ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 2, February 2013

Localized Routing and Cluster Technique in Mobile Adhoc Networks

Udaya Kumar. N
PG Scholar, Department of
Computer Science & Engineering
Adhiyamaan College of Engineering
Hosur, Tamilnadu, India

Suresh.S
Associate Professor, Department of
Computer Science & Engineering
Adhiyamaan College of Engineering
Hosur, Tamilnadu, India

***Abstract*—This paper proposes a new multipath reliable routing protocol which has the fast-recovery capability of link failures on MANETs. The provision of QoS guarantees is much more challenging in MANETs than wired networks due to node mobility, limited power supply and a lack of centralized control. Many researchers have been done so as to provide QoS assurances by designing various MANET protocols. The proposed work will provide the finest path in MANET in the way of clustering the network based on energy consumption. This energy conserved clustering on QoS for MANET will progress the life span of both the nodes and the network. The simulation results show that our protocol is better in terms of end-to-end packet reception rate, end-to-end delay, and path recovery time.**

***Keywords*—** Routing, Mobile Adhoc Networks, Cluster, AODV.

1. INTRODUCTION

The concept of QoS routing has emerged from the fact that routers direct traffic from source to destination, depending on data types, network constraints and requirements to achieve network performance efficiency.

MANET topologies can change often and unpredictably. Most protocols for multihop MANET routing maintain best effort routes. High churn or node mobility can cause QoS requirements to become unachievable. Excessive node mobility can lead to topology changes before network updates can propagate [3].

This paper addresses Routing, stability, recoverability, Utilization of maximum bandwidth and avoid delay are the main problems in routing QoS traffic in mobile networks.

The first issue is stability. With most ad hoc wireless networks that support QoS, each node acts as a router. In many distributed reactive routing schemes, if a node does not know the QoS parameters of its neighbors it broadcasts the route request packet and the neighboring nodes share their QoS parameters using broadcast packets. The broadcast packets used to discover the QoS parameters of nodes neighbors and negotiate QoS paths can flood the network. A clustered approach can lower this communication overhead to more scalable levels by limiting intercluster control communication to gateway nodes.

The second issue is recoverability can be done by using AODV protocol. In on-demand routing protocols, such as Ad-hoc On-Demand Distance-Vector routing (AODV), the route is discovered when source node desires data transmission to the destination. Eliminating periodic update and obtaining a route only when needed, AODV is the recent trend in ad-hoc networks. QoS-aware routing protocol, which is based on residual bandwidth estimation during route set up. QoS-aware routing protocol is built off AODV, in which the routing table is used to forward packets, "Hello" messages are used to detect broken routes and "Error" messages are used to inform upstream hosts about a broken route.

Localized QoS routing [2] is proposed to achieve QoS guarantees and overcome the problem of using global network state information. Using such an approach, the source node makes its own routing decisions based on the information collected by monitoring the traffic generated from it. In localized QoS routing each source node is required to first determine a set of candidate paths to each possible destination.

This paper presents the key features, definitions, and assumptions of the Cluster Formation [1], Routing protocol, which is based on-demand basis, it recovers from failures. Create a route based on clustered approach to avoid communication overhead.

RELATED WORK

2.1 Quality-of-Service Routing

Many QoS routing algorithms have been proposed. Local Proportional Sticky Routing (PSR) was the first localized QoS routing scheme [11]. PSR is simple yet stable and is used as an alternative to global QoS routing. PSR operates in two stages: proportional flow routing and computing flow proportions. Proportional flow routing determines the path of traffic during a cycle. When a cycle is complete, a new flow proportion is found for each path based on blocking probabilities.

Quality-Based Routing (QBR) determines paths based on QoS metric values. QBR monitors a path and translates flow values into average path qualities. QBR rewards successful flow and punishes flow error like CBR. The difference is that CBR assigns credits based on blocking probabilities while QBR uses average path quality.

Delay-Based QoS Routing (DBR) uses the average delay on a path to make its routing decisions. The average path delay is used to measure the path's quality, and, upon flow arrival, the path with the least average delay is used to reroute the incoming traffic. Several clustering algorithms have been proposed in MANET to choose clusterheads, namely: (I) Lowest-Degree heuristic, (II) Highest-ID heuristic and (III) Node-Weight heuristic etc.

