Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

# A Novel Message Driven Local Repair Algorithm for MANET

A.E.NARAYANAN[1], A.VINCENT JEYAKUMAR[2]
[1,2]Periyar Maniammai University, Thanjavur, Tamilnadu, 613403

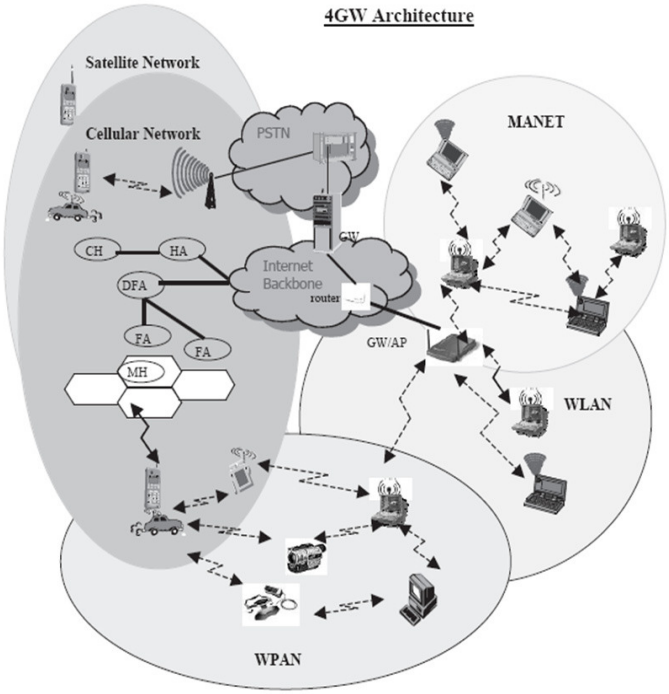
## Abstract

Mostly MANET is used for emergency situation like disaster management which has the task of supplying QoS in an effort to provide the right information to the right place at the right time. In this scenario with the capability to send and receive reliable real-time video and voice packets, the involved network must be able to provide a certain level of QoS. So thet the QoS impact in the face of network failures must be minimized in MANET like temporary emergency networks..In such a situation,when an intermediate node fails the network has no built in protocol to respond to this link breakage, the connection will have to be rerouted from the source and QoS will have to be re-established. This global fault-tolerance method means the source will have to recomputed and renegotiate a new QoS path which, depending on the network size, could be costly in terms of computation and communication time necessary for path negotiation (negotiation includes implementation of the new path). In contra , the proposed fault-tolerant algorithm makes it possible for the Intermediate nodes to efficiently repair the failed connection locally and reroute the packets to the destination.

***Keywords:Quality of service(QoS) , Fault tolerant routing ,message driven***

## I. Introduction

Over the decades, the use of personnal communication devices like mobile phones, personal digital assistants(*PDAs*) and mobile computers have taken an exponential growth. This tendency is reinforced when the cost of these small devices are reduced and further equipped with one or more wireless interfaces. The wireless interfaces allow the devices to get connected with the
access points available in various location such as air ports, railway stations, restaurants, city centers etc.. At the same time, they also enable the devices to interconnect directly with each other in a decentralized way and *self-organize* into "*Ad Hoc Networks.* During the situation of natural calamity, when the fixed infrastructure networks damaged , for the disaster management and mitigation the MANET is very much relied upon.




The fault tolerant approach is used in possibly prevent the malfunctioning node will affect the overall task of the network. Fault tolerance is used to enhance system reliability. It may be of different types as follows:

a. Fault tolerance in Node Failures
b. Fault tolerance in Link failure and Network Failure
c. Fault tolerance in Transmission Power and Energy
d. Fault tolerance using check-pointing, message logging, reducing overload etc.

