A Fault-tolerant Scheme for Routing Path Re-establishment for reliable communication in Heterogeneous Networks

R.S. Shaji, Dr. R.S. Rajesh, B Ramakrishnan

Abstract – In heterogeneous environments, devices accessible in different networks may help to provide new opportunities for utilizing new services when they are connected efficiently. In mobile ad hoc networks, it is very difficult in maintaining the links among the devices, because of the frequent change in the density of mobile devices, their various medium of access nature and their mobility. In this paper, we have considered various aspects of the occurrence of fault in the routing path by predicting in earlier. We have tested the path maintenance procedure using three fault occurring scenarios. Ours is a novel routing scheme named SFUSP (Self-eliminating Fault-tolerant based Un-interrupted reliable Service switching mobile Protocol) which is basically a proactive scheme with added functionalities like clustering and self-elimination. It is specially designed for establishing reliable route in the heterogeneous networks and senses the path link break in advance. Performance evaluation is done for routing metrics and fault-tolerant metrics compared with other protocols like AODV, DSR and OLSR under various mobility models like Random Waypoint, Brownian and Manhattan for different mac layers.

Index Terms – Fault-tolerant, Heterogeneous, Self-elimination and Service Discovery.

1 INTRODUCTION

A Mobile Ad-hoc Network (Manet) is composed of mobile nodes without any infrastructure. In this environment, multicast routing protocols are faced with the challenge of producing multi-hop routing under host mobility and medium of access constraints [01].

Clustering of devices in Manet could reduce overhead, flooding and collision in communication and make the network topology more stable. Location aided clustering technique for mobile ad hoc network routing helps in improving the routing performances of the Manet protocols [02][11]. But, Location based routing is difficult when there are gaps in the network topology due to the mobility of nodes [07][10]. Also, due to mobility, it is necessary to control the broadcasting mechanism to avoid flooding of control messages to find out the nearby servicing node [08].

The challenges in wireless, mobile inter-domain routing include dynamic network topology, medium of access on devices, intermittent connectivity, routing protocol heterogeneity [03]. The performance of routing protocols in Manet depends on the availability and stability of wireless links [04]. Topology control in Manet is needed for reducing interference collisions and in consequently retransmission [05][06].

The mobility model is one of the main aspects for testing the performance of MANETs routing protocols [09][10]. Also, the medium of access of different devices available in the network causes delay in establishing delay among heterogeneous devices in the network [12]. Thus, it is very complex to maintain the service provisioning link established by the routing protocols under heterogeneous networks due to the varying mobility and mac layers of the devices available in Manets. Link Break in routing path leads to the setback in the communication process among the devices and it is necessary to design an efficient fault-tolerant routing scheme that predict the occurrence of fault in prior for maintaining a new routing path without any delay. Proactive recovery is an essential method for ensuring the long-term reliability of fault-tolerant systems in Manets [13][14].

Here, we have discussed a proactive based fault-tolerant routing scheme specially made for establishing reliable path and maintain the path among the mobile devices communication in heterogeneous environment. Our scheme is a proactive based routing one which maintains updated information about the status of devices in the network. A mac level communication and user interface identification is done during the time of broadcasting the message. In addition to the mentioned characteristics, a Location based clustering and comparison based self-elimination methods are used in the selection of reliable and correct service provisioning nodes in the service path. Prediction on location and mobility speed of the devices in the network for a time instant helps in finding the occurrence of fault in prior and three scenarios are set and discussed for overcoming the issues. The performance metrics are compared to AODV, DSR and OLSR [17][18] protocols under Random Waypoint, Brownian and Manhattan mobility models for Wired and Wireless LANs [15][16][21].

2 Related Works

Proactive recovery is an essential method for ensuring the

long-term reliability of fault-tolerant systems [13]. A fault tolerant service discovery protocol for Manet using quorum of directories by selecting their weight values is discussed in [14]. A routing protocol, referred to as Adaptive routing protocol for fast Recovery from large-scale Failure, is discussed in [23] to recover a network quickly after failures over large areas. ARF detects failures by counting the packet losses from parent nodes, and upon failure detection, it decreases the routing interval to notify the neighbor nodes of the failure.

But none of these works describe the fault which occurs at the time of communication after the path is established. There are unexpected occurrences of fault like communicating nodes move away from the routing path, better strength nodes enter in the path and removal of un-necessary nodes (not needed for communication at that time instant) in the communication path. Our routing scheme identifies the fault with respect to location and time, re-establishes the path for devices with heterogeneous mac under varying mobility condition [09].

3 Reliable Routing Scheme

3.1 Self-eliminating Fault-tolerant based Un-interrupted reliable Service switching mobile Protocol (SFUSP)

Self-eliminating Fault-tolerant based Uninterrupted reliable Service switching mobile Protocol (SFUSP), is a protocol specially designed for heterogenous computing environment for Manets. It is basically a proactive protocol [20] with additional functionalities added such as clustering and self-elimination. It is a well supported context-aware and fault-tolerant service discovery routing protocol. SFUSP is reliable while searching the exact service offering node and thus, it reduces the searching time and balance load among the nodes which are all involved in the process of discovery in heterogeneous networks.

