International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 5, Issue 4, July - August 2016 ISSN 2278-6856

Comparative study on Fault tolerant routing Protocols in Mobile Ad hoc Networks

Suneel Kumar Duvvuri[1], Dr. S Ramakrishna[2], Dayalan M[3]

[1]Research Scholar, Department of Computer Science,
Sri Venkateswara University, Tirupati, India

[2]Professor, Department of Computer Science,
Sri Venkateswara University, Tirupati, India

[3]Research Scholar, Department of Computer Science,
Sri Venkateswara University, Tirupati, India

Abstract

Designing of a stabilized routing protocol in MANET is challenging because of the open medium and frequent restructuring of paths. The faults in the transmission failures create malfunctioning nodes in the network. These faults can diminish the performance of the routing protocols. Identification and elimination of faulty nodes during data transmission is very crucial because of unavailability of path information. One of the research challenges for MANET is the design of efficient fault tolerant routing protocol to achieve highest packet delivery ratio. In this article the recent implementations on fault tolerant routing protocols is presented. The fault tolerant routing protocols based on numerical estimation is discussed.

Keywords: MANET, Fault Tolerance, E^2FT, WEFTR, LAFTRA

1. INTRODUCTION

Mobile Ad hoc Networks are self configuring cooperative multi hop networks. MANET comprises of autonomous nodes without any central administrative control. MANETS are constituted by several limitations such as limited transmission range of wireless links, high mobility, limited bandwidth and constrained power resources [1].

In MANETs, there is no dedicated router, so routing task need to be performed by nodes itself. Each node in Mobile Ad hoc Networks is intelligent to serve as a router and end devices.

Routing in MANETs is a challenging problem due to dynamic movements of nodes. The nodes forward packets from the source to destination in most efficient path. For the nodes that are not in communication range, the intermediate nodes cooperate in forwarding the packets.
In view of constant migration of nodes, the task of establishing efficient route between nodes is hard-won. Many routing protocols are developed to win the race.

In this paper a comprehensive survey of different fault tolerant routing protocols has been presented. Design issues, pros and cons, complexities and performances of these protocols are summarized.

The remainder of this paper is organized as follows. In section 2, Routing Protocols in MANETs has been discussed. In section3, the major challenges in designing Fault tolerant routing protocols have been discussed. In section 4, the well known Fault tolerant routing protocols namely E^2FT, WEFTR and LAFTRA will be explained with detailed operational descriptions will be discussed. Section 5 describes the performance evaluation and Section 6 concludes the paper.

2. ROUTING PROTOCOLS IN MANET:

Routing protocols in MANETs are catalogued based on how they are maintaining routing information. Typically those are catalogued as Proactive, Reactive and Hybrid Protocols [2] [3].

A Proactive routing protocol maintains the routing information to all the nodes for all the time in each nodes routing table. Examples of Proactive routing protocols are Destination Sequenced Distance Vector (DSDV) [4] and Optimized Link State Routing (OLSR) [5]. Proactive routing protocols are advisable for fewer numbers of nodes.

A Reactive routing protocol is an On-Demand routing protocol, which is based on Query-Reply mechanism. It establishes routes to destination when ever communication needed. Examples of Reactive routing Protocols are Ad hoc On demand Distance Vector (AODV) [6] and Dynamic Source Routing (DSR) [7].

A Hybrid routing protocol mingles all the aspects of Proactive and Reactive routing protocols.
The Routing Protocols in MANETs operate in either Unipath are Multipath.
In Unipath routing, the packet get to destination by pair of nodes through a single unique path. AODV, DSR protocols are the prominent Unipath routing protocols.
In Multipath routing, multiple redundant packets sent through different multiple paths between source-destination via a pair of nodes.

Multipath routing protocols [8] are more reliable than the Unipath routing protocols because of redundancy. Multipath routing protocols have maximum likelihood of successful delivery of packets by at least one of the path but submit superfluous overhead in the networks holding ability.

Some of the problems encountered in Unipath protocols also emerge in Multipath protocols. Mobility of nodes has direct inference on cached information of intermediate nodes which becomes out of date in quick succession, so the routing performance degraded greatly.

3. FAULT TOLERANCE IN MANETS:

A node actively participating in routing at one point of time may become weak at a later point. There are several causes that exposes the system failure in MANETs include

- Errors occurred during Transmission
- Frequent node failures due to energy constraints.
- Link and route breakages
- Packets loss a result of congested nodes or links

The medium of transmission of information in MANETs is wireless, which is relatively pretended and more exposed to noise. MANETs composed with battery powered devices, which are awkward; low powered and needs frequent recharging.