## II.Fault-tolerance in QoS Ad Hoc Networks

Chen and Nahrstedt [26] propose fault-tolerance techniques in an effort to reduce the impact on QoS disruptions due to link failures caused by network dynamics. It is important to note that Chen and Nahrstedt only consider applications which do not require hard guarantees. "Soft QoS means that there may exist transient time periods when the required QoS is not guaranteed due to path breaking or network partition" [26]. Further, Chen and Nahrstedt state that many multimedia applications accept soft QoS and use adaptation techniques to reduce the level of QoS disruption [6], [15], [28]. One technique presented is to repair the broken path at the node failed by shifting the traffic over to a neighboring node and then routing around the breaking point. This method avoids the costly process of rerouting the traffic from the source.

The second technique involves using a multilevel *path redundancy* scheme. The idea is to establish multiple paths for the same connection. The *First-Level Redundancy* sends all data along all paths independently. independently. This redundancy level is used for 'critical' QoS connections. The *Second-Level Redundancy* sends data along only the primary path and uses any secondary paths only in the event that the primary path is lost. This redundancy level is used for QoS connections which can tolerate a certain degree of QoS failure. The *Third-Level Redundancy* is similar to the second level

except the secondary paths are not reserved; only calculated. If a failure should occur, an attempt will be made to reserve the secondary path. The first technique, the repair algorithm, is proposed as the single approach to the following cases:

1) The source moves out of range of the first intermediate node in the path
2) An intermediate node moves out of range of either a preceding or successive node (preceding node is on the source's side of the intermediate node, successive is on the destination's side of the intermediate node)
3) The destination moves out of range of its preceding node (preceding node is the last intermediate node before the destination)
4) Any node in the path leaves the network

When case 2 occurs, the preceding node broadcasts a *repair-requesting* message to all its neighbors asking if any of them are able to take over the job of the defunct intermediate node. The neighbors that have links to the successive node reply their resource availabilities to the preceding node. If, based on the replies, the preceding node finds node *i* has sufficient resources for that role, it adds the link from itself to node *i* to the routing path and then sends *i* a *path-repairing* message. When *i* receives the *path*

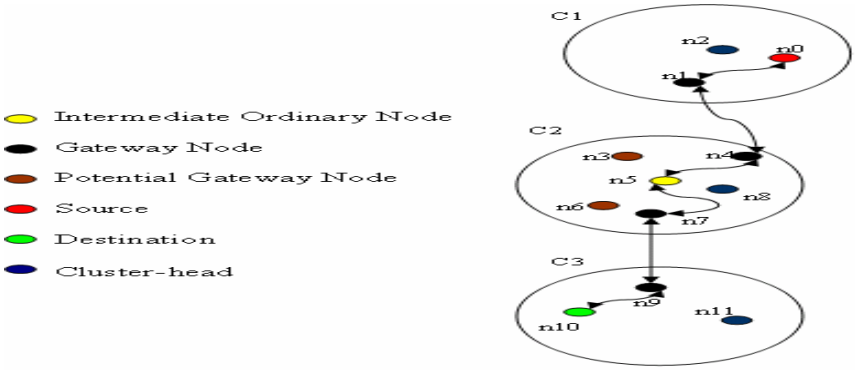



**Fig 2.Clustered ad hoc network**

*repairing* message, it reserves the required resources and adds the link from itself to the successive node to the routing path. Once the path has been repaired, a *path-validation* message is generated to insure that the repaired path does not violate any of the end-to end requirements. The *path-validation* message is sent to the destination which then sends the message to the source. The source then checks to see if the end-to-end requirements have been violated. If they have, the source will reroute the traffic or some QoS negotiation will take place with the user application. The performance metric used during simulation of the repair algorithm is the QoS ratio, defined as:

$$QoS\ ratio = \frac{Total\ QoS\ time}{Total\ QoS\ time + best\text{-}effort\ time}$$

Where *best-effort time* is defined as the amount of time spent repairing the brokenpath. The x-axis is the mobility ratio, defined as:

$$Mobility\ ratio = \frac{total\ moving\ time}{total\ stationary\ time + total\ moving\ time}$$

The simulation results provided include a single graph which shows for a mobility ratio of less than 10%, the QoS ratio is above 95%. As expected, the QoS ratio decreases as the mobility ratio increases. For a mobility ratio more than 35%, the QoS ratio is below 80%. The conclusion is that Chen and Nahrstedt's routing algorithm should not be used in networks with high node mobility.