SFUSP follows a new technique to eliminate less strengthened nodes (ie. Low Resource, Low Power, Low Bandwidth and Unstable) during the path discovery process. The additional task of refreshing the existing list of nodes will be reduced, if the unwanted nodes are got eliminated in the time of path discovery itself. SFUSP will keep only the best selected node details in its database.

SFUSP is as intelligent as a reliable manet routing scheme for any heterogeneous environment which can along with any service discovery protocol, can help in providing good service provisioning. Location based systems can be used along with this scheme for the unfavorable situations to find the previous servicing location details and a new link can be established, if the service gets interrupted due to non-availability of nodes nearby the link.

The routing scheme contains three main functionalities such as clustering, self-elimination and routing path establishment and is shown in Fig. 1. The steps involved in SFUSP routing are as follows,

Step01: Broadcast the message to discover the node which is requested by the requester node.
Step02: If Unfound then representative nodes will be activated to generate a new search in their nearby locations.
Step03: If requester node is found then
Step04: Group all the representatives by location.
Step05: Find the most reliable representative.
Step06: Do the self-elimination technique to reduce to best suited nodes in order to find the most reliable path.
Step07: Establish the reliable path with the best strengthened nodes in the path.

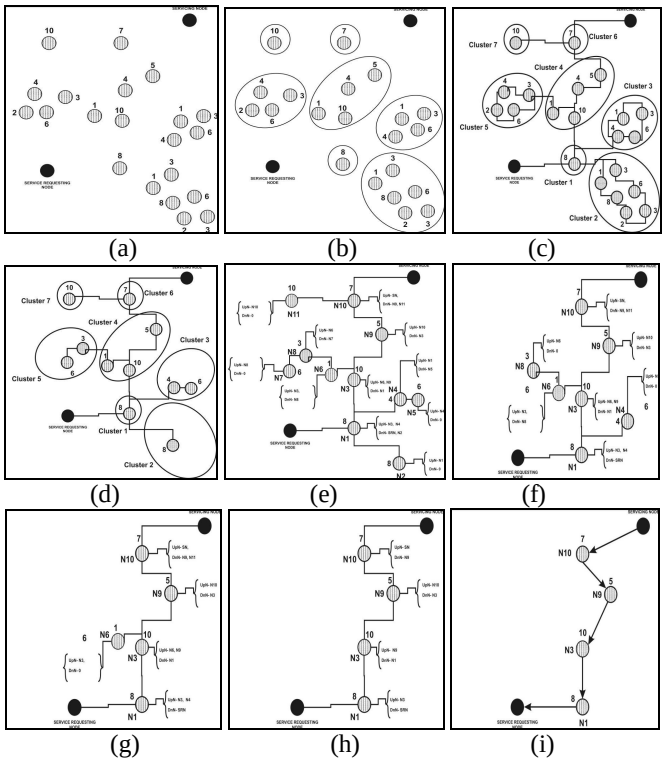

Fig. 1 (a) SFUSP node Discovery (initial stage), (b) SFUSP nodes Clustering. (c) SFUSP Cluster Linking, (d) SFUSP Elimination Session starts, (e,f,g,h) SFUSP Elimination Sessions, (i) SFUSP finds a Reliable Service Path.

The Nodes with black shade represent the service requester and service provider for a time instant. Other nodes may act as representative nodes in the routing path. Next session shows how the nodes in the network are grouped under any clusters available in the network and the algorithm is given as Fig. 2.

Clustering

After the eligible nodes are selected, they are grouped by locations by using a clustering method, where no node will be made excluded from the available clusters in the network. Grouping of nodes in the cluster under an area is done through a location based

system. Increase in the number of clusters represents the increase of the network area.

Algorithm : Clustering

```
Algorithm CLUSTERING ()
Ci← 1(Initialize);
Node[i]←list of nodes in the area
Area[i] ← Area ID //Get the area list
For each Area[i] do
        For each node[i] do
                NodeClusterID ← Ci
        NodeAreaID ← Area i
        End For
Ci←Ci+1
 //when area increases the no. of cluster also get increases
 End For
End
```

Fig. 2 Algorithm for Clustering

Clustering enables the comparison of better strengthened node in that area. In manets, nodes often change their native cluster due to the mobility and so get reduced in the efficiency of communication among source and destination. The comparison among the nodes may be done among themselves by using a technique called self-elimination and is given in Fig. 3.

Time taken for processing individual node and all clusters in the network are calculated by their data length, transmission rate and their delay values. Time taken for single node simulation for any nodes in the network can be calculated as follows,

$$\left(\frac{Data\,Length}{Transmission\,Rate}\right) X\ Transmission\,Delay \quad (1)$$

Cluster simulation time can be calculated as the total number of clusters available in the network with the data length to the transmission rate of each node in the cluster multiplied with the transmission rate. It is calculated as follows,

$$\left(\frac{Total\,Number\,of\ Nodes\ \text{in a cluster}\ X\ Data\,Length}{\sum_{i=0}^{n} Transmission\,Rate_i}\right) X\ Transmission\,Delay \quad (2)$$

Our clustering is a location based clustering scheme, where cluster sizes are constant one. Cluster size may not be increased with the increase in network size; instead number of clusters may be increased. A cluster may contain any number of nodes in its account.

