Giving an Effective Fault Tolerance and Cluster Head in an Mobile AD-HOC Networks

SKB. RATHIKA

Abstract:**In this Distributed Fault-Tolerant Quality of Wireless Networks approach they proposed EFDCB that unifies modified GDMAC and FDCB protocols and uses CFSR for QoS routing. Here in the proposed system, proposing the weighted clustering algorithm, leads to a high degree of stability in the network and improves the load balancing in GDMAC. The load balancing is accomplished by determining a pre-defined threshold on the number of nodes that a clusterhead can cover ideally. This ensures that none of the clusterheads are overloaded at any instance of time. Moreover the stability can be accomplished by reducing the number of nodes detachment from its current cluster and connect to another existing cluster. 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. 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. The Weighted Clustering Algorithm (WCA) takes the factors into consideration and makes the selection of clusterhead and maintenance of cluster more reasonable.**

The factors are node degree, distance summation to all its neighboring nodes, mobility and remaining battery power respectively. And their corresponding weights are w1 to w4. Besides, it converts the clustering problem into an optimization problem since an objective function is formed.

Keywords: mobile ad-hoc network, quality of service, load balancing

SKB.RATHIKA., M.E, ASSISTANT PROFESSOR/CSE, SNS COLLEGE OF ENGINEERING, COIMBATORE, TAMILNADU, EMIL ID:rathika.skb@gmail.com

1. Introduction

1.1 objectives

The main objective of this work is maintaining the cluster stable . Then only the quality communication is provided in wireless network. In order to achieve stable clusters, the cluster-heads maintaining the cluster should be stable with minimum overhead of cluster re-elections. In this work propose a Weighted Clustering Algorithm (PAIWCA) which can enhance the stability of the clusters by taking battery power of the nodes into considerations for the clustering formation and electing stable cluster-heads using cluster head probability of a node. In this simulation study a comparison was conducted to measure the performance of thd algorithm with EFDCB in terms of the number of clusters formed, the connectivity of the network, dominant set updates, throughput of the overall network and packet delivery ratio. The result shows that the algorithm performs better than existing one and is also tunable to different kinds of network conditions

1.2 overview of the project work

In the algorithm election, clusterhead is adaptive invoked based on moving of nodes or changing the relative distance between the nodes and clusterhead. Election is repeated until all of node must be as a member of any cluster or as a clusterhead.

In Load-balancing, assume that there are a predefined threshold number of mobile nodes that a cluster can cover. When the number of cluster's members is too large, that may produce a small number of clusters which make bottleneck of a wireless network and reduce system throughput. Moreover, too-small cluster's member may produce a large number of clusters and thus resulting in extra number of hops for sending a packet from sthece to destination, and longer end-to-end delay. When a cluster size exceeds its predefined limit, election procedure is repeated to adjust the number of mobile nodes in that cluster.

If the distance between clusterhead and cluster member is within the transmission range, that with result a better communication. · The relative distance between nodes affects the consumption of the battery power. It is known that more power is required to communicate through a larger distance. Since clusterheads have the extra responsibility to send packets to other nodes, they consume battery power more than ordinary nodes.

Mobility is one of the most important challenges of wireless network, and it is the main factor that would change network topology. A good electing clusterhead does not move very quickly, because when the clusterhead changes fast, the nodes may be moved out of a cluster and are joined to another existing cluster and thus resulting in reducing the stability of network . In the algorithm we used Random Way Point Model.

2. Literature review

Lowest-ID clustering algorithm [5, 14] presents these steps, 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 Cluster head (CH) 3) the lowest-ID node hears is its cluster head, unless the lowest-ID specifically gives up its role as cluster head. 4) A node, which can hear two or more cluster heads, is a gateway.5) Otherwise the node is an ordinary node.

The Highest-Degree Algorithm [15, 16], also known as connectivity-based clustering algorithm, 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 cluster head. The neighbors of a cluster head 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 cluster head is directly linked to each of its neighbors in the cluster. Basically, each node either becomes a cluster head or remains an ordinary node (neighbor of a cluster head).