## III. Proposed local repair algorithm

An intermediate (I) node is any node that supports a Quality of Services (QoS) connection. A defunct (D) node is a cluster node that previously was an I node; but has either moved out range or has failed. A gateway node (GWN) is defined as a cluster node which is used to communicate with a neighbor cluster. A potential GWN (P-GWN) is a node which has the ability to communicate with the same neighbor cluster as the current GWN. But it is only used in the event that the current GWN becomes a defunct node. As an example, in the following n3 is a P-GWN since it can communicate with n1 and the current GWN is n4 since it is currently communicating with n1. A cluster-head (CH) is a cluster node which has the responsibility of monitoring and updating a cluster table which records all QoS connections currently supported by the cluster. The CH is also responsible for initiating QoS connection repairs. A CH can also be a GWN. An ordinary node is a node that is neither a CH nor a GWN. In figure 2 n0 be the source, n10 the destination and P the QoS path. Each node i in P has a successive node except n10. Further, each node i in P has a preceding node except n0.

### 3.1 Situations for path breakage:

Ordinary I node moves out of range of a successor I node in the cluster (i.e., n5 moves out of range of n7)

1) Ordinary I node moves out of range of a predecessor I in the cluster (i.e., n5 moves out of range of n4)
2) I GWN moves out of range of a successor I node in the cluster (i.e., n4 moves out of range of n5)
3) I GWN moves out of range of a predecessor I node in the cluster (i.e., n7 moves out of range of n5)
4) I GWN moves out of range of a successor I GWN in the cluster (i.e., if n4 were connected to n7 and n4 moves out of range of n7)
5) I GWN moves out of range of a predecessor I GWN in the cluster (i.e., if n4 were connected to n7 and n7 moves out of range of n4)
6) I GWN moves out of range of a successor GWN in the cluster (i.e., n7 moves out of range of n9)

7) I GWN moves out of range of a predecessor GWN in the cluster (i.e., n4 moves out of range of n1)
8) I CH moves out of range of a successor or predecessor I node in the cluster (i.e., if P were such that n8 had n4 as a predecessor node and n7 as a successor node and n8 moved out of range of either node)

**3.2 Assumptions:**

- All nodes have a unique identifier
- Two nodes can be cluster members of the same cluster if and only if their Euclidean distance is ≤30m (approximate range of 802.11g)
- Nodes signal their presence via a periodic beacon message and the drifting in of a new node is realized when its new neighbors hear its beacon.
- When a node does not hear signals from a known neighbor within a certain amount of time, it assumes the neighbor to be either "dead" or out of range due to mobility
- Determining a node has failed or moved out of range will prompt the corresponding procedure
- All nodes have a single larger bandwidth interface (eg., FSO transceiver, directional RF transceiver, etc.) for each node they can communicate with via 802.11 ( the link state data is based on this link)

All procedures are atomic expect the Route_traffic (u) procedure and the procedures excepted for receipt of the PATH (v.rsrcs, dst) and CTS (u) messages. Order to discuss the relevant features of EECHSFT, it is assumed that, all gateway nodes in the network have path routing table entries for all network destinations. Also it is assumed that is assumed that the associated applications using this QoS network have soft QoS constraints and use adaptive techniques to help minimize QoS disruptions. Combinatorial stability is also adopted. Further, with this model, nodes have the ability to send and receive 802.11 best-effort traffic while sending and receiving QoS traffic along larger bandwidth, directional links. Finally, resources allocated for a QoS connection are de-allocated after a specified period of inactivity

