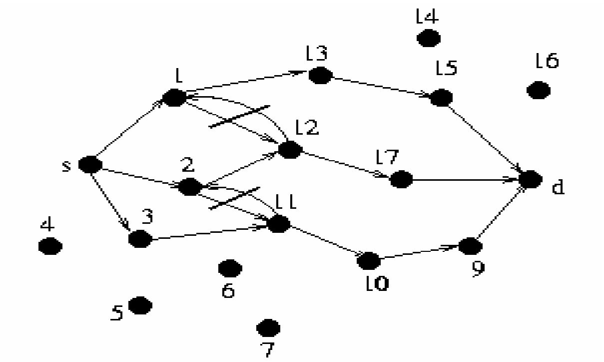


Figure 12. Reorganization during second route discovery.

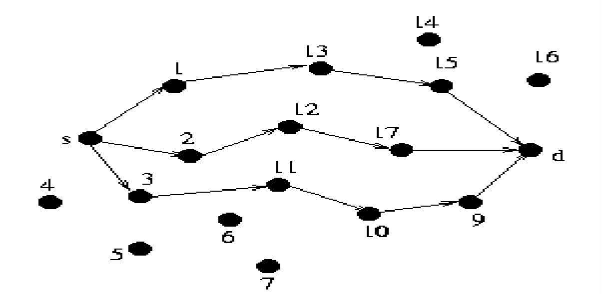


Figure 13. The set of node-disjoint paths after reorganization in second route discovery.

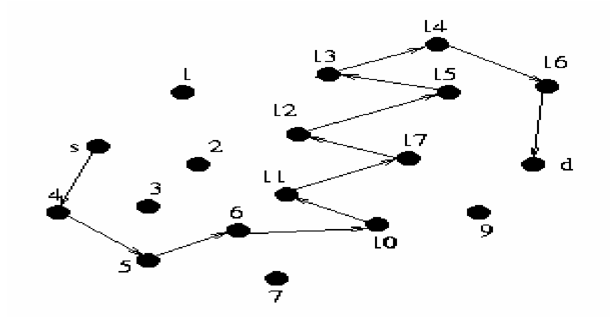


Figure 14. CurrentPath discovered in third route discovery.

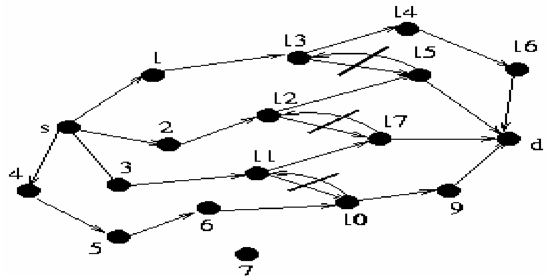


Figure 15. Reorganization during third route discovery.

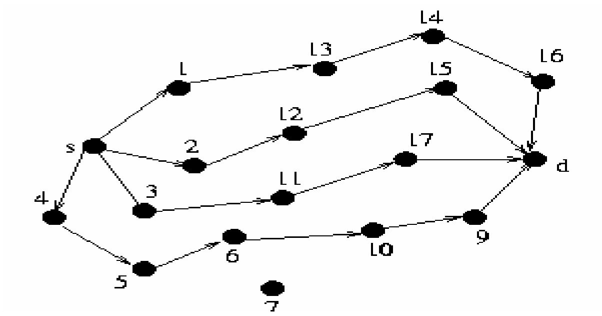


Figure 16. Final set of node-disjoint paths after reorganization in third route discovery.

Therefore, this is the final set of node-disjoint paths from the source to the destination. In MNDP as well as MAMR, if the source needs to discover more paths and it has initiated a route discovery, it cannot use the paths identified in the previous route discovery because that set of node-disjoint paths may change in the reorganization phase. In other words, the set of node-disjoint paths can be used by the source only if it does not need to discover more node-disjoint paths.

There are RREQ forwarding schemes that try to identify multiple node-disjoint paths in a single route discovery. These schemes try to identify them with high probability. However, there is no guarantee about identification of a maximal set of node-disjoint paths that exist between a given pair of nodes. Two such schemes are: (i) All Disjoint Copies (ADC), and (ii) at most One Copy per Neighbour (OCN) [5]. In ADC, an intermediate node forwards the first copy of an RREQ as such, and copies of the RREQ whose path traversed is disjoint with the copies already forwarded by the intermediate node. In OCN, an intermediate node may forward at most one copy per neighbour. In both of these schemes, all other copies of the RREQ are simply discarded. The destination is responsible for computation of node-disjoint paths and sending the RREPs so as to inform the source about the computed node-disjoint paths.

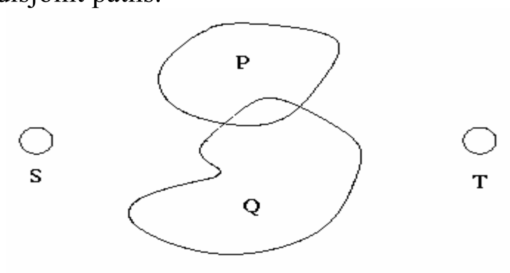


Figure 17. A scenario after the first route discovery and before the reorganization phase.