2.2 Lowest-ID Algorithm:

In this algorithm was originally proposed by [4, 6] each node is assigned a distinct ID and the clusters are formed following the steps given below:

1. Periodically a node broadcasts the list of nodes that it can hear (including itself).
2. A node, which only hears nodes with ID higher than itself, becomes a Clusterhead (CH).
3. The lowest-ID node that a node hears is its clusterhead, unless the lowest-ID specifically gives up its role as a clusterhead.
4. A node, which can hear two or more clusterheads, is a Gateway.
5. Otherwise the node is an ordinary node.

Major drawbacks of this algorithm are its bias towards nodes with smaller ids which may lead to the battery drainage of certain nodes, and it does not attempt to balance the load uniformly across all the nodes.

2.3 Highest-Degree Algorithm:

The Highest-Degree Algorithm, also known as connectivity-based clustering algorithm, was originally proposed by Gerla and Parekh [5,7] in which the degree of a node is computed based on its distance from others.A node x is considered to be a neighbor of another node y if x lies within the transmission range of y. The node with maximum number of neighbors (i.e.,maximum degree) is chosen as a clusterhead. The neighbors of a clusterhead become members of that cluster and can no longer participate in the election process. Any two nodes in a cluster are at most two-hops away since the clusterhead is directly linked to each of its neighbors in the cluster. Basically, each node either becomes a clusterhead or remains an ordinary node (neighbor of a clusterhead).

Major drawbacks of this algorithm are the number of nodes in a cluster is increased, the throughput drops and hence a gradual degradation in the system performance is observed, and another limitation is the reaffiliation counts of nodes are high due to node movements and as a result, the highest degree node (the current clusterhead) may not be re-elected to be a clusterhead even if it loses one neighbor. All these drawbacks occur because this approach does not have any restriction on the upper bound on the number of nodes in a cluster.

2.4 Node-Weight Algorithm

Basagni et al. [8, 9] proposed two algorithms, namely distributed clustering algorithm (DCA) and distributed mobility adaptive clustering algorithm (DMAC). In this approach, each node is assigned weights (a real number above zero) based on its suitability of being a clusterhead. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring clusterhead. The smaller ID node id is chosen in case of a tie. The DCA makes an assumption that the network topology does not change during the execution of the algorithm. To verify the performance of the system, the nodes were assigned weights which varied linearly with their speeds but with negative slope. Results proved that the number of updates required is smaller than the Highest-Degree and Lowest-ID heuristics. Since node weights were varied in each simulation cycle, computing the clusterheads becomes very expensive and there are no optimizations on the system parameters such as throughput and power control. In this work, I am going to use an Efficient Clustering Algorithm which preserves the energy altitude of the network to balance both energy level and mobility rate. The proposed work will provide the finest QoS in MANET in the way of clustering the network based on energy consumption, fault tolerance rate, and mobility rate.

3. PROPOSED WORK

The proposed work is efficiently designed with the endeavor of providing fault tolerance, which is a significant characteristic in providing QoS in the connection failure-prone atmosphere of mobile networks. In order to support applications with quality of service requirements, the desired amount of bandwidth is taken to consideration and is represented in different ways. AODV is used routing QoS packets in demanding MANET environments where communication links can smash regularly and without caution. When EFDCB is flourishing, packets are distributed such that the applications reliant ahead the network are fully practical. A fault tolerance method has been proficiently attained by the EFDCB technique which has been completed by sustaining nodes in the network. A cluster-head (CH) is a node dependable for observing and informing a cluster table that accounts all QoS associations the cluster supports. At these circumstances, the packet data might get vanished. So to circumvent the CH being unsuccessful, this work presented a technique named ECA (Efficient Cluster Algorithm) is presented here shown in the figure.1.