Impact of Mobility on clusters includes movement of nodes from and to the cluster, change in the transmission range, data rate and medium of access of individual nodes in the cluster. This leads to problems like non-availability of routing path, link break and understandability among devices in the cluster. These are dealt with our simulation models by altering the packet structure of the protocol by including the information about each device which is gathered at the time of broadcasting.

Self-elimination Process

The highlight of the SFUSP routing scheme is the self-elimination procedure that helps in finding the most strengthened node in the cluster. Along with the proactive nature of the scheme, the self-elimination process finds effective node in the cluster for establishing reliable links on the basis of the strength of the nodes. The strength may be calculated with the context specified by the requester and the packet structure is given in the next section. The following algorithm in Fig. 3 illustrate the processes involved in the self-elimination technique.

Algorithm : Self-elimination

```
Algorithm SELF_ELIMINATION()
cluster_nodes[i] ← all node ids in the cluster
me_node_id ← service providing node id
Threshold  ← n
DownNode(x) ← Service request node ids
Up_Node← original first service request node id
for each cluster_nodes do
if (cluster_nodes[x].DownNode(x) ≠ "empty")
{
        If(cluster_nodes[x].strength ≥  threshold)
        {
                cluster_nodes[x]← "alive"
                //context service found
        If (cluster_nodes[x].context =
        me_node_id.context) then
        // check the context
        {
        If (service request node ≠     null)
        {
                Send message to service request node
        (Up Node) "chosen"
        }
        End if
        If (Up_Node ≠ null) then
         //to check the node reached the original first
                        location
        {
        Send message to Up_node "Path  Fixed"
        }
End if
}
Else
//if context not matched and threshold value is not satisfied
{
        cluster_nodes[x] ← Representative node
}
End if
```

```
}
else
//if the node has no strength means it can be in a
  not alive stage
{
        cluster_nodes[x] ← "not alive"
}
End if
}
else
{
        cluster_nodes[x]← "not alive"
}
End if
//if the path is fixed and the node is not selected means
        it can be not alive by it-self
If (cluster_nodes[x].message = "path fixed") then
{
        If (cluster_nodes[x] ≠  "chosen") then
        {
                cluster_nodes[x] ← "not alive"
        }
        End if
}
End if
End if
End For
End
```

Fig. 3 Algorithm for Self-elimination.

In the self-elimination process, each node will collect the information about their service providing and service requesting nodes ie. Up and down nodes. Then, the nodes which have not participated in the routing process in the list will be deleted first and this will be repeated until a reliable path is established. In Fig. 3(a), the requested nodes are discovered during the path establishment process. The two black shaded nodes in the figure represent the source and destination nodes. The intermediate nodes are the nearby nodes or nodes in the coverage area. In Fig. 3(b), all the nodes are clustered on the location basis. Each node is represented by a value which indicates the strength of the node on the basis of bandwidth, connectivity etc.. All the location wise identified clusters are linked together and so, a location may contain n-clusters. From which the best suited node in the cluster will be selected by the self-elimination process.

In the first session of elimination, all the grouped clusters, unwanted or low strength nodes will be eliminated. In each group, a comparison will be done to find the most strengthened node to establish a reliable path. Fig. 3 (c) and (d) illustrates the cluster linking and the start of elimination session. In the elimination session, in Fig. 3 (e,f,g,h) all the weak and unreliable nodes are get eliminated from the reliable link. By this, we can get an exact and reliable path to continue with the communication.

This is an intelligent elimination technique, where most reliable path can be set by removing the less reliable nodes which still get connected after the session of elimination. Here each node has to generate a list of Upper Node (UpN) and Down Node (DnN) detail from their connection establishment. After generating this list, the nodes which have no Down Node (DnN) can be voluntarily got off from the connection establishment. Both the elimination and clustering processes are done without the knowledge of the service requesting node.

Finally, the most reliable path for routing in the network will be established as in the Fig. 3 (i) . The communication will be done through this path. The path will get often changed in the mobile environment and so the process is repeated at once with any node that senses the nearby node that is left or a new node that gets entered in the cluster or un-necessary nodes in the link.

Simulation time of non-eliminated nodes in a cluster can be calculated as the total number of current nodes present in the cluster with the sum of the node data length to the sum of the transmission rate of all nodes multiplied with transmission delay.

The non-eliminated nodes in the cluster can be calculated as follows,

$$\left(\frac{Total\ Number\ of\ Current\ Nodes\ presnet\ in\ a\ Cluster\ X \sum_{i=0}^{n} Node_i\ DataLength}{\sum_{i=0}^{n} TransmissionRate_i} \right) X\ TranmissionDelay \quad (3)$$

Now, we can calculate the self-elimination time of a node in the cluster by the following formula,

$$ClusterSimulationTime - Non - E\ \text{li}\min ated\ NodesSimulationTimeof\ aCluster \quad (4)$$

The routing process taking place during the path discovery is illustrated in the following algorithm in Fig. 4.