To ensure reliability, a "tolerant" mechanism needed for preventing malfunctioning nodes that lead to network failure while making routing decisions [9].

3.1: Challenges in designing fault tolerant routing protocols:

Enormous Overhead**:** Multipath routing protocols are tolerant to link failures and route breakages than Unipath routing protocols but those are unsuitable for MANETs due to excessive overhead by replicated packets.

Duplicate packet processing**:** In Multipath routing, duplicate packets sent using different paths to ensure reliability. So, Fault tolerant routing protocols need a mechanism at the destination to filter out these duplicate packets

Protracted Delay**:** Multipath routing protocols work by sending packets usually travels through faraway hops compared to the Unipath routing. So huge delay incurred in transmission.

Special Control Message for Path Dynamics**:** For choosing a Fault tolerant path from all the available multiple paths the source node needs statistical information of each node along that path. So, a feedback tool required back to source from destination. These special Control packets can crush the network. Because of limited bandwidth, these packets can affect the performance of a network. Fault tolerant routing protocol requires a mechanism to limit Special Control Messages.

4. FAULT TOLERANT ROUTING PROTOCOLS

4.1: End to End Estimation based Fault Tolerant routing algorithm (E^2FT):

Xue and Nahrstedt (2003) [10] designed an efficient algorithm called End to End Estimation based Fault Tolerant routing algorithm. E^2FT consists of two major phases: 1. Route estimation phase 2. Route selection phase

In Route estimation phase the source node estimates the packet delivery ratio γ(p) of a path p by sending a number of data packets along that path and measuring the number of packets that can reach the destination. A feedback mechanism is implemented either at network layer as an Acknowledgment or at transport layer as a piggybacking message to TCP Acknowledgment. Clearly, for a path p, γ(p) is calculated by the source node as

$\gamma(p) = \frac{n'}{n}$, where n is the number of sent packets and n' is the number of received packets along that path. The accuracy of γ(p) depends on the size of n.
The feedback control packets sent by the destination to source introduce more control overhead and increases the route estimation time. In order to balance between overhead and delay γ(p) is calculated iteratively.
In the each iteration, a set of "b" packets is sent. Initially γ(p) is set to 0. In i^{th} iteration $\gamma(p)_i$ is calculated as

$$\gamma(p)_i = (1 - \frac{1}{i})\,\gamma(p)_{i-1} + \frac{1.b'}{b}$$

The estimation results of different paths depend on different packets sent along that path. So, each value of γ(p) have different accuracies. For achieving unified estimated value from the calculated different accurate values, the authors suggested α-estimation, as a level of confidence.
For a path p, the α-estimation of γ(p) is calculated as

$$\gamma(p) = \max\left[\gamma(p) - \frac{1}{\sqrt{4n(1-\alpha)}}, 0\right]$$

In Route selection phase, at the beginning all the paths between a pair of source-destination nodes are used to sent packets. After the calculation of accurate path estimation values the fair values are selected permanently for routing and all other paths are dropped. This is done by comparing calculated values with expected packet delivery ratio γ^*.

4.1.1: Numerical Estimation:

Parameters:

Expected packet delivery ratio $\gamma^* = 0.85$

Level of Confidence $\alpha = 0.85$

In figure 1, all the available paths between Source node (A) to Destination node (I) are as follows

Path1: A→B→C→I

Path2: A→D→E→I

Path3: A→F→G→H→I

Path4: A→B→E→I

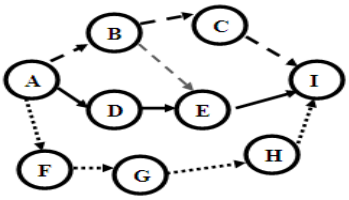

Figure 1: Network Topology

Let us assume that source sends 30 packets at each iteration (i.e., b=30) and E, G are declared as faulty nodes. Initially, all four paths will be used. Table 1 illustrates the Iterations and estimated values.