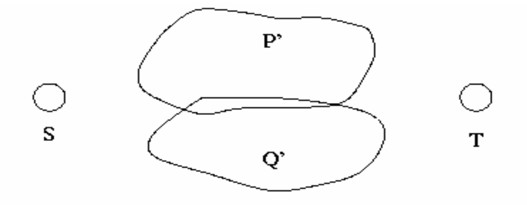


Figure 18. A scenario after the first route discovery and after the reorganization phase.

Having a background and understanding of both an incremental approach and the hybrid approach described above, in the next subsection, we show that the guarantees provided by the incremental protocol are preserved in the hybrid approach.

5.2. Preserving Guarantees

We wish to show that in a hybrid protocol such as MAMR, one can preserve the guarantees of discovery of a maximal set of node-disjoint paths irrespective of the number of node-disjoint paths discovered in the first route discovery and irrespective of the strategy used to discover multiple node-disjoint paths in the first route discovery. We state and prove the above statement as a theorem.

Theorem 1: the hybrid approach preserves the guarantees of discovering a maximal set of node-disjoint paths irrespective of the strategy used in the first route discovery.

Proof: let the set of paths discovered in the first route discovery be P as shown in Figure 17. Let the set of paths that still need to be discovered be Q. The path that is currently discovered is reorganized with paths that are in P. After the reorganization phase the set of paths will either be incremented by one or will remain the same. In the later case, there will be no further route discovery. In the former case, there shall be subsequent route discoveries and after each route discovery a change in the number of paths after the reorganization phase shall decide whether the next route discovery shall be initiated or not. Let the maximal set of node-disjoint paths be a finite set M.

Let us use any strategy in the first route discovery to discover as many paths as possible. The strategy tries to discover as many paths as possible, however, there is no guarantee of discovering the maximal set (or always a fixed number of node-disjoint paths, in general) of node-disjoint path. At this time instant, the strategy used discovers a set of paths, say P such that $|P| \leq |M|$. Now, there are two cases.

Case 1: if $|P|=|M|$, no subsequent route discovery is required, the maximal set of node-disjoint paths has been discovered.

Case 2: if $|P|<|M|$, there shall be one or more subsequent route discoveries using an incremental protocol. After the first subsequent route discovery, let the set of paths after the reorganization phase be P' as shown in Figure 18. Then, P' is such that either $|P'|=|P|$ (no subsequent node-disjoint path exists) or $|P'|=|P|+1$, (when a path is added to P). Whatever is the value of $|P|$, the set of node-disjoint paths after reorganization phase in the subsequent route discovery, P' is deterministically determined. Even if there is no guarantee for fixed value of $|P|$, but the paths that remain after P are determined deterministically. Further, after each subsequent route discovery P is augmented by exactly one path until it reaches M. Since the set M is finite, therefore, the route discovery process will terminate deterministically after a finite number of steps. Therefore, there is guarantee that the remaining paths of the set M-P will be determined deterministically in an incremental fashion. This makes the guarantee that the maximal set of node-disjoint paths M will be determined irrespective of the set P that was returned by a strategy used in the first route discovery. This completes the proof. In what follows, we analyze MNDP and MAMR and compare them with the schemes presented in [18].

6. Analysis

We wish to analyze three parameters: (a) number of RREQ transmissions, (b) route discovery time, and (c) route failure time. On the basis of these parameters, we compare the following protocols (i) MNDP, (ii) MAMR, and (iii) ADC/OCN.

Assume that the network is represented by an undirected graph $G = (V, E)$, where $|V| = n$ and $|E| = m$. Let there exist k node-disjoint paths between a given pair of nodes.

6.1. Number of RREQ Transmissions

In case of MNDP, only one path is discovered in the first route discovery. Each subsequent route discovery adds one path if it exist between the given pair of nodes. In other words, k route discoveries are required to discover k node-disjoint paths. Further, each node broadcasts (or unicasts) the RREQ at most once in each route discovery. As a result, the number of RREQ transmissions by a node is at most k. The total number of RREQ transmissions in the network is $O(kn)$ as shown in Table 3.

In case of MAMR, the destination computes multiple node-disjoint paths in the first route discovery. The remaining paths are computed incrementally. The number of RREQ transmissions required in the first route discovery is $O(n)$. The number of RREQ transmissions required in each subsequent route discovery is $O(n)$. Therefore, the total number of RREQ transmissions is $O(kn)$. In other words, the upper bound on the number of RREQ transmissions in case of MNDP and MAMR is the same. Specifically, MAMR discovers more than one node-disjoint paths in the first route discovery, the number of RREQ transmissions required by it are less than that required by MNDP.