Routing

The routing process comprises of clustering process then followed by self-elimination technique in each cluster.

Algorithm : Routing

```
Algorithm ROUTING_PROTOCOL_SFUSP(list
                                    _of_replied_nodes)
Location_Li ← All Geographic Location List of the replied
                                                  nodes
call CLUSTERING();
If (service='established') then
for each Location_Li Do
        Nodei ← All Nodes in the Location Li
        for each Node in Li Do
                If Nodei greater in the group then
                        Keep the Node Alive
                        call SELF_ELIMINATION()
```

else
Delete the node from the Cluster;
End If
End For
End For
End If
End

Fig. 4 Algorithm for Routing.

Our aim is to find a better path establishment time with minimum number of effective nodes in the service path which is established in a less number of time. The path Establishment time can be the sum of the cluster simulation time and is given as

$$\sum_{i=0}^{n} Cluster_i SimulationTime \quad (5)$$

4 FAULT_TOLERANT ROUTING

4.1 Location Base table

This table maintains the most reliable node's Location and Identity in each location and assume each cluster is defined here as location. The Location table will be maintained by the service requesting node. Also, servicing node will periodically maintain the detail of the service requesting node. When any accident occurs to the intermediate nodes, service path will be detached and to overcome this, the SFUSP will maintain this Location base table with periodic update of the details of the nodes for a particular interval of time.

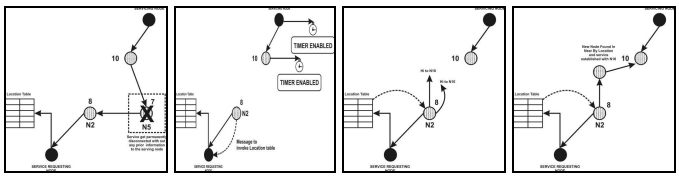

Fig. 5 (a) Service Interruption, (b) Location Base Table Maintenance, (c) Based on location new search invoked, (d) New Location is found and Service path is established.

For example, in the above figure, the node N5 is got detached from the service path, without any information to the nearby nodes which are in the service path. Now N2 will give a message to the service requesting node to fetch the Location base table because, N2 doesn't know about any other nodes other than N5 to establish a service link and so to continue a new search for the nearby locations.

After the service is get interrupted, each node will enable a timer to keep the service provisioning process active and the down link nodes will request the service requesting node, which has the location base table to invoke a nearby node. This is because no other nodes in the service link maintain this table because the location base table is a temporary path table to be maintained.

4.2 Fault Arising Scenarios

Servicing Nodes Move Away from Service Path

Nodes may move away from the routing path even after path establishment is done by the routing protocol due to the mobility of nodes. Fig. 6 shows a situation where a node moves away from the routing path and how it is handled by our routing scheme. Distance and time are the two parameters by which the nodes movement can be predicted and the following algorithm in Fig. 7 describes how the route will be maintained from the cited fault occurring scenario.

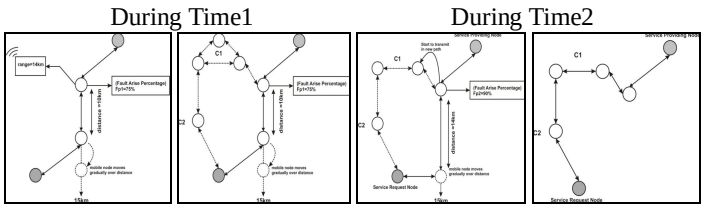

Fig. 6 (a,b,c,d) New Path Establishment when servicing, nodes moves away from service path.

Formula to get the percentage of distance covered over a transmittable range:

d = distance
T1, T2 = time
Fp1, Fp2 = Fault percentage
Range = Transmittable Node Range

During time1:

$$time1\_distance\_provider = \left(\frac{d}{range}\right) X\ 100$$

During Time2:

$$time2\_distance\_provider = \left(\frac{d}{range}\right) X\ 100$$

Algorithm : Servicing Nodes Move Away

Algorithm DISTANCE_INCREASE ()
*time1_distance_coverage ← (current distance of service getting node/radio_range)*100*
threshold1← n1; fixed threshold value for distance coverage
threshold2← n2; fixed threshold value for distance coverage
If (time1_distance_coverage ≥threshold1) then
Start New Service Discovery Nearby to reach the target node;
Continue transmit in current path;
*time2_distance_coverage ← (current distance of service getting node/range)*100*
Else If (time2_distance_coverage ≥threshold2) then
Continue transmit in new path;

```
        End If
Continue transmit in current path;
End If
END
```

Fig. 7 Algorithm for servicing nodes move away from Service Path

Better Strength Node Enters in the Service Path

Our next fault occurring scenario is explained in Fig. 8 where, a new better strength node approaches near the routing path. The algorithm in Fig. 9 show how the new node is identified by clustering technique and how the node is found as good strength node by using self-elimination technique.