**3.3 Messages:**

A node v uses message CH (v) to communicate with its neighbors, when it intends be a cluster-head. To become a part of cluster u node v will sent a message JOIN (v,u), to communicate to its neighbors. To require the resignation of any receiving cluster heads with weight ≤ w RESIGN (w) message is used. Source v uses PATH (v.rsrcs,dst) message to request each node u along the path of a new potential QoS connection to destination dst. The CTS (u) message is sent by destination v back to source u along the initialized path to finalize the resource allocations. The information about supported connections (v.NT), as well as the available resources (v.AT), of node v to the cluster-head will be sent by using PMETS (v.NT, v.AT, Cluster-head). The cluster node's QoS table is updated by broadcasting message CLSTR_PMTS_UPDT (v, CT) at regular intervals. The cluster head sends message REPAIR (ConnexPmets (p), v, Cluster-head) to notify node v to restore a connection using the information in the ConnexPmets (p) table. After the failed link has been repaired, to ensure the end-to-end QoS constraints sustainability QOS_VALID (u, v) is sent. The source u will initialize a

QoS validation message once LINK_REPAIRED (failed_node, v, u) message is received from the node that is new to the path v. When the attempts of v to repair the failure, fails REPAIR_FAILURE (v, Cluster-head) message is sent. When the failed connection supporting path p could not be repaired a message FAILED_CONNEX (failed_node, p, v) is sent to source v of source v of QoS path v. In different clusters gateways are created using HELLO (u, Cluster-head, Init) through neighboring nodes. Node u sends HELLO (u, Cluster-head, Init) message to receive HELLO (v, Cluster-head, Reply) in a periodic intervals.

**3.3 Procedures for Fault Tolerence:**

**3.3.1 Node_failure Procedure**

When node b knows the failure of node a, b checks, whether its role is cluster head and a was in its cluster. If it is so, b removes a from cluster (b). In this situation, if a was a node which was supporting a connection, cluster head b collects all cluster resources and all supported QoS traffic and decides the possible QoS paths. Then all suitable cluster nodes are advised by node b to support the appropriate QoS connection with the REPAIR message. If the cluster head determines an impossible situation, that is the cluster resources currently available cannot support all QoS connections, the cluster head sends a FAILED_CONNECT message to the sources, which are using the resources on the failed node and no path changes made in the cluster. If node a was the cluster head, supporting a QoS connection, the new cluster head will try to repair the fail connection of a if and only if, when it is an ordinary node with the next weight Wb in descending order after Wa (ie.,$\exists$ b | $\forall$z $\epsilon$ { $\tau$(b)-{a}}:wb>wz), then node b will be the new cluster head and tries to fix the failed connection of a. When b is an ordinary cluster member node and doesn't have the next weight Wb after Wa, and a was the cluster head, node b will wait for a period of time for the message CH(c) from node c, which possesses the next highest weight Wc. If the message CH (c) is not received by node b, then b joins to the cluster head with highest weight and sends its NT and AT to the new cluster head. The re-clustering process is invoked, when node b is not the new cluster head. As the time needed for the re-clustering process will increase the time of the connection recovery process, re-clustering is engaged in this situation. Since the combinatorial stability is already assumed, when the cluster head fails, there will be a slight in the cluster. As the information of potential cluster heads are announced through CT periodically, once a cluster head fails, other ordinary nodes wait for a short period of time to receive the CH (b) message from the next new cluster head. The short period is to manage propagation delay, processing delay and error. If the nodes don't receive the ch(b) message, with in the given time, they will decide their new roles.