Table1: Iterations of E2FT Algorithm

	Path	Packets sent	Packets Received	γ(p)
Iteration 1	1	30	30	0.7642977
	2	30	15	0.2642977
	3	30	10	0.0976311
	4	30	15	0.2642977
Iteration 2	1	60	60	0.8333333
	2	60	30	0.3333333
	3	60	30	0.3333333
	4	60	25	0.25
Iteration 3	1	90	88	0.841695
	2	90	50	0.4194728
	3	90	45	0.3639172
	4	90	40	0.3083617
Iteration 4	1	120	118	0.8654822
	2	120	68	0.4488155
	3	120	55	0.3404822
	4	120	55	0.3404822

After 4th iteration Path1 will be confirmed for further data transmission as a result of $\gamma(p) > \gamma^*$ and all the remaining paths will be dropped.

4.2 Weak Estimation based Fault Tolerant Routing (WEFTR):

Oommen and Misra [11] proposed the Weak Estimation based Fault Tolerant Routing Algorithm using statistical Weak Estimation Learning principle. This algorithm estimates the packet delivery probability of a node by an efficient estimator called Stochastic Learning Weak Estimator [12]. As in E2FT, the WEFTR also consists of route estimation and route selection phases.

In Route selection phase, the source node sends n number of packets along a path p and calculates packet delivery probability $\hat{\gamma}_0(p) = \frac{n'}{n}$

This Packet delivery probability estimate is refined further in number of iterations. In the each iteration, a set of packets will be transmitted through all the available paths between source and destination pairs and Packet delivery probability will be estimated by using Weak Estimation Learning as follows:

$$\hat{\gamma}_0(p) = \begin{cases} \lambda \times \hat{\gamma}_0(p) \text{ if the path deos not forward the packet} \\ 1 - \lambda \times [1 - \hat{\gamma}_0(p)] \text{if the path forwards the packet} \end{cases}$$

Where λ is the learning parameter, such that $0 < \lambda < 1$.

The ideal value for λ lies in the interval [0.9, 0.99]

In Route selection phase, the paths will be confirmed for further consideration in future routing if $\hat{\gamma}(p) \geq \gamma^*$, Where γ^* is the minimum required packet delivery probability.

A path with minimum packet delivery estimation value p_{min} from all the available paths π will be dropped when the packet delivery probability of remaining paths guarantees the minimum required packet delivery probability such that at least a copy of packets will be received among all of them.

The path dropping constraint is calculated as follows:

$$\hat{\gamma}_{\frac{1}{m}}(\pi') = 1 - \pi_{p\in\pi'} [1 - \hat{\gamma}(p)]$$

Where $\pi' = \pi - \{p_{min}\}$

The path p_{min} will be dropped from estimation when $\hat{\gamma}_{\frac{1}{m}}(\pi') \geq \gamma^*$

Table 2, represents the progressive calculation of $\hat{\gamma}_0(p)$ for 20 packets on recipient of acknowledgements.

Table2: Calculations of Expected Packet delivery Probabilities for 20 packets

Packet Sent	1	2	3	4	5	6	7	8	9	10
Delivery Status	Success	Success	Success	Success	Success	Failed	Success	Success	Success	Failed
	0.1	0.19	0.271	0.3439	0.40951	0.36856	0.4317	0.48853	0.53968	0.48571
Packet Sent	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Delivery Status	Failed	Success	Success	Failed	Failed	Failed	Success	Success	Success	Failed
	0.43714	0.49343	0.54408	0.48968	0.44071	0.39664	0.45697	0.51128	0.56015	0.50413

4.2.1 Numerical Estimation:

For our Network Topology as in Figure1, the iterations of WEFTR algorithm are shown in Table3.

Let us assume that source sends 30 packets at the each iteration through all the selected paths and E, G are declared as faulty nodes.

Minimum Expected packet delivery probability γ^* =0.85 and learning parameter λ=0.9

Table3: Iterations of WEFTR Algorithm

	Path	Packets sent	Packets Received	γ(p)	Status
Iteration 1	1	30	30	0.957608842	Confirmed
	2	30	28	0.926755405	Confirmed
	3	30	25	0.8095368	Dropped
	4	30	21	0.426894715	Dropped
The Packet delivery Estimate Probability of Path1 and Path2 are above the γ^*. So,Path1 and Path2 will be confirmed. The dropping algorithm now chooses Path 4 as minimum i.e., p_{min} = 0.426894 and path dropping constraint $\gamma_{1/m}$(p)=0.999408 which is greater than γ^*. so, Path4 will be dropped. Again Path 3 selected and dropped because γ1/m(p)=0.996895					
Iteration 2	1	60	58	0.949826629	Confirmed
	2	60	51	0.712371476	Dropped
The Packet delivery Estimate Probability of Path1 is more than the minimum required Probability So,Path1 will be confirmed for all future routing					