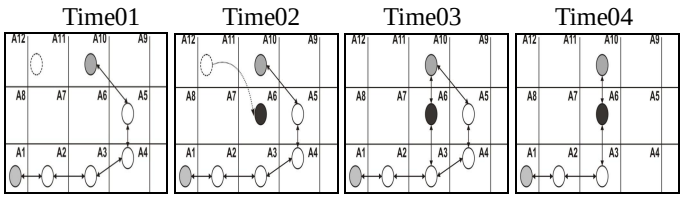

Fig. 8 (a,b,c,d) New Path Establishment when Better Strength Nodes Enters the service path.

Algorithm : Better Strength Node Enters

```
Algorithm NEW_NODE_NEAR_BY()
for each n-seconds do
        Do New Service Discovery Near By Location;
        Continue transmit in current path;
        If New service found then
                compare me.service_request_node.Area ID
                and NewNode.AreaID;
                If NewNode.AreaID is nearby then
                        Establish SERVICE with  NewNode;
                If NewNodeID.ESTABLISH=TRUE then
                Continue the Transmission in the new
                path established;
                End If
        End If
End If
Loop
END
```

Fig. 9 Algorithm for Better Strength Nodes Enters in Service Path

Un-necessary Nodes in Service Path

Due to mobility of the nodes in the network, the communication path may contain new nodes with better strength got connected in the routing path. But some nodes may not be disconnected form the routing path due to the lack of predictability in the distance and time. These nodes are un-necessarily available in the network. These node must be eliminated from the routing path and the algorithm in Fig. 10 shows how the path is re-established with that type of connectivity.

Algorithm : Unnecessary Node in Service Path

```
Algorithm UNNECESSARY_SERVICING_NODES()
if distance ≥ threshold then
If service_provider_node.AreaID = service_request_node. AreaID
then
        request the Up_Node_ID of the
        service_request_node
For each n-seconds do
Do New Service Discovery for
        Up_NodeID
                If service found then
                        ESTABLISH the new path;
                        continue transmit in new
                                path;
                End If
        End FOR
End If
        continue service in current path;
        UNNECESSARY_SERVICING_NODES():
                        //Recursive calls
End If
END
```

Fig. 10 Algorithm for Unnecessary Nodes in Service Path

The calculation of distance among the devices and the routing path for the particular time is done by the location based systems. The procedure will be called recursively to find out the nodes that should be removed from the communication path.

5 Performance Analysis

5.1 Packet Structure

The packet structure for broadcasting, service requesting and service providing is given below. The broadcasting packet structure contains the broadcasting node address, packet type, context which contains individual node characteristics with user interface of requester, location status and is represented by

<Service-Requester-Address, Packet_type, Context, Location_id, Mac_id, user_interface_type >.

For service requesting packet contains the service requester address, the packet type, context searched with the user interface, state of the requester, distance status, cluster

status, transfer rate, radio range and delay details and is given as

<Service-requester-address, Packet_type, Context, Location_id, Mac_id, user_interface_type, Transfer_rate, Radio_range, Radio_delay>.

The service provider uses the packet structure which contains service requester as well as service provider address, the packet type, its service description with user interface, state of the node, location, transfer rate, radio range and delay. The packet structure is given as

<Service-provider-Address, Service-requester-address, Packet_type, Context, Location_id, Mac_id, user_interface_type, Transfer_rate, Radio_range, Radio_delay>.

5.2 Simulation Environment

The simulation is done in network simulator ns2.34 [19], which is a discrete event simulator. The above mentioned packet structures are used for broadcasting, requesting and providing the data for communication. Various networks such as IEEE802.11, 802.15 and 802.16 are tested under their corresponding mac layer. For about 10 services as contexts are assigned randomly to all nodes with varying node density of 50, 100, 200 and 400. The pervasive discovery protocol (PDP) [22] implemented in Ns2 is used for service provisioning in the environment. The mobility scenario is set to Random Waypoint, Brownian and Manhattan Model. The testing was compared with other manet reactive protocols like AODV, DSR and proactive protocol OLSR.

5.3 Parameters

The parameter is set to the following values and the testing was done under normal conditions.

Network Area	1500 x 1500 m
Channel Type	Wireless
Propagation Model	Two Way Ground
Radio Range	100 m
Radio Delay	10 ms
Traffic Type	CBR
Duration	200 Seconds
MAC Layer	IEEE 802.3, 802.11, 802.15 & 802.16
Protocol	SFUSP, AODV, DSR, OLSR
Mobility	Random Waypoint, Brownian, Manhattan Model.
Node Strength	Energy, Bandwidth, Context, Node Speed
Context	Contexts such as internet, disk, printer, games etc.. are arbitrarily assigned to all nodes.
No. of Nodes	50, 100, 200, 400
Speed	25 m/s with a pause time of 10 ms. Manhattan min. speed 25 m/s and max. speed 100 m/s.
Transmission rate	9.6 Kbps
Data Payload	512 bytes
Traffic Load	Packet Sent in Every 10 ms

Fig. 11 Ns2.34 Parameter Values

5.4 Metrics

Performance metrics are compared to the reactive protocols like AODV, DSR and proactive protocol OLSR under various mobility models such as Random Waypoint, Brownian and Manhattan for various networks with mac IEEE 802.11, 802.15 and 802.16.