In case of MAMR, the destination has to compute multiple node-disjoint paths in the first route discovery. As discussed in [5], the overhead incurred in computing disjointness at the destination is $O(zn\log n + z^2 n)$, where z is the number of copies of the RREQ received by the destination. Note that the reorganization step requires $O(n^2)$ computation effort at the source. In case of MAMR, since the number of route discoveries is reduced, the number of reorganization steps is also reduced. As a result the computational overhead incurred in reorganization as a whole is less in MAMR as compared to MNDP. Further, the number of RREQ transmission is reduced in MAMR as compared to MNDP.

Table 3. The number of RREQ transmissions for different protocols.

	MNDP	MAMR*	ADC/OCN#
First Route Discovery	O(n)	O(n)	O(kn)
Each Subsequent Route Discovery	O(n)	O(n)	no subsequent route discovery
Total	O(kn)	O(kn)	O(kn)

* Assuming the use of OFC to compute multiple node-disjoint paths in the first route discovery
# no Guarantee of discovering all node-disjoint paths.

In case of ADC or OCN, each node may transmit b copies of the RREQ. The total number of RREQ transmissions are $O(bn)$, where b is number of neighbours of a node. Since $b \approx k$, the number of RREQ transmissions are $O(kn)$. The protocols MNDP as well as MAMR are guaranteed to discover all node-disjoint paths. However, no such guarantee is provided by ADC or OCN. The number RREQ transmissions for different protocols are shown in Table 3.

Table 4. The number of RREQ transmissions for different protocols.

	MNDP	MAMR	ADC/OCN
First Route Discovery	t_{rd}	t_{rd}	t_{rd}
Each Subsequent Route Discovery	t_{rd}	t_{rd}	no subsequent route discovery
Total	kt_{rd}	$\leq kt_{rd}$	t_{rd}

6.2. Route Discovery Time

We now compare the route discovery time of MAMR and MNDP with that of ADC or OCN. Let t_{rd} denote the route discovery time of one route discovery. In case of ADC or OCN, the route discovery time is t_{rd}.

In case of MNDP, the route discovery time is kt_{rd}. Recall that the number of route discoveries in MAMR are $k - k_1 + 1$, where k_1 is the number of multiple node-disjoint paths discovered in the first route discovery. Therefore, route discovery time in case of MAMR is $(k - k_1 + 1)t_{rd}$ which is upper bounded by kt_{rd}. As a conclusion, we can say that route discovery time of ADC or OCN is less than or equal to that of MAMR which is further less than or equal to that of MNDP. The route discovery time for different protocols is summarized in Table 4.

6.3. Scalability Analysis

We wish to analyze whether MAMR scales well with the number of source-destination pairs and with the number of node-disjoint paths available in the network. For that, we carried out simulations. It is customary to consider only those topologies of a network where there exist at least k node-disjoint paths among every pair of nodes.

Note that, a network that is k-connected shall provide k node-disjoint paths between every pair of nodes. A topology of the network that is not able to pass k connectivity test has to be discarded. The procedure to determine whether a topology of the network has passed the k-connectivity test is as follows. Generate a trial topology. If for every pair of nodes there exist k node-disjoint paths in the trial topology, the network is k-connected. To determine whether there exist k node-disjoint paths between a given pair of nodes, transform the network into a flow network. The number of node-disjoint paths is equal to the value of max flow in the unit capacity flow network.

We then assumed that one is able to determine $k_1 \leq k$ paths in the first route discovery using OFC. The number k_1 is randomly and uniformly distributed between 1 and *MaxPaths*. The variable *MaxPaths* denotes the maximum number of paths that may exist between a given pair of nodes in the network.

We computed the number of route discoveries required by MAMR to find *k* node-disjoint paths between a given source and a destination. In other words, we computed the value of $k - k_1 + 1$, when k_1 is a uniformly distributed random number lying between 1 and *k*. We repeated the simulation and then averaged out for *l* source-destination pairs. We observed that average number of route discoveries for $k = 2$ and $l = 10$ is 1.5. It is almost same for $l = 20, 30, ..., 100$. For $k = 3$, it is around 2.0 irrespective of the value of *l*. For $k = 4$, it is 2.5, and for $k = 5$, it comes out to be 3.0 [7]. It means that if there is an increase in the number of node-disjoint paths in the network, one would be able to identify the remaining paths left in the first route discovery in subsequent route discoveries. As a result one may conclude that there is no effect either of the number of node-disjoint paths or of the number of source-destination pairs on the guarantee provided by MNDP, and consequently by MAMR, in identifying all node-disjoint paths that exist between a given source and a destination.

6.4. Route Failure Time

In the following, we analyze the route failure time of MNDP, MAMR and ADC/OCN. We divide the analysis into two parts. In the first part, we analyze the route failure time of these protocols assuming that individual lifetimes are exponentially distributed random variables. In the second part, we analyze the route failure time when the individual lifetimes may not be exponentially distributed.

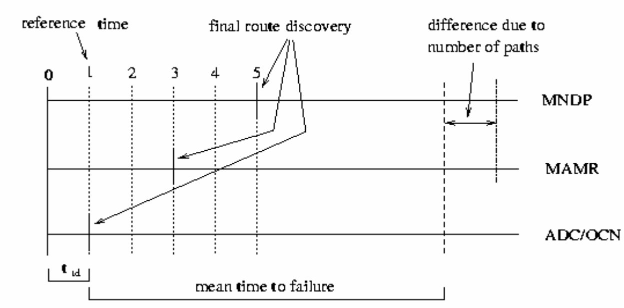


Figure 19. Route failure time of different protocols.