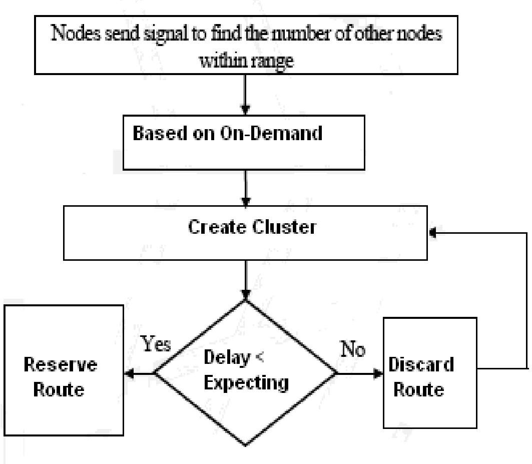

Fig. 1 Proposed Flow Diagram

3.1 Qos routing using AODV protocol

AODV is a routing protocol used in this paper. In AODV, each node maintains a routing table which is used to store destination and next hop IP addresses as well as destination sequence numbers. Each entry in the routing table has a destination address, next hop, precursor nodes list, lifetime, and distance to destination. To initiate a route discovery process a node creates a route request (RREQ) packet. The packet contains the source node's IP address as well as the destination's IP address. The RREQ contains a broadcast ID, which is incremented each time the source node initiates a RREQ. The broadcast ID and the IP address of the source node form a unique identifier for the RREQ. The source node then broadcasts the packet and waits for a reply. When an intermediate node receives a RREQ,

it checks to see if it has seen it before using the source and broadcast ID's of the packet. If it has seen the packet previously, it discards it. Otherwise it processes the RREQ packet. To process the packet the node sets up a reverse route entry for the source node in its route table which contains the ID of the neighbor through which it received the RREQ packet. In this way, the node knows how to forward a route reply packet (RREP) to the source if it receives one later. When a node receives the RREQ, it determines if indeed it is the indicated destination and, if not, if it has a route to respond to the RREQ. If either of those conditions is true, then it unicasts a route reply (RREP) message back to the source. If both conditions are false,i.e. if it does not have a route and it is not the indicated destination, it then broadcasts the packet to its neighbors. When an intermediate node receives the RREP, it sets up a forward path entry to the destination in its routing table. This entry contains the IP address of the destination, the IP address of the neighbor from which the RREP arrived, and the hop count or distance to the destination. After processing the RREP packet, the node forwards it toward the source. The node can later update its routing information if it discovers a better route. This could be used for QoS routing support to choose between routes based on different criteria such as reliability and delay.

3.2 Reliable Route Recovery-AODV (RRR-AODV)

RRR-AODV is designed to improve the performance of routing protocols (Especially AODV routing protocol). This employs local recovery of routes for reliability and reduces the number of control messages with the help of backup nodes to improve the efficiency of the network. It also considers a backup node's mobility and conducts a route recovery process implicitly. This scheme not only reduces message overhead but also improve routing performance. Various operations done in RRR-AODV were explained as follows:-

A. Establishment of Backup Node:

Backup nodes are established by overhearing the transmitted data of the sequent three nodes that are in the original route. Once the backup node is established, it makes a backup routing table and store up data in buffer. It Shows the Procedure for establishing backup nodes in figure.2.

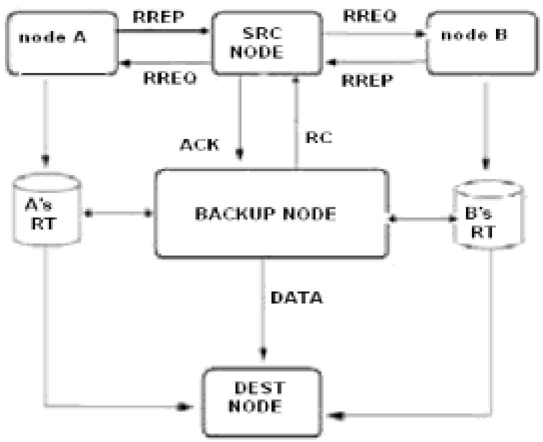

Fig. 2 establishment of backup node

B. Route Recovery Process:

If a node that is in route tries to retransmit data to failed link and a backup node detects it, the backup node begins fast route recovery. The backup nodes listen to the retransmission and then wait for a period of time selected by a random backoff algorithm. If backup nodes do not overhear acknowledgement during the back-off time, backup nodes send the Route Change (RC) packet to the node that tries retransmission. The node updates the routing table and sends the acknowledgement to backup node once it receives the RC. The backup node that receives the acknowledgement creates a new routing table and deletes the backup routing table. Next, the backup node transmits buffered data to the next node and the next node updates its routing table. Therefore the data is delivered through a backup route. When the link between nodes S and A fails, the backup node B conducts the route recovery process. After route recovery, the original route changes into the backup route in figure.3.