Normalized Routing Overhead

A number of routing packets are transmitted for every data packet sent. Each hop of the routing packet is treated as a packet. Normalized routing load are used as the ratio of routing packets to the data packets [11].

For Random mobility with 802.16 mac, because of the limited transmission capacity of the underlying mac, our scheme captures the best nodes as early as possible and thus reduces the routing overhead.

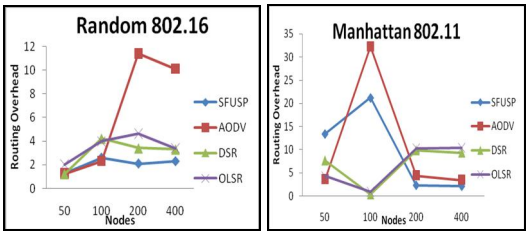

Fig. 12 Normalized Routing Overhead of various protocols under Fixed, Random and Manhattan Mobility Models with mac layers IEEE 802.11 and 802.16

For Manhattan model under 802.11 mac, SFUSP performs well for increasing number of nodes because of the number of clusters it creates and of the proactive nature in maintaining the path.

Path Establishment Time before Fault Occurs

Path establishment time is calculated from the time taken for simulating all clusters available in the network by the self-elimination process.

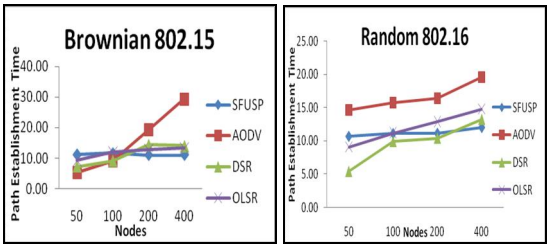

Fig. 13 Path Establishment Time before fault occurs for various protocols under Brownian and Random Mobility Models with mac layers IEEE 802.15 and 802.16

For Brownian motion under mac 802.15, SFUSP shows a constant performance although the number of nodes is increased and is because of the movement of the node and the area of transmission capacity of the underlying mac layer. For Random mobility under 802.16 mac, our scheme performs better because of the transmission limit of the network.

Fault-tolerant Success Percentage

Percentage of the number of link cut occurs to number of recoveries and is calculated as the time between the first connection establishments to the connection break occurs within 10ms.

Servicing Nodes Move Away from Service Path

It is found that, for Brownian 802.15 Sfusp performs good for any number of nodes in network. Olsr is performing constantly below 50%. Dsr and Aodv is not performing as good for larger number of nodes. For manhattan 802.11,

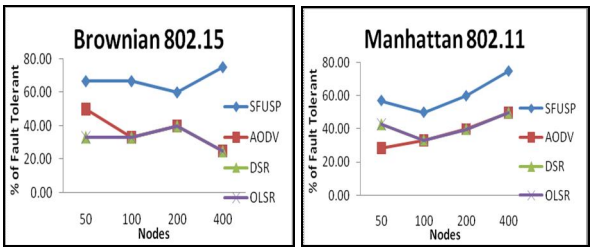

Fig. 14 Fault-tolerant success for various protocols under Brownian and Manhattan Mobility Models with mac layers IEEE 802.11 and 802.15

Sfusp is showing a good performance compared to other protocols. Olsr is not good for increasing number of nodes.

Better Strength Node Enters in Service Path

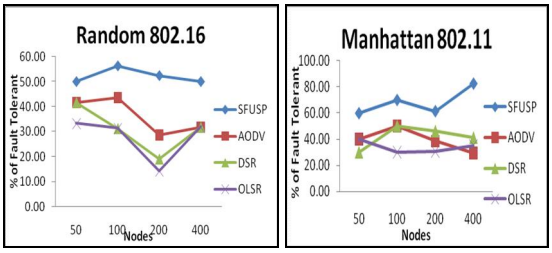

Fig. 15 Fault-tolerant success for various protocols under Random and Manhattan Mobility Models with mac layers IEEE 802.16, and 802.11

For Random 802.16, Sfusp is showing a better performance than other three protocols, but other protocols are showing fluctuating one. For Manhattan 802.11, Sfusp is showing a good performance for lesser number of nodes than higher number of nodes where, Olsr is showing the least performance but a consistent one. Aodv and Dsr are showing a variation in performance with varying number of nodes.

Unnecessary Node in Service Path

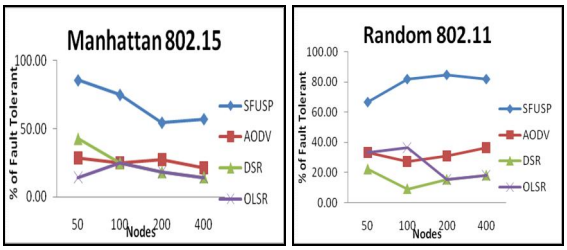

Fig. 16 Fault-tolerant success for various protocols under Manhattan and Random Mobility Models with mac layers IEEE 802.15, and 802.11

For manhattan 802.15, Sfusp is showing a better tolerant for lesser number of nodes, but it is ahead of other protocols. All other three protocols are maintaining a steady performance below 50 percentage. For random 802.11, Sfusp is showing a better performance ahead of other protocols, but others are showing a lesser percentage for higher number of nodes.