Distributed Mobility Adaptive Clustering (DMAC) algorithm [6, 16] is a distributed algorithm in which cluster heads are selected using a weight-based criterion that depends on node mobility-related parameters. This algorithm is suited to manage highly mobile networks. DMAC overcomes a major braw back found in most clustering algorithms. A common assumption that is presented in most algorithms is that during the set up time nodes do not move while they are being grouped into clusters. Normally, clustering algorithms partition the network into clusters and only after this step has been accomplished, the non mobility assumption is released. Afterwards, the algorithm tries to maintain the cluster topology as nodes move. In real ad hoc situations this assumption can not be made due to the constant mobility of nodes. Therefore one important feature of DMAC is that nodes can move, even during the clustering set up.

A weight based distributed clustering algorithm (WCA) [4, 9] which can dynamically adapt itself with the ever changing topology of ad hoc networks. In this approach it restricts the number of nodes to be catered by a cluster head so that it does not degrade the MAC functioning. It has also the flexibility of assigning different weights and takes into account a combined effect of the ideal degree, transmission power, and mobility and battery power of the nodes. The algorithm is executed only when there is a demand, i.e., when a node is no longer able to attach itself to any of the existing cluster heads. Clustering algorithm tries to distribute the load as much as possible. It was observed that there is a pattern of how the LBF (load balance factor) changes to distribute the load. There is a gradual increase in the LBF due to the diffusion of the nodes among the clusters. The sharp decrease is due to the imbalance caused by the clustering algorithm to ensure that the nodes are connected, which helps in routing messages from any node to any other node. Hence, there is trade-off between the load handled by the cluster heads and the connectivity of the network. Simulation experiments to measure the performance of clustering algorithm was conducted and demonstrated that it performs significantly better than both of the Highest-Degree and the Lowest-ID heuristics. In particular, the number of reaffiliations for WCA is about 50% of that obtained from the Lowest-ID heuristic. Though the approach performs marginally better than the Node-Weight heuristic, it considers more realistic system parameters and has the flexibility of adjusting the weighing factors.

A novel weight based adaptive clustering [7] approach that can be applied in MANETs to improve upon their stability. In the approach is adaptive in nature as the algorithm WBACA adapts itself to the changing topology of the network. A number of parameters of a node such as transmission power, transmission rate, mobility, degree, and battery power are taken into consideration for assigning weight to a node, This algorithm finds the local minima of weights for duster head selection and also assures that no two cluster heads be one-hop neighbors. The performance evaluation of the proposed WBACA ‘demonstrated that it outperforms the Lowest-ID and WCA algorithms and improves the stability of the clustered topology by reducing significantly on the number of clusters formed and the number of reaffiliations under different scenarios, This algorithm has also proved to be faster than the WCA algorithm in starting up the clustering process.

Ad hoc networks consist of a set of identical nodes that move freely and independently and communicate with other node via wireless links [3]. Such networks may be logically represented as a set of clusters by grouping together nodes that are in close proximity with one another. Cluster heads form a virtual backbone and may be used to route packets for nodes in their cluster. Nodes are assumed to have non-deterministic mobility pattern. Clusters are formed by diffusing node identities along the wireless links. Different heuristics employ different policies to elect cluster heads. Several of these policies are biased in favor of some nodes. As a result, these nodes shoulder greater responsibility and may deplete their energy faster, causing them to drop out of the network. Therefore, there is a need for load-balancing among cluster heads to allow all nodes the opportunity to serve as a cluster head. Then propose a load balancing heuristic to extend the life of a cluster head to the maximum budget before allowing the cluster head to retire and give way to another node.

3. Existing System

Yuan proposes two heuristics for multiconstrained QoS routing based on the extended Bellman-Ford algorithm (EBFA). EBFA computes a feasible path given multiple constraints, but its runtime/memory can be exponential. The motivation of both heuristics is to limit the number of optimal QoS paths maintained in each node. The time complexity of the limited granularity heuristic is O(|N|k|E|) for N nodes, E edges, and k QoS constraints. The second heuristic, limited path, limits QoS paths in each node for scalability. O(|N|2lg(|N|)) QoS paths are stored per node, which is optimal. The probability of finding a QoS path that satisfies constraints is high. Both heuristics employ sthece routing and scale poorly.