4.3 Learning Automata based Fault Tolerant Routing Algorithm:

Misra et al [13]. Proposed learning automata based fault tolerant routing approach for selecting optimized paths among multiple paths having faulty nodes. The basic idea behind using learning automata is to gain knowledge from the environment In LAFTRA, a self learning machine code i.e., an automaton [14] is stationed in every node, which will monitor the traffic flow through that node. Based on the feedback from the network the learning process takes appropriate actions. The automaton continuously gains the healthiness of its neighbor nodes. Before forwarding the packet each node itself decides which neighbor to be picked to forward the packet to achieve high packet delivery ratio. At the time of learning process, each node calculates the goodness value of the path based on the responses from the other nodes. The response from the node will be sent through an Acknowledgement.

The intermediate node, Up on receiving a positive feedback will be rewarded contrarily will be penalized. Subsequently the "Goodness Value" of the path is calculated. Each node rewards and penalizes the path from itself to destination.

A node will be rewarded by adding a reward constant R (0<R<1) to its previous goodness value G after forwarding the packet successfully. The robustness of the path selection is based on R value. The ideal value for R should be less than 0.1.

A node will be penalized by decreasing the goodness value of the node by a Penalty constant P (0<P<1) if failed to forward the packet. The consistency of the path selection is based on P value. The ideal value for P should be in between 0.3 and 0.5. The Low P value does not detects the faulty paths effectively.

The pseudo codes for reward and penalize functions are

```
reward()
{
  if(currentnode==destination)
  {     G=G+R;
        Y=G;/*Goodness value of the Path*/
  }
  else
  {     G=G+R;
        if(Y<=T)/*Thresold value:Minimum required Goodness value*/
        {       Y=weightfactor*G+(1-weightfactor)*Y; /*Update Y*/
        }
  }
}
penalize()
{
  if(currentnode==destination)
  {     G=G*P;
        Y=G;
  }
  else
  {     G=G*P;
        Y=weightfactor*G+(1-weightfactor)*Y;
  }
}
```

In this algorithm, A Goodness value table is maintained by every node to store goodness values of each path. This table gets updated by an update message packet which will be sent to the neighbors during Acknowledgment/route reply. When a node needs to send a packet to the destination first it will check into Goodness value table and choose the optimal next hop to send the data. The format of Goodness value table and Update message packet header are shown below.

Table 4, represents the progressive calculation of Goodness Value Y for 10 packets on recipient of delivery status.

Table4: Calculation of Path Goodness at a node based on delivery status

Packet Sent	1	2	3	4	5	6	7	8	9	10
Delivery Status	Success	Success	Success	Failed	Success	Failed	Failed	Success	Success	Failed
Goodness	0.0015	0.00427	0.00813	0.00916	0.01154	0.01168	0.01086	0.01168	0.01387	0.01375

4.3.1 Numerical Estimation:

For our Network Topology as in Figure1, the iterations of LAFTRA algorithm are shown in Table3.

Let us assume that source sends 50 packets to destination and E, G are declared as faulty nodes.

(Reward Constant=0.09, Penalty Constant=0.5, Threshold=0.65 and weight=0.15)

Iteration1: Packets forwarded through P1, P2, P3 and P4 are 22,13,9 and 6 respectively. After receiving the acknowledgments the goodness values of the paths at source node is

P1: 0.5798 P2: 0.1445 P3: 0.0617 P4: 0.0189

Iteration2: The Goodness value of the Path1 is high, so before forwarding the packet the source node checks the goodness value table and sent more packets through the path having leading goodness value. After sending 35,10,3 and 2 packets along the paths P1, P2 P3 and P4 the Goodness values updated as

P1: 0.6624 P2: 0.1576 P3: 0.0568 P4: 0.0137

Since Path1 goodness value bypasses threshold value Path1 will be confirmed for further routing.

5. PERFORMANCE EVALUATION:

An E^2FT algorithm does not influenced by the mobility of the nodes. Simulation [15] studies indicate that E^2FT performs outstanding when the percentage of faulty nodes increased. Due to the heavy learning process LAFTRA incurs slight delay but achieves high packet delivery ratio than E^2FT and WEFTR. LAFTRA deals with link failures effectively than the other two. Overhead of WEFTR algorithm significantly decreased than E^2FT. LAFTRA reduces overhead in low mobility environments. These algorithms are designed to achieve fault tolerance and no significance is given to the energy resources of the node. Even though these algorithms increase the packet delivery probability in faulty environments but the energy consumption of these protocols increased significantly.