Average Path Re-establishment Time after Fault Occurrence

Servicing Nodes Move Away from Service Path

For Brownian 802.11, Sfusp is showing a better path re-establishment time than other protocols for all density of nodes. Aodv shows a steady outcome. For manhattan 802.16, all protocols have a same type of performance for less

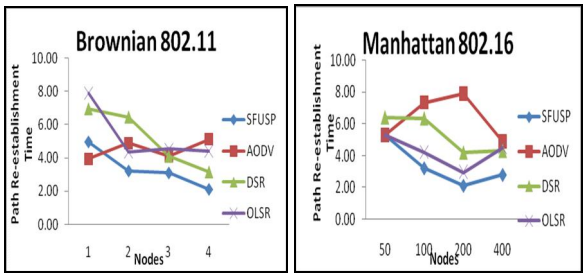

Fig. 17 Path Re-establishment Time for various protocols under Brownian and Manhattan Mobility Models with mac layers IEEE 802.11, and 802.16

Density nodes, for increasing number of nodes nodes, Sfusp is showing good performance than other protocols.

Better Strength Node Enters in Service Path

For Brownian 802.15, all protocols are showing a same path re-establishment time for lesser nodes, but Sfusp and Aodv are showing a good gain in time for more number of nodes. For random 802.16, all protocols have more or less

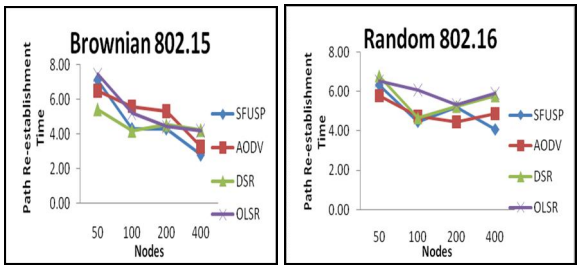

Fig. 18 Path Re-establishment Time for various protocols under Brownian and Random Mobility Models with mac layers IEEE 802.15, and 802.16

equal path re-establishment time. For more number of nodes, Sfusp is taking less time in path creation than other protocols.

Unnecessary Node in Service Path

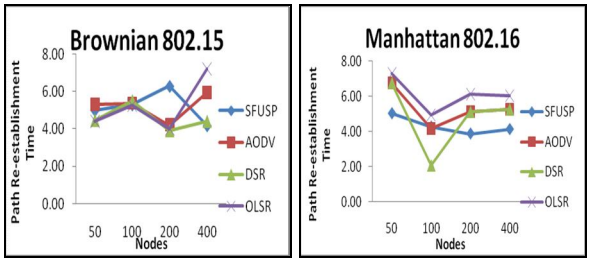

Fig. 19 Path Re-establishment time for various protocols under Brownian and Manhattan Mobility Models with mac layers IEEE 802.15, and 802.16

For Brownian 802.15, Sfusp is showing a good gain in path re-establishment time for increasing in number of nodes. But, other protocols are showing a low performance for higher number of nodes except Aodv. For Manhattan 802.16, Sfusp is showing a constant rate in path re-establishment for increasing in number of nodes, while other protocols are showing a varying performance. The better performance of Sfusp is because of the proactive nature of knowing about the nodes characteristics with reactive functionalities.

6 Conclusion

In this paper we propose a proactive fault-tolerant routing scheme with clustering and self-elimination techniques, to establish a reliable route among mobile devices of varying mobility and mac in heterogeneous environment. The proposed scheme works with efficient broadcasting technique that helps in finding best nodes in the network for communication with respect to service, transmission and stability. The routing scheme was tested in various fault arising scenarios and the results are compared to other manet protocols under different mobility conditions for different mac layers. It is found that for various routing and route maintenance metrics, our scheme shows better efficiency than other protocols because, it can handle the fault situations with its proactive knowledge and reactive sense of acting upon the situations. Our next work is to extend our routing capability for migrating services when fault occurs to deliver a streaming performance in heterogeneous environments.

References

[01] Joe C. Chan, Doan B. Hoang, "*Service Architecture For Integrating Manets With Heterogeneous Ip Networks*", IEEE Communications Society / Wcnc 2005, Pp:2270-2275.

[02] Shin-Jer Yang And Hao-Cyun Chou, "*Design Issues And Performance Analysis Of Location-Aided Hierarchical Cluster Routing On The Manet*", 2009 International Conference On Communications And Mobile Computing, Pp:26-31.

[03] Biao Zhou, Zhen Cao, Mario Gerla, "*Cluster-Based Inter-Domain Routing (Cidr) Protocol For Manets*", IEEE 2009.

[04] Fu Yongsheng, Wang Xinyu, Li Shanping, "*Performance Comparison And Analysis Of Routing strategies In Mobile Ad Hoc Networks*", 2008 International Conference On Computer Science And Software Engineering, Pp:505-510.

[05] Anahita Naghshegar, Arash Dana, "*Topology Control Scheme In Manets For Aodv Routing*", Ictta'08.