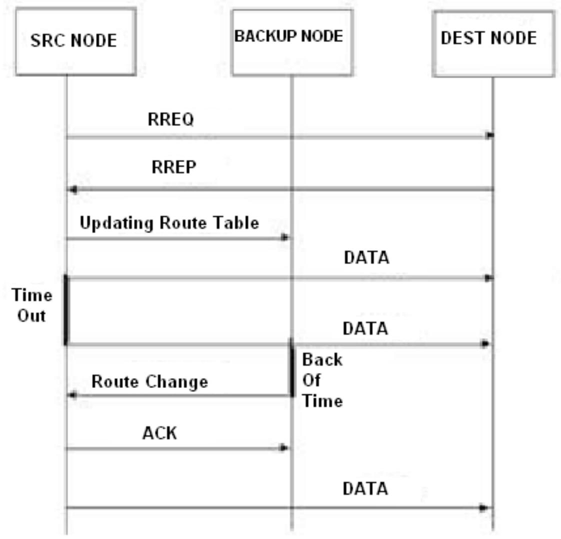

Fig. 3 route recovery process

While competition of multiple backup nodes and backup node's misjudgment problem are also solved using Random back-off time algorithm.

3.3 Cluster Formation

Each link signifies that two hosts are within the transmission range of each other. Assume every link is bidirectional so that link (i, j) exists if and only if $j \in Si$. Each host has an unique identifier (H_ID), CHs are easily identified by their identical H_ID and CH_ID values. Control information is communicated through 'Hello' messages, transmitted on the common wireless channel. Every host acquires information from incoming 'Hello' message sent by its neighbors.

A.Efficient Cluster Algorithm

Algorithm considers both location and power information to partition a MANET into separate clusters. In this context, I introduce the concept of "cluster head competence" (CHC) which represents the competence of a host to undertake the role of a CH. Each 'Hello' message includes identifications of its sender (H_ID) and sender's assigned CH (CH_ID). CHC represents a weighted sum of sender's degree (number of neighbors) and its battery power level. Finally, the 'Option' message is used for cluster size management purposes.

CHC values are calculated according to the following equation:

$$CHC = (c1 \times d + c2 \times b) - p \ldots\ldots\ldots\ldots\ldots\ldots\ldots \mathbf{(1)}$$

• $c1$, $c2$: weighted coefficients of host degree and battery availability, respectively ($0 \leq c1, c2 \leq 1$, $c1 + c2 = 1$);
•d: Number of neighbors (degree of host);
• b: Remaining battery lifetime (percentage of remaining over full battery power);
• p: 'handover' penalty coefficient.

The algorithm's execution involves the following steps:
(1) Each host sends a 'Hello' message randomly during a 'Hello' cycle. If a host has just joined the MANET, it sets CH_ID value equal to a negative number. That signifies a host is not a member of any cluster and has no knowledge of whether it is within transmission radius of another host.
(2) Each host counts how many 'Hello' messages it received during a 'Hello' period, and considers that number as its own degree (d).
(3) Each host broadcasts another 'Hello' message, setting CHC equal to the value calculated from Equation (1).
(4) Recording received 'Hello' messages during two 'Hello' cycles; each host identifies the sender with highest CHC value and thereafter considers it as its CH. In the next 'Hello' cycle, CH_ID value will be set to elected CH's ID value. In the case of two or more host having the same lowest CHC value, the one with the lowest ID is 'elected' as CH. Following the aforementioned algorithm steps, clustering procedure is completed within two 'Hello' cycles.

B.Cluster size Management

The objective of clustering algorithms is to partition the network into several clusters. Optimal cluster size is dictated by the tradeoff between spatial reuse of the channel (which drives towards small clusters) and delay minimization (which drives towards large clusters) [10]. In addition, large clusters lead to rapid exhaustion of CH battery power, while CHs represent network bottleneck points. On the other hand, small cluster sizes lead to formation of multiple clusters, implying growth of routing information and also network topology which is difficult to manage.

3.4 QoS Metric Measures

Admission Control

Admission control decision at each node ensures that the requested bandwidth and end to end delay constraints are satisfied.