6.4.1. Exponentially Distributed

Let ξ_1, ξ_2, ..., ξ_1 be the rate of failures of the paths identified one by one in each route discovery [8]. The mean lifetime of path *i* added in route discovery *i* is

$$\frac{1}{\xi_i} - (k-i)t_{rd} \quad (1)$$

where $i = 1, k$. Alternatively, the rate of failure of path *i* is given by

$$\xi_i' = \frac{1}{\frac{1}{\xi_i} - (k-i)t_{rd}} \quad (2)$$

If $\xi_i = \xi$, $\forall i = 1, k$, then

$$\xi' = \frac{1}{\frac{1}{\xi} - (k-i)t_{rd}} \quad (3)$$

Since $(k-i)t_{rd}$ is positive, it implies that $\xi' > \xi$. Recall that when all the paths are discovered simultaneously, the expected value of route failure time of a system of *k* node-disjoint paths is given by [3]

$$E[\triangle] = \frac{1}{\xi} \ln\ k \quad (4)$$

Let $\triangle'$ denote the time after which all paths may fail in case of the protocol that discovers paths incrementally. Then,

$$E[\triangle'] = \frac{1}{\xi'} \ln\ k' \quad (5)$$

From 2 we have $\xi' > \xi$. Then for $k \approx k'$, it implies that $E[\triangle'] < E[\triangle]$. In other words, route failure time of an incremental protocol is less than that of a protocol that discovers routes in a single route discovery. We have tried to convey this fact in Figure 19. However, the same will be exemplified in the next sub-subsection.

In what follows, we analyze route failure time when individual lifetimes may or may not be exponentially distributed.

6.4.2. Generalized Analysis

In this sub-subsection, we analyze the route failure time of the over all system of node-disjoint paths from a given source to destination without imposing a restriction of their distribution.

Table 5. Lifetimes of individual paths and time of use using MNDP.

Path	Lifetime	Time of Use
1	T_1	$T_1 - (k-1)\delta$
2	T_2	$T_2 - (k-2)\delta$
3	T_3	$T_3 - (k-3)\delta$
4	T_4	$T_4 - (k-4)\delta$
5	T_5	$T_5 - (k-5)\delta$
…	…	…
i	T_i	$T_i - (k-i)\delta$

Let there be k node-disjoint paths from a source to a destination with lifetimes $T_1, T_2, ..., T_k$. Using a protocol such as ADC/OCN, assume that one is able to discover all of them in a single route discovery. After the route discovery has finished, these paths are available for use at the source for sending packets along them. In general, the average route failure time is given by

$$\overline{\Delta}_{ADC/OCN} = \frac{\sum_{i=1}^{k} T_i}{k} \tag{6}$$

In the following, we prove a theorem that relates lifetime of MNDP with that of ADC/OCN.

Theorem 2: Let there be k node-disjoint paths from a given source to a destination and δ be the average route discovery time of each route discovery in MNDP. Then, the average route failure of the overall system of node-disjoint paths in MNDP is approximately given by

$$\overline{\Delta}_{MNDP} = \overline{\Delta}_{ADC/OCN} - \frac{k-1}{2}\delta$$

Proof: in case of MNDP, the paths are discovered incrementally, i.e., exactly one path is added to the set of node-disjoint paths after each route discovery. The set of paths discovered in any intermediate route discovery cannot be used by the source because it has to be reorganized with that of the ensuing route discovery.

Table 6. Lifetimes of individual paths and time of use using MAMR.

Path	Lifetime	Time of Use
1	T_1	$T_1 - (k-k_1)\delta$
2	T_2	$T_2 - (k-k_1)\delta$
3	T_3	$T_3 - (k-k_1)\delta$
i, i=1, k_1	T_i	$T_i - (k-i)\delta$
4	T_4	$T_4 - \{k-(k_1+1)\}\delta$
5	T_5	$T_5 - \{k-(k_1+2)\}\delta$
i, i= k_1 +1,k	T_i	$T_i - (k-i)\delta$

As a result, even if a discovered path is not going to be changed, it will be idle till the final route discovery is over. Given that δ is the average time incurred in each route discovery. Table 5 shows lifetimes of individual paths and the time of their use.

Actually, the entries in the third column of Table 5 are nothing but $T_i - (k-i)\delta$, where i=1, k. As a result, in general, the route failure time of an incremental protocol (such as MNDP) is given by

$$\overline{\Delta}_{MNDP} = \frac{\sum_{i=1}^{k}\{T_i - (k-i)\delta\}}{k} \tag{7}$$

$$\overline{\Delta}_{MNDP} = \frac{\sum_{i=1}^{k} T_i - \frac{k(k-1)}{2}\delta}{k} \tag{8}$$

$$\overline{\Delta}_{MNDP} = \overline{\Delta}_{ADC/OCN} - \frac{k-1}{2}\delta \tag{9}$$

which proves Theorem 2. We now state another theorem that relates the route failure time of MAMR with that of ADC/OCN.