Nargunam and Sebastian's fully distributed cluster-based (FDCB) algorithm addresses QoS routing in MANETs. With FDCB, scalability issues in centralized routing are circumvented. The FDCB method is similar to hierarchical routing in that each cluster node only maintains QoS information for other cluster members, a fraction of the network. Thus, an increase in nodes should not significantly increase memory or runtime. Further, since global network state is shared and maintained by all, the communication overhead is greatly reduced. In FDCB, if a flow's sthece and destination are not in the same cluster, the sthece sends a route request packet to the gateway node, which forwards it to adjacent cluster(s). As long as the intermediate gateway nodes and links can support the requested QoS constraints, this process is repeated until the destination is found. The discovered path is sent back to the sthece and the resthece reservation made. The distributed nature of FDCB allows it to avoid unmanageable shared global state. FDCB's distributed routing adds initial latency for the route discovery. Route requests may not flood the network due to its clustered architecture, but precautions are needed to ensure route queries propagate efficiently from sthece to destination.

Extended fully distributed cluster-based (EFDCB) routing protocol, which is a fault-tolerant extension to FDCB. EFDCB extends FDCB to provide the scalability, efficiency, and fault tolerance critical to maintain QoS connections in a mobile environment. The goal is to determine if EFDCB provides efficient QoS route recovery by testing it against FDCB. EFDCB algorithm only has to consider a fraction of the total number of network links when finding a new feasible path through local recovery in the cluster. Hence, the burden of negotiating newly calculated QoS paths, as is done in rerouting by FDCB, is significantly reduced. For this reason, the new local method is expected to have a considerable runtime advantage resulting in improved QoS route recovery time. Faster QoS recovery time equates to lower QoS disruption time, fewer dropped packets, and improved throughput.

Larger cluster sizes typically yield high communication overhead and lack diversity from other networks with respect to changing channel conditions. Gupta et al. showed that larger cluster sizes can cause exponentially higher overhead. Some time one cluster head is overloaded while other cluster head is having only few nodes.

4. Proposed System

Thus the enhancing EFDCB by using weighted clustering algorithm (EWCA). Thus can show that this enhancement depends on two factors, improving the load balancing and performing the stability in the network. The load balancing is accomplished by determining a pre-defined threshold on the number of nodes that a clusterhead can cover ideally. This ensures that none of the clusterheads are overloaded at any instance of time. Moreover the stability can be accomplished by reducing the number of nodes detachment from its current cluster and connect to another existing cluster. Finally, the simulations results show that the proposed enhancement provides better performance in terms of stability of the created clustered topology, load balancing and number of clusterhead change. The algorithm is executed only when there is a demand, i.e. when a node moving and changing the relative distance between nodes and clusterhead. A number of parameters of nodes were taken into consideration for assigning weight to a node. In the algorithm, thus considered load balancing and stability factors. Thus assumed a predefined threshold for the number of nodes to be created by a clusterhead, so that it does not degrade the MAC function and to improve the load balancing.

5.Methodology

Using ns2 simulation the system is configured . Each node has a simulated best-effort omnidirectional interface (for cluster maintenance) as well as a QoS supporting directional interface. At simulation start, all QoS links have the ability to support any one of the requested QoS connections, but once a QoS connection has been established the associated intermediate nodes may or may not have the bandwidth available to support additional QoS requests. The arrows indicate gateway node and potential gateway node connections. Clusters are defined at simulation initialization. The goal is to balance the gateway interconnections, the number of cluster, and the overall size and complexity in the scenario tested.

In the simulation experiments, N was varied between 30 and 300 ,and the transmission range was varied between 0 and 200. At every time unit the nodes are moved randomly according to the random waypoint model in all possible directions in 250 × 250 meters square space with velocity distributed uniformly between 0 and maximum displacement along each of the coordinates. This behavior is repeated for the duration of the simulation.Assumed a predefined threshold for each clusterhead which can handle (i.e. cluster size) at most 5 nodes (ideal degree). Due to the importance of keeping the node degree approximate to the ideal as possible and to satisfy load balancing for each cluster and due to the stability of the topology network, must select the weight w1 and w3.Distance and battery power were given low weights.

The values used for simulation were w1 = 0.45, w2 = 0.05, w3 = 0.45 and w4 = 0.05. Note that all weights are kept fixed for a given system and the sum of these weighting factors equal 1.