[06] Luo Junhai, Ye Danxia, Xue Liu, And Fan Mingyu, "*A Survey Of Multicast Routing Protocols For Mobile Ad-Hoc Networks, IEEE Communications Surveys & Tutorials*", Vol. 11, No. 1, First Quarter 2009, Pp:78-91.

[07] Ljubica Blazevic, Jean-Yves Le Boudec, And Silvia Giordano, "*A Location-Based Routing Method For Mobile Ad Hoc Networks*", IEEE Transactions On Mobile Computing, Vol. 4, No. 2, March/April 2005, Pp:97-110.

[08] Zhang Jianwu, Zou Jingyuan, Zhao Qi, "*Manet Routing Protocol For Improving Routing Discovery Based On Aodv*", 2009 International Conference On Networks Security, Wireless Communications And Trusted Computing, Pp:197-200.

[09] Nadir Shah, Depei Qian, Khalid Iqbal, "*Performance Evaluation Of Multiple Routing Protocols Using Multiple Mobility Models For Mobile Ad Hoc*

Networks", Proceedings Ofthe 12th IEEE International Multitopic Conference, December 23-24,2008, Pp:243-248.
[10] N. Enneya, K. Oudidi And M. Elkoutbi, "*Network Mobility In Ad Hoc Networks*", Proceedings Of The International Conference On Computer And Communication Engineering 2008, Pp:969-973.
[11] Mingyang Zhang, Peter H. J. Chong, "*Performance Comparison Of Flat And Cluster-Based Hierarchical Ad Hoc Routing With Entity And Group Mobility"*, IEEE Communications Society, Wcnc 2009.
[12] Ebrahim Mahdipour, Ehsan Aminian, Mohammad Torabi, Mehdi Zare, "*Cbr Performance Evaluation Over Aodv And Dsdv In Rw Mobility Model*", International Conference On Computer And Automation Engineering, 2009, Pp: 238-242.
[13] W. Zhao H. Zhang, "*Proactive service migration for long-running Byzantine fault-tolerant systems*", IET Softw., 2009, Vol. 3, Iss. 2, pp. 154–164.
[14] Vaskar Raychoudhury, "*Efficient and Fault Tolerant Service Discovery in MANET using Quorum-based Selective Replication*", IEEE, 2009.
[15] Michael S. Thompson, Scott F. Midkiff, "*Experiences Using IEEE 802.11b For Service Discovery"*, Proceedings Of The Fourth Annual IEEE International Conference On Pervasive Computing And Communications Workshops (Percomw'06).
[16] Kamran Etemad, "*Overview Of Mobile Wimax Technology And Evolution*", IEEE Communications Magazine, October 2008, Pp: 31-40.
[17] Jose Luis Jodra, Maribel Vara, Jose M Cabero, Josu Bagazgoitia, "*Service Discovery Mechanism Over Olsr For Mobile Ad-Hoc Networks*", Proceedings Of The 20th International Conference On Advanced Information Networking And Applications (Aina'06).
[18] Aamir Saeed, Laiq Khan, Nadir Shah, Hashim Ali, "*Performance Comparison Of Two Anycast Based Reactive Routing Protocols For Mobile Ad Hoc Networks*", IEEE 2nd International Conference On Computer, Control And Communication, 2009. Ic4 2009.
[19] www.isi.edu/nsnam/ns/
[20] Jyotsna Rathee & Anil Kumar Verma, "*Simulation and Analysis of DSDV protocol in Manets*", International Journal of Information Technology and Knowledge Management, July-December 2009, Volume 2, No. 2, pp. 441-444.
[21] Pangun Park, Piergiuseppe Di Marco, Carlo Fischione, Karl Henrik Johansson,"*Adaptive IEEE 802.15.4 Protocol for Reliable and Timely Communications*", pp.1-18.
[22] http://www.it.uc3m.es/celeste/pdp/
[23] Joa-Hyoung Lee and In-Bum Jung, "*Speedy Routing Recovery Protocol for Large Failure Tolerance in Wireless Sensor Networks*", Sensors 2010, pp:3389-3410

Mr. R S Shaji received his MCA degree from Manonmaniam Sundaranar University, Tirunelveli and M.Tech. in Computer Science and Engineering from Pondicherry University. He is currently a Research Scholar in Manonmaniam Sundaranar University, Tirunelveli, India. Also, he is working as an Assistant Professor in the Department of Computer Applications at St. Xavier's catholic college of Engineering. He has totally, 10 years of teaching experience and 5 years of industrial experience. His research interests are Service Discovery and routing in Mobile Adhoc networks, Pervasive Applications and High Performance Computing.

Dr. R S Rajesh is currently working as Associate Professor in the Department of Computer Science and Engineering in Manonmaniam Sundaranar University, Tirunelveli, India. He has nearly twenty years experience in teaching and has published more than 50 research papers in national and international journals. He has attended and presented papers in number of conferences in and outside India.

B Ramakrishnan is currently working as Associate Professor in the Department of Computer Science in S.T. Hindu College, Nagercoil. He has a teaching experience of nearly 25 years for both undergraduate and post graduate courses. He has completed his post graduation in the field of Electronics and Computer Science.