Theorem 3: let k_1 node-disjoint paths out of k node-disjoint paths be discovered using OFC in MAMR. The rest of the node-disjoint paths be identified incrementally. Then, the average route failure of the overall system of node-disjoint paths in MAMR is approximately given by

$$\overline{\Delta}_{MAMR} = \overline{\Delta}_{ADC/OCN} - \frac{(k-k_1)\delta}{k}\left\{k_1 + \frac{(k-k_1-1)}{2}\right\}$$

Proof: in case of MAMR, let us assume that in the first route discovery, we are able to discover 3 paths using OFC. The remaining paths are discovered incrementally as in MNDP. Table 6 summarizes individual lifetimes of paths and their respective times of use.

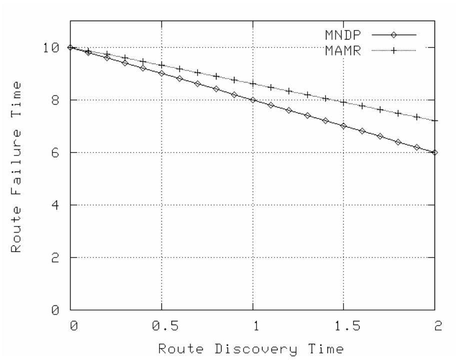


Figure 20. Empirical values of average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for MNDP and MAMR, given that k=5, and $k_1 = 3$.

The entries in the third column of Table 6 contain $T_i - (k-k_1)\delta, i = 1, k_1$, and for rest of the paths the entries resemble $T_i - (k-i)\delta, i = k_1+1, k$. Combining them gives

$$\overline{\Delta}_{MAMR} = \frac{\sum_{i=1}^{k} T_i - \sum_{i=1}^{k_1}(k-k_1)\delta - \sum_{i=k_1+1}^{k}(k-i)\delta}{k} \tag{10}$$

$$\overline{\Delta}_{MAMR} = \frac{\sum_{i=1}^{k} T_i - k_1(k-k_1)\delta - \frac{(k-k_1-1)(k-k_1)}{2}\delta}{k} \tag{11}$$

$$\overline{\Delta}_{MAMR} = \frac{\sum_{i=1}^{k} T_i}{k} - \frac{(k-k_1)\delta}{k}\left\{k_1 + \frac{(k-k_1-1)}{2}\right\} \tag{12}$$

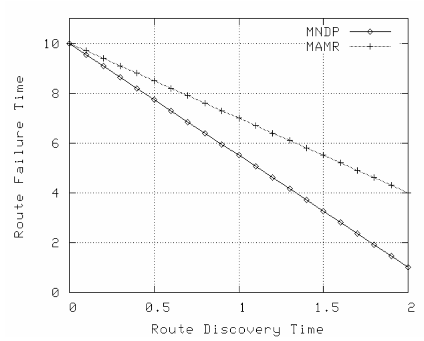


Figure 21. Average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for MNDP and MAMR, given that k=10, and $k_1 = 6$.

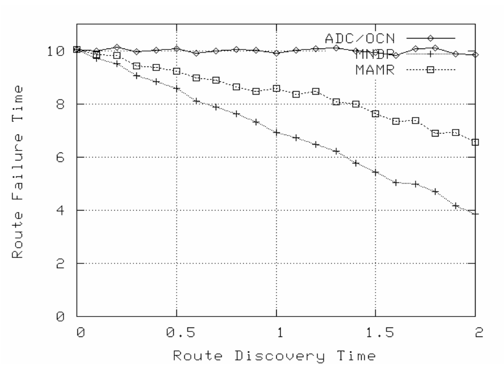


Figure 22. Simulated values of average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for MNDP and MAMR, given that k=5.

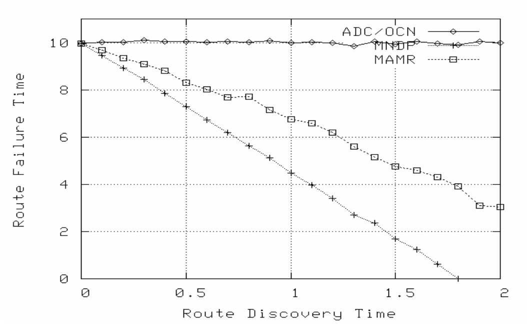


Figure 23. Simulated values of average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for MNDP and MAMR, given that k=10.

$$\overline{\Delta}_{MAMR} = \overline{\Delta}_{ADC/OCN} - \frac{(k-k_1)\delta}{k}\left\{k_1 + \frac{(k-k_1-1)}{2}\right\} \quad (13)$$

which proves Theorem 3. To verify these theorems, let us consider the following example.