**3.3.2. New_link procedure:**

When a new node 'a' is found by node 'b', it will check, whether a is a cluster head. If it is true and the weight is higher than the current cluster-head of cluster (b), a will be cluster-head and b sends PMETS message to a. In contrast, if b is the cluster-head and the member of its neighboring cluster-heads greater than b, the weight of the cluster-head d that violates the k=0 condition is fixed. If Wb>Wd, then node d will receive RESIGN message. If there is no cluster-heads such that Wb>Wd, b will no longer be a cluster-head and will join the cluster-head with biggest weight. Then node b will send PMETS message to the new cluster-head

**3.3.3 Route traffic (a)** procedure:

The associated application will make aware source node b, the need to route the new traffic. Node b verifies that its cluster members to see whether a is in this set of nodes. If a is within the cluster, the available resources of a are obtained from CT table. If the necessary resources are available, they are reserved and the traffic is sent. If the destination is not in the cluster, b forwards PATH (b, rsrcs,dst) to the cluster gateway nodes.

## IV. Result Discussion

The proposed message driven fault tolerant algorithm is tested in NS2 environment. The results are discussed with the existing global fault tolerant algorithm FDCB inter interms of the important QoS factors Throughput, Dropped packets, Recovery time.

The figure 3 shows that the recovery time of LRAFT is much better than the existing global technique. The plot drawn between recovery time and failures per second. As this algorithm rectify in the proximity itself the time taken for recovery is very less.

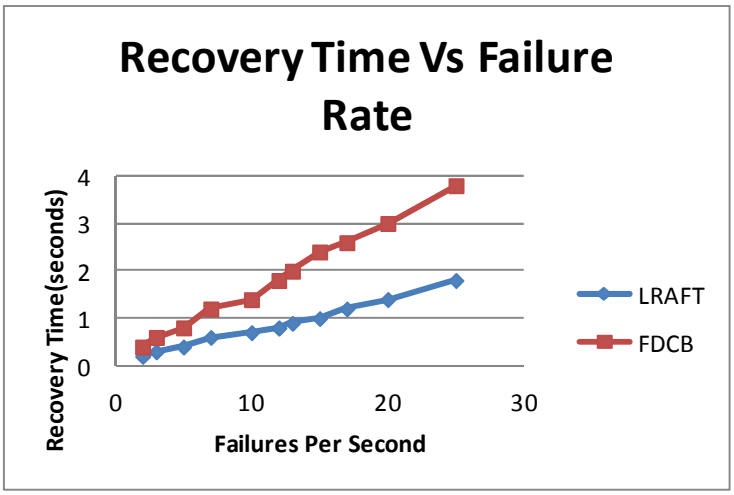



**Fig 3: Recovery Time Vs Failure Rate**

Figure 4 shows the plot between the dropped packets and failure rate. The number of dropped packets is high in FDCB.

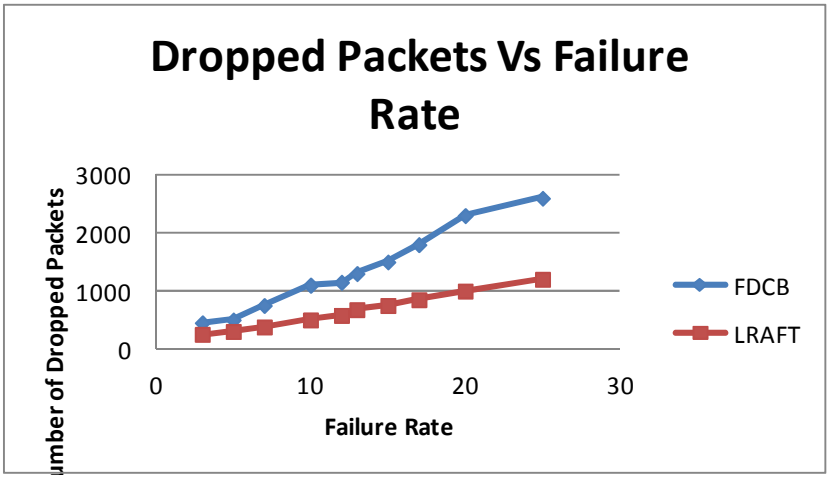


**Fig 4: Dropped Packets Vs Failure Rate**

The throughput is an important QoS factor as this decides the performance of the network. The optimal throughput can be calculated by dividing the realized throughput by throughput without failures. LRAFT provides better optimal throughput as shown in the graph.