1) Bandwidth Estimation: The difficulty of calculating bandwidth of a wireless channel arises

ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 2, February 2013

from the fact that it is shared by multiple nodes unlike in the wired scenario. A simple method of estimating the total available bandwidth of the wireless channel of a node is by calculating the total bandwidth consumed by all the nodes in the interference region and deducting it from the raw data rate of the node.

2) Avoid Delay: The source node computes the path that may satisfy the QoS delay requirements. It uses a setup message to travel along the selected path with each connection request. The message stores the delay over the outgoing link, and each intermediate node performs an admission test for the outgoing link, adding the outgoing link delay to the previous delay. If the delay that the message experiences is less than the QoS delay, the delay is reserved for that flow and the message is forwarded to the next node. The flow is accepted if the delay experienced in the selected path is less than the QoS delay, which means that end-to-end over that path satisfies the delay constraint.

Otherwise, if the delay over the selected path exceeds the QoS delay on any part of the path, a failure message is propagated back to the source node and the flow is rejected. This means that the delay over that path does not satisfy the delay constraint. The information regarding flow acceptance or rejection is acknowledged to the source node in order to collect flow statistics.

4. CONCLUSION

The work presented energy conserved fault-tolerant clusters with QoS routing in wireless ad-hoc networks to enhance an effective communication. Clustering is done effectively based on energy consumption. Energy conserved clustering on QoS metrics progress node's life time, network life time and minimizes path link breakdown and variation of node's changeable mobility. Since fault tolerance rate is measured in terms of number of failed links in QoS path, a local repairing of intermediary nodes has been done automatically. This scheme reliable route recovery which employs local recovery of routes for reliability and reduces the number of control messages with the help of backup nodes to improve the efficiency of the network. Mobility rate of the nodes in the QoS Paths are sustained and avoid unnecessary delay.

REFERENCES

[1] Larry C. Llewellyn "Distributed Fault-Tolerant Quality of Wireless Networks" ieee transactions on mobile computing, vol. 10, no. 2, february 2011.

[2] A.H. Mohammad and M.E. Woodward, "Localized Quality Based QoS Routing," Proc. Performance Evaluation of Computer and Telecomm. Systems (SPECTS), pp. 209-216, 2008.

[3] Chakrabarti.S and A. Mishra, Feb 2001 "QoS Issues in Ad Hoc Wireless Networks," IEEE Comm. Magazine, vol. 39, no. 2, pp. 142-148.

[4] D.J. Baker and A. Ephremides, "A distributed algorithm for organizing mobile Radio telecommunication networks", in: Proceedings of the 2nd International Conference on Distributed Computer Systems, April 1981, pp. 476–483.

[5] A.K. Parekh, "Selecting routers in ad-hoc wireless networks", in: Proceedings of The SBT/IEEE International Telecommunications Symposium.

[6] P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering In Mobile Ad Hoc Networks," in Proc. IEEE ICDCSW' 01, Apr. 2001, pp. 413–18.

[7] M. Gerla and J.T.C. Tsai, Multicluster, mobile network, Wireless Networks 1(3) (1995) 255–265.

S. Basagni, "Distributed clustering for ad-hoc networks", in: Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, June 1999, pp. 310–315.

[8] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in Multi-hop wireless networks", in: Proceedings of Vehicular Technology Conference, VTC, Vol. 2, 1999-Fall, pp. 889–893.

[9] C. R. Li, M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", IEEE Journal of Selected Areas in Communications.

[10] C.E. Perkins and E.M. Royer, "Ad-Hoc on Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA).

[11] S. Alabbad, and M. E. Woodward, "Localized Credit Based QoS Routing", Proceedings of IEE, vol. 153, PP. 787-796, 2006.

About authors

Mr. Udaya Kumar.N obtained his bachelor's degree in Computer Science from BTLIT affiliated to Vishveshwaraya Technological University, Belgaum. Currently he is pursuing his Master degree in CSE at ACE, Hosur, Tamilnadu. His research interests in Mobile Computing, Cloud Computing.

Prof. Suresh.S obtained his bachelor's degree in Computer Science from the PSG College of Technology, TamilNadu, India. Currently he is pursuing his Ph.D in Computer Science at Anna University, Chennai. He is having more than 8 years of teaching experience. He has published more than 10 papers in conferences and journals. His research interests include Server Virtualization, Cloud Computing and High Performance Computing using OPNET IT GURU.