Example 3: assume that there are 5 node-disjoint paths from a given source to a destination with the following lifetimes

Path	T_1	T_2	T_3	T_4	T_5
Lifetime	10	11	9	12	8

Suppose that in case of ADC/OCN, all these paths are identified in a single route discovery and all of them are available for use. Using (6), route failure time of the overall system is given by

$$\overline{\Delta}_{ADC/OCN} = \frac{10+11+9+12+8}{5} = 10. \quad (14)$$

In case of MNDP, assume that the value of δ is 0.1 time units, then using equation 9 the value of $\overline{\Delta}_{MNDP}$ comes out to be 9.8 which is less than that of ADC/OCN. This can also be verified as the respective times of use of these paths are $10-4\delta$, $11-3\delta$, $9-2\delta$, $12-\delta$, and 8. Adding them up and dividing the sum by 5 gives the average value of route failure time to be 9.8.

For MAMR, using equation 13 for k=5 and k_1=3, $\overline{\Delta}_{MAMR}$ comes out to be 9.86, which is larger than that of MNDP and is smaller than that of ADC/OCN. This can also be verified as follows. Out of 5 paths, 3 paths are discovered in the first route discovery and the rest of the paths are discovered incrementally in the two successive route discoveries one in each. The three paths that are identified in the first route discovery cannot be used until all route discoveries are finished. As a result, the times of use of three paths that are identified in the first route discovery are $10-2\delta$, $11-2\delta$, $9-2\delta$ and the times of use of those identified incrementally are $12-\delta$, 8. Adding them up and dividing the sum by 5 gives the average route failure time in case of MAMR to be 9.86. In what follows, we present results and discussion.

7. Results and Discussions

We first discuss some empirical results and then we shall discuss results obtained through simulations.

For empirical results, assume that route failure times of individual paths be {9, 10, 11, 12, 8} for 5 paths and {9, 11, 10, 8, 12, 10, 11, 9, 12, 8} giving their average value to be 10 in both the cases. This average value gives the route failure time of overall system in case of ADC/OCN. In other words, route failure time in case of ADC/OCN is $\overline{\Delta}_{ADC/OCN} = 10$.

Figure 20 shows empirical values of average route failure time, $\overline{\Delta}$ as a function of route discovery time, δ, for MNDP and MAMR, given that k=5, and $k_1 = 3$ using equations 6 and 13 discussed above. Figure 21 shows empirical values of the average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for MNDP and MAMR, given that k=10, and $k_1 = 6$. Note that empirical value of the average route failure time for ADC/OCN is 10.0 and is not shown in the graphs. In both these cases, we observe that the relationship between these two parameters is linear and the route failure time of MNDP and MAMR decreases with the increase in route discovery time. We defer the reason of the observed behaviour till simulation results.

To validate the above analytical relationships, we carried out simulations in C++. The values of individual lifetimes of paths were generated randomly using a uniform random number generator. The values of individual lifetimes of paths are randomly though uniformly distributed between 8.0 – 12.0 time units, and the average value of the route discovery time in all cases is assumed to be 0.1 time units. This seems a bit realistic in the sense that lifetime of a path is on an average 10 seconds and the average value of route discovery time is 100 milliseconds. In case of ADC/OCN, the value of route failure time is the average of the uniformly random values generated in the range [8.0:12.0], i.e., the average is approximately 10.0. In case of MAMR, the value of k_1 was generated randomly such that $1 \le k_1 \le k$. In other words the value of number of paths discovered in the first route discovery using OFC is uniformly and randomly distributed between 1 and k. Note that each point represents an average of the values obtained in 10000 runs. In what follows, we discuss results obtained through simulations.

Figure 22 shows values of average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for ADC/OCN, MNDP and MAMR, given that k=5. Figure 23 shows Simulated values of average route failure time, $\overline{\Delta}$, as a function of route discovery time, δ, for ADC/OCN, MNDP and MAMR, given that k=10.

In accordance to empirical results, we observe that the relationship between these two parameters is almost linear and the route failure time of MNDP and MAMR decreases with the increase in route discovery time. However, as pointed out for empirical results, the rate of decrease in case of MNDP is larger than that of MAMR. The reason is that, in case of MNDP, routes are discovered incrementally, one in each route discovery, and the discovered routes are to be reorganized with the routes discovered in the current route discovery, therefore, the routes cannot be used until all routes are discovered. However, in case of MAMR, a number of routes are discovered in the first route discovery using OFC, and the remaining routes are discovered as in MNDP. As the number of route discoveries is decreased, therefore, the discovered routes can be used quite before as that in case MNDP. This accounts for the observed behaviour of these two protocols.

Further, we observe that the amount of decrease when there are 10 paths is larger as compared to the situation when there are only 5 paths between a given source to the destination. This is in accordance with empirical analysis. Note that the trend or the observed behaviours about the variations of route failure time are more or less similar in both the cases i.e., empirical as well as that in simulations. However, the decrease in case of simulation results is a bit more than that in case of empirical values. The reason is that in case of empirical results the number of paths discovered in the first route discovery is fixed. Specifically, k_1=3 for k =5, and $k_1 = 6$ for k=10. As opposed to it, in simulations the value of k_1 may vary from 1 to k. In each run, a different value of k_1 might have been generated and the net effect after an average over the number of runs is depicted in simulations results. This accounts for the observed behaviour.