6. CONCLUSION AND FUTURE WORK:

The Mobile Ad hoc Networks have been recent interests of many researchers. Most of the researches have been motivated to design an effective and efficient protocol for MANETs. This paper presented a survey of most recent fault tolerant routing protocols for MANETs. The surveyed protocols showed that significant improvement in the network performance in the faulty environments. Yet, it is hard to find a single protocol to improve all the performance metrics. Selection of these protocols is depends up on the application. Still some of the objectives such as energy efficiency need to be improved.

In future, this work can be extended to improve the algorithms to achieve high packet delivery ratio while considering energy efficiency.

References

[1]. C.Sivaram murthy, B.S.Manoj, “Adhoc wireless networks: Architectures and protocols”, Pearson Edu. 2004.

[2]. Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, “A review of routing protocols for mobile ad hoc networks”,Elsevier Ad hoc Networks 2004

[3]. Changling Liu, Jorg Kaiser, “A Survey of Mobile Ad Hoc network Routing Protocols” kaiser@ivs.cs.uni-magdeburg.de

[4]. Charles E. Perkins and Pravin Bhagwat, “Highly Dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. 1994

[5]. T.Clausen and P. Jacquet “Optimized Link State Routing Protocol (OLSR)” RFC 3626, IETF Network Working Group, October 2003

[6]. C.E. Perkins and E.M. Royer, “Ad hoc on-demand distance vector routing”, In Proceedings of the 2nd IEEE workshop on Mobile Computing Systems Applications, New Orleans, Louisiana, 1999, pp.207-218

[7]. D.B Johnson and D.A. Maltz, “Dynamic Source Routing in Ad hoc Wireless Networks”, Mobile Computing, 1996, pp.153-181

[8]. M.Tarique et al., “Survey of Multipath Routing Protocols for Mobile Ad hoc networks,” Journal of network and computer applications, 2009. pp. 1125-1173

[9]. Y. Xue, Nahrstedt, “Fault Tolerant routing in Mobile Ad hoc Networks”, In Proceedings of the IEEE wireless communications and networking conference, New Orleans, Louisiana, 2003, pp. 1174-1179

[10]. Y. Xue, Nahrstedt, “Providing Fault Tolerant Ad hoc Routing service in adversarial environments”, Wireless personal Communications, Vol.29, 2004. pp. 367-388

[11]. Oommen B.J, Misra S, “A fault tolerant routing algorithm for mobile Ad hoc networks using a stochastic learning based weak estimation procedure”, In Proceedings of the IEEE wireless and mobile computing, networking and communications international conference, New Orleans, Louisiana, 2006. pp. 31-37

[12]. J. Oommen and L. Rueda, “A New Family of Weak Estimators for Training in Non-Stationary Distributions”, In Proceedings of the 2004 International Symposium on Structural, Syntactic, and Statistical Pattern Recognition, Lisbon, Portugal, August 2004, pp. 644-652.

[13]. Sudip Misra et al., “A learning automata-based fault-tolerant routing algorithm for mobile ad hoc networks”, Journal of Supercompt(2012) 62:4–23

[14]. Thathachar MAL, Sastry PS (2003) Networks of learning automata. Kluwer Academic, Dordrecht

[15]. ns-2: Network Simulator 2, http://www.nsnam.org (2011)

AUTHOR

Suneel Kumar Duvvuri received his MCA degree from Andhra University in 2012. He is currently a research scholar, Computer science department in Sri Venkateswara University, Tirupati. His research interests include Data Structures, Mobile ad hoc networks, vehicular ad hoc networks, Artificial Intelligence and Machine learning

Dr. S Ramakrishna received Ph.D. in mathematics in 1988. He is a professor in Computer science department, Sri Venkateswara University, Tirupati. His research interests include Fluid Dynamics, Network Security and Data Mining. He has guided about 20 Ph.D., students so far. To his credit he has published around 95 research articles in reputed journals.

Dayalan M received his MCA from SV University in 2010. He is currently research scholar, Computer science department in Sri Venkateswara University, Tirupati. His research interests include Mobile Computing and Wireless Networks.