6. Implementation outline

6.1 Module description

6.1.1. Creation of wireless network

At cluster set up, or when a node is added to the network, its variables are initialized as follows,All nodes have a unique identifier. Two nodes can be members of the same cluster if their euclidean distance is ≤ 30 m (as in 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 beacons.. When a node does not hear from a known neighbor within a set time span, it assumes the neighbor is 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 procedures are atomic except Route_traffic(u) and the procedures executed to respond to the PATH(v.rsrcs, dst) and CTS(u) messages. When discussing the relevant features of EFDCB, it is assumed that the CFSR has converged. That is, all gateway nodes in the network have path routing table entries for all network destinations. Also, it is assumed that applications using this QoS network have soft QoS constraints and use adaptive techniques to minimize QoS disruptions. Combinatorial stability is also assumed. Further, nodes have the ability to send and receive best-effort traffic along with QoS traffic. Finally, restheces allocated for a QoS connection are deallocated after a specified period of inactivity.

6.1.2 Implementation of EFDCB

EFDCB is an inherently message-driven protocol. A few of the basic clustering procedures are largely the same as in FDCB, but are included for completeness. Most procedures are new. As stated earlier, it represents a modification or addition FDCB. Given the focus of this research, the intent is not to implement all of EFDCB, but only the features required to determine if the EFDCB QoS routing protocol provides efficient route recovery. Hence, the clustering portion of EFDCB is "bootstrapped," meaning the MANET is already clustered when the system initializes. Since clustering is hardcoded, node i moving out of communication range of node j is simulated by forcing i to fail. CFSR is also bootstrapped in this simulated system by using a (centralized) QoS routing algorithm which employs sthece routing based on bandwidth. The algorithm models traffic requests as multi commodity flows to determine if traffic bandwidth demands can be satisfied. With this routing model, all nodes have complete network state knowledge as with CFSR; however, traffic flows can be split. Simulations are run on ns2 using custom middleware application agents and a custom routing module to emulate EFDCB. The MANET is simulated by using a method in which lower bandwidth is used for routing control and beacon packets (i.e., 54 Mbps throughout cluster) and higher bandwidth (i.e., 100-200 Mbps links using channels/power to go between adjacent nodes) for data traffic. Mobility was emulated by causing links to fail or recover. This allowed greater consistency between trials and easier comparisons between EFDCB and FDCB. The standard nodal protocol stack was used in ns2. Failures were simulated by causing nodes to go offline.

6.1.3 EFDCB QoS routing

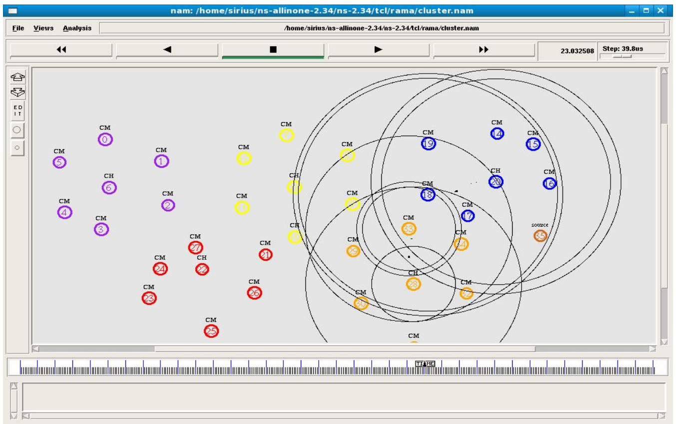
FDCB uses an on-demand reactive routing scheme, but EFDCB adopts a more proactive approach. The QoS routing scheme used by EFDCB is Clustered Fisheye State Routing (CFSR) . CFSR proposes a clustering framework to reduce redundant broadcast routing control messages. For FSR, the frequency at which node i sends its link state information to node j depends upon the distance from i to j (namely, the scope j falls in). The greater the distance, the less frequent the link state update. In CFSR, cluster-heads and gateways execute the original FSR protocol to send link state updates about the cluster, while ordinary nodes only send link state about themselves. This limits the messages from much of the network (ordinary nodes). The result is lower overhead. Assuming combinatorial stability, each node becomes aware of the complete network state with lower bandwidth. The disadvantage is that routing control messages traverse the network at a lower rate since a smaller fraction of network nodes broadcast full control messages. In CFSR's protocol, redundancy is not minimized; however, it is reduced considerably. In order to minimize redundancy, each cluster-head must not receive link state information about the same cluster from more than one gateway node. To accomplish this, the entire clustered network is partitioned into as many disjoint sets as the cluster has gateway nodes. These partitions are determined by finding the distance from each external network node to each local cluster gateway node using the topology graph stored in the routing table. The local cluster gateway node that has the shortest distance to the external network node to each local cluster gateway node using the topology graph stored in the routing table. The local cluster gateway node that has the shortest distance to the external network node. includes that external node in its control message. Due to the FSR mechanics, the gateway node closest to the external network node will be the first gateway to receive the external node's link state update. It is responsible for providing this information to the cluster