We would like to emphasize that as we go on increasing route discovery time, the route failure time decreases. At one point, it will become 0. Let $\overline{\Delta}_{ADC/OCN} = \overline{\Delta}$. In case of MNDP, the value of route discovery time for which the average route failure time becomes 0 is given by

$$\delta_{MNDP} = \frac{2\overline{\Delta}}{k-1} \tag{15}$$

In case of MAMR, this value of route discovery time is given by

$$\delta_{MAMR} = \frac{2k.\overline{\Delta}}{(k-k_1)(k+k_1-1)} \tag{16}$$

On the basis of the theorems and the example discussed above, we can say that route failure time of an incremental protocol (such as MNDP or MAMR) is less than that of a protocol (such as ADC or OCN), which possibly discovers all routes in a single route discovery [17]. The route failure time in case of MAMR is larger than that of MNDP because the number of route discoveries required by MAMR is less than that required by MNDP. As a result, in case of MAMR the source can start using discovered node-disjoint paths earlier than that in MNDP. The effective route failure time of MNDP is smaller than that of MAMR, which is in turn smaller than that of ADC/OCN.

From the above discussion we can say that the protocol that discovers node-disjoint paths incrementally is suitable for low mobility scenarios where paths do not fail frequently. At high node mobilities, the discovered paths can fail frequently. By the time a new path is discovered, some paths already discovered might have failed. Therefore, the paths discovered incrementally will not be of much use for data transfer. Therefore, we conclude that an incremental protocol is suitable for applications where the mobility is low or the delay requirement is not stringent.

8. Conclusion

In this paper, we discussed that it is possible to combine approaches that try to discover multiple node-disjoint paths in a single route discovery and those that try to discover them incrementally for identifying a maximal set of node-disjoint paths between a given source and a destination and still preserving the guarantee in the hybrid approach that are provided by an incremental protocol. The contributions of the paper are as follows.

- We presented a version of MNDP [18] so that it does not explicitly require a single path routing protocol (such as DSR) in the first route discovery to identify an initial reference path.
- We proposed a protocol that adopts an approach that is hybrid of the approaches taken by a protocol that tries to discover node-disjoint paths one at a time in an incremental fashion and that of a protocol that tries to identify them in a single route discovery. We call the hybrid protocol MAMR. This can reduce the number of subsequent route discoveries if the number of paths identified in the first route discovery is more than one.
- We compared MAMR with MNDP and with the schemes such as ADC/OCN discussed in [6]. We argued that without incurring a significant amount of additional communication overhead, the incremental protocols (MNDP and MAMR) are guaranteed to discover all node-disjoint paths while no such guarantee is provided by ADC/OCN which try to discover paths in a single route discovery.
- The cost paid is in terms of route discovery time. The route discovery time of MNDP is k times of the time taken by one route discovery, where k is the number of node-disjoint paths. The number of route discoveries is reduced in case of MAMR as compared to MNDP which in turn reduces communication overheads.
- We analyzed the route failure time of ADC/OCN, MNDP and MAMR when the lifetimes of individual paths are: (i) exponentially distributed, and (ii) uniformly and randomly distributed within a given range.
- We carried out simulations and the results obtained through simulations are in accordance with those obtained empirically.
- We observed that the route failure time in case of MAMR is larger than that of MNDP and is less than that of ADC/OCN. This suggests that MNDP and MAMR may be used in low mobility scenarios where routes do not fail frequently.

The design of a protocol that relies on partial information about the topology and discovers all node-disjoint paths in a *single route discovery* is an open problem and that forms the future work.

References

[1] Abbas M. and Abbasi A., "An Improvement over Incremental Approach for Guaranteed Identification of Node-Disjoint Paths in Mobile Ad hoc Networks," *in Proceedings of 2nd IEEE/ACM International Conference on Communication Software and Middleware (COMSWARE)*, pp. 1-10, 2007.

[2] Abbas M. and Istyak S., "Multiple Attempt Node-Disjoint Multpath Routing for Mobile Ad hoc Networks," *in Proceedings of 3rd IEEE/IFIP International Conference on Wireless and Optical Communication Networks (WOCN)*, Bangalore, pp. 1-5, 2006.

[3] Abbas M. and Jain N., "An Analytical Framework for Route Failure Time of Multiple Node-Disjoint Paths in Mobile Ad hoc Networks," *in Proceedings of 12th IEEE/VDE International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, New Delhi, pp. 1-6, 2006.

[4] Abbas M. and Jain N., "Mitigating Path Diminution in Node-Disjoint Multipath Routing for Mobile Ad hoc Networks," *International Journal of Ad hoc and Ubiquitous Computing (IJAHUC)*, vol. 1, no. 3, pp. 137-146, 2006.