*Percentage of optimal throughput =Realized throughput / Throughput without failures*

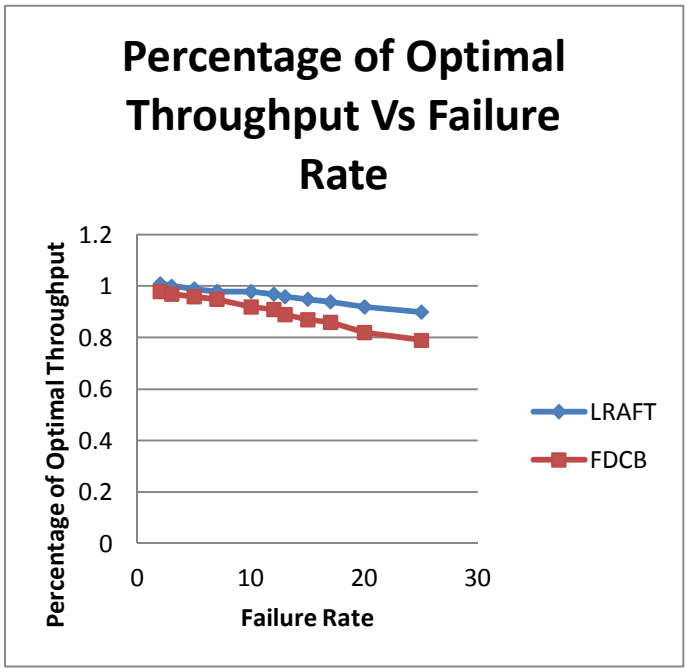



**Fig 5: Optimal Throughput Vs Failure Rate**

## V. Conclusions

The important design characteristics of the fault tolerant algorithm is explaied . One of the main objective is to employ a cluster head model to extenuate the link failures. In this algorithm the cluster head is made aware of the present corroborated QoS connections in the cluster by applying a cluster state knowledge sharing process. All cluster members of the cluster are also be shared the cluster state knowledge to support, when the cluster head failures. As this Fault tolerance algorithm handles the connection failures locally, rather rerouting the traffic from the source, which reduces the packet transmission delay. This fault tolerant approach dissent from Chen and Nahrstedt's work. In this work a clustered formulation is used that avert the disadvantages of Chen and Nahrstedt's repair algorithm . Esecially The limitation predecessor of the failed node is able to reach the successor of the failed node. There for this fault tolerent protocol show an progressive solution for fault-tolerance in MANET which support QoS.

As a summary this local repair method is more efficient thand than the global approach. This is ensured by performance metrics evaluation and analysis.

## References

1. D. Maheshwari and A.Dhanalakshmi, "Fault tolerance in Mobile ad hoc Network: A Survey," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.
2. Nargunam, A.S. and M.P. Sebastian. "Fully distributed cluster based routing architecture for mobile ad hoc networks," *Proceedings in the Wireless And Mobile Computing, Networking And Communications* (*WiMob'2005*), *IEEE International Conference*. 383-389. IEEE, 2005.
3. Ghosh, R. and S. Basagni, "Limiting the impact of mobility on ad hoc clustering,"*Proceedings of the 2nd ACM international workshop on Performance evaluationof wireless ad hoc, sensor, and ubiquitous networks*. 197-204, ACM Press: 2005.
4. *The network simulator--ns-2.* Version 2.29, Computer Software. Informaiton Sciences Institute, Marina del Rey CA, 2006.
5. Yuan Xue and Klara Nahrstedt, "Fault tolerant routing in mobile ad hoc networks," IEEE wireless communications and networking conference (WCNC), New Orleans, Louisiana, pp. 1174–1179, March 2003.
6. Chen, S., *Routing Support For Providing Guaranteed End-to-end Quality-of-Service*, PhD dissertation. University of Illinois: Urbana. 1999.