6.1.4 Enhancing EFDCB using weighted clustering algorithm

The Weighted Clustering Algorithm (WCA) was originally proposed takes fthe factors into consideration and makes the selection of clusterhead and maintenance of cluster more reasonable. As is shown in equation , the fthe factors are node degree, distance summation to all its neighboring nodes, mobility and remaining battery power respectively. And their corresponding weights are w1 to w4. Besides, it converts the clustering problem into an optimization problem since an objective function is formed.

$v\ 1\ v\ v\ v\ v\ W = w\,\mathrm{D} + w\,D + wM + w\,P$ (1)

Although WCA has proved better performance than all the previous algorithms, it lacks a drawback in knowing the weights of all the nodes before starting the clustering process and in draining the CHs rapidly. As a result, the overhead induced by WCA is very high.



7. RESULTS AND DISCUSSION:

Fully distributed cluster based algorithm provides the Quality of Service in MANET. Generalized distributed and mobility adaptive Clustering provides the Fault Tolerant in the MANET. Extended fully distributed cluster based algorithm provides a distributed fault-tolerant routing protocol With Quality of Service support in mobile ad-hoc networks. The enhancement on weighted clustering algorithm (EWCA) was explained which leads to a high degree of stability in the network and improves the load balancing and form the average number of cluster formation.

8.References

[1] Scott R. Graham, Larry C. Llewellyn, Kenneth M. Hopkinson, “Distributed Fault-Tolerant Quality of Wireless Networks” IEEE February 2011.

[2] Abdel Rahman H. Hussein, Sufian Yousef, and Omar Arabiyat Faculty of Science & Tecnology “A Load-Balancing and Weighted Clustering Algorithm in Mobile Ad-Hoc Network” Anglia Ruskin University,UK.

[3] Amis.D and R. Prakash, “Load-Balancing Clusters in Wireless Ad Hoc Networks,” in Proc. 3rd IEEE ASSET’00, Mar. 2000, pp. 25–32

[4] Chatterjee, S. K. Das, and D. Turgut,”WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks,” Journal of Cluster Computing, No. 5, 2002, pp. 193-204.

[5] Basu.P, 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.

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

[7] Dhurandher.S.K & G.V. Singh, 'Weight based adaptive clustering in wireless ad Hoc Networks", IEEE International Conference on Personal Wireless Communications, New Delhi, India, 2005, 95-100.

[8] Perkins.E, Ad Hoc Networking, Addison-Wesley, 2001.

[9] YU.Y and P. H. J.CHONG,”A Survey of Clustering Schemes for Mobile Ad hoc Networks,”IEEE Communications Surveys and Tutorials, First Quarter 2005, Vol.7, No. 1, pp. 32-48.

[10] P. Mohapatra, S. V. Krishnamurthy, AD HOC NETWORKS Technologies and Protocols, Springer Science + Business Media, 2005.

[11] R. Ramanathan and J. Redi. "A Brief Overview of Ad Hoc Networks: Challenges and Directions", IEEE Communication Magazine, 40(5), 2002.

[12] T. Ohta, S. Inoue, and Y. Kakuda, “An Adaptive Multihop ClusteringScheme for Highly Mobile Ad Hoc Networks,” in Proc. 6th ISADS’03,Apr. 2003.

[13] A.S. Nargunam and M.P. Sebastian, “Fully Distributed Cluster Based Routing Architecture for Mobile Ad Hoc Networks,” Proc.IEEE Int’l Conf. Wireless and Mobile Computing, Networking, and Comm., pp. 383-389, 2005.

[14] 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.

[15] A.K. Parekh, "Selecting routers in ad-hoc wireless networks", in: Proceedings of The SBT/IEEE International Telecommunications Symposium, August 1994.

[16] M. Gerla and J.T.C. Tsai, Multicluster, mobile, multimedia radionetwork, Wireless Networks1 (3) (1995)255–265.

First Author skb.rathika

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEEERING

SNS COLLEGE OF ENGINERRING, Email: rathika.skb@gmail.com