[5] Abbas M. and Jain N., "Path Diminution in Disjoint Multipath Routing for Mobile Ad hoc Networks," *in Proceedings of 15th IEEE Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Barcelona, pp. 130-134, 2004.

[6] Abbas M. and Jain N., "Path Diminution is Unavoidable in Node-Disjoint Multipath Routing in a Single Route Discovery," *in Proceedings of 1st IEEE International Conference on Communication Software and Middleware (COMSWARE)*, New Delhi, pp. 1-6, 2006.

[7] Abbas M., "Disjoint Multipath Routing in Mobile Ad hoc Networks," *PhD Thesis*, New Delhi, 2005.

[8] Abbas M., Khandpur P., and Jain N., "A Disjoint Alternate Path Routing for Mobile Ad hoc Networks," *Computer Journal of Internet Technology*, vol. 6, no. 1, pp. 111-119, 2005.

[9] Abbas M., Khandpur P., and Jain N., "NDMA: A Node Disjoint Multipath Ad hoc Routing Protocol," *in Proceedings of 5th World Wireless Congress (WWC)*, San Francisco, pp. 334-339, 2004.

[10] Ahn C., Shin J., and Huh N., "Enhanced Multipath Routing Protocol Using Congestion Metric in Wireless Ad hoc Networks," *in Proceedings of IFIP International Conference on Embedded And Ubiquitous Computing (EUC)*, pp. 1089-1097, 2006.

[11] Ahuja K., Magnanti L., and Orlin B., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, New Jersey, 1993.

[12] Alsuwaiyel H., *Algorithms: Design Techniques and Analysis*, World Scientific, 1999.

[13] Arora S., Lee H., and Thurimella R., "Algorithms for Finding Disjoint Paths in

Mobile Networks," http://www.cs.du.edu/ramki/papers/distributed Disjoint. pdf, 2003.

[14] Bertsekas D. and Gallager R., *Data Networks*, Prentice Hall, 1992.

[15] Bettstetter C., "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," *in Proceedings of 3rd ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc)*, pp. 80-91, 2002.

[16] Cormen H., Leiserson E., and Rivest L., *Introduction to Algorithms*, Prentice Hall of India, New Delhi, 2001.

[17] Johnson B. and Maltz A., *Dynamic Source Routing in Ad hoc Wireless Networks,* Academic Publishers, 1996.

[18] Liu C., Conner S., Yarvis D., and Guo X., "Guaranteed On-Demand Discovery of Node-Disjoint Paths in Ad hoc Networks," *Intel Technical Report IR-TR-2004-261*, 2004.

[19] Liu C., Conner S., Yarvis D., and Guo X., "Guaranteed On-Demand Discovery of Node-Disjoint Paths in Ad hoc Networks," *Elsevier Journal on Computer Communications*, vol. 30, no. 14, pp. 2917-2930, 2007.

[20] Mueller S., Tsang P., and Ghosal D., "Multipath Routing in Mobile Ad hoc Networks: Issues and Challenges," *Computer Journal of Lecture Notes in Computer Science*, vol. 2965, no. 2, pp. 209-234, 2004.

[21] Nasipuri A., Castaneda R., and Das R., "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad hoc Networks," *Computer Journal of Mobile Networks and Applications (MONET)*, vol. 6, no. 4, pp. 339-349, 2001.

[22] Perkins E. and Royer B., "Ad hoc On-Demand Distance Vector Routing," *in Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, pp. 90-100, 1999.

[23] Pham P. and Perreau S., "Performance Analysis of Reactive Shortest Path and Multipath Routing Mechanism with Load Balance," *in Proceedings of IEEE Conference on Computer and Communication (INFOCOM)*, USA, pp. 251-259, 2003.

[24] Ramasubramanian S., Harkara M., and Krunz M., "Linear Time Distributed Construction of Colored Trees for Disjoint Multipath Routing," *Elsevier Journal on Computer Communication*, vol. 51, no. 10, pp. 2854-2866, 2007.

[25] Sambasivam P., Murthy A., and Royer B., "Dynamically Adaptive Multipath Routing Based on AODV," *in Proceedings of Mediterranean Ad Hoc Networking Workshop (MedHocNet)*, Turkey, pp. 561-568, 2004.

[26] Sue C. and Chiou J., "A Hybrid Multipath Routing in Mobile ad hoc Networks," *in Proccedings of 12th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 399-400, 2006.

[27] Waharte S. and Boutaba R., "Totally Disjoint Multipath Routing in Multihop Wireless Networks," *in Proceedings of IEEE International Conference on Communications (ICC)*, pp. 5576-5581, 2006.

[28] Ye Z., Krishnamurthy V., and Tripathi K., "A Framework for Reliable Routing in Mobile Ad hoc Networks," *in Proceedings of IEEE Conference on Computer and Communication (INFOCOM)*, pp. 270-280, 2003.

Ash Abbas revived his Bachelor of science in computer engineering in 1994, and Master of science in communication and information systems in 1996, both from Aligarh Muslim University, India. He obtained his PhD in computer science and engineering from Indian Institute of Technology Delhi, India in 2006.