© INNOVATIVE RESEARCH THOUGHTS | Refereed | Peer Reviewed | Indexed
ISSN : 2454 – 308X | Volume : 07 , Issue : 02 | April - June 2021

Wireless Network Using Clustering and Fault Detection

***Pankaj Research Scholar SGI, Palwal , MDU , University**
****Ms. Sarita Bhan , Head of CSE Department , SCET, Palwal , MDU , University**

Abstract - Wireless sensor networks have been demonstrated, at an early stage in their development, to be a useful measurement technology for environmental monitoring applications. Based on their independence from existing infrastructures In this paper, the problem of determining faulty node in a wireless sensor network. Since the accuracy of data is important to the whole system's performance, detecting nodes with faulty readings is an essential issue in network management. Removing nodes with faulty readings from a system or replacing them with good ones improve the whole system's performance and at the same time prolong the lifetime of the network. In general, wireless sensor nodes may experience two types of faults that would lead to the degradation of performance.

Keywords - Manet, Faluts Diagnosis in manet, Clustering

I. INTRODUCTION

Mobile ad-hoc network is an infrastructure-less, dynamic network. Mobile ad-hoc network is a collection of wireless mobile nodes that can communicate with each other without help of any centralized authority. To provide end-to-end communication throughout the network, nodes cooperate with each other to handle network functions, such as packet routing. These networks are fully distributed and can work at any place without the help of any fixed infrastructure as access points or base stations. Figure 1 shows a simple ad-hoc network with 3 nodes. Node 1 and 3 are not within range of each other so node 2 can be used to forward packets between node 1 and 3. Node 2 will act as a router. All three nodes together form an ad-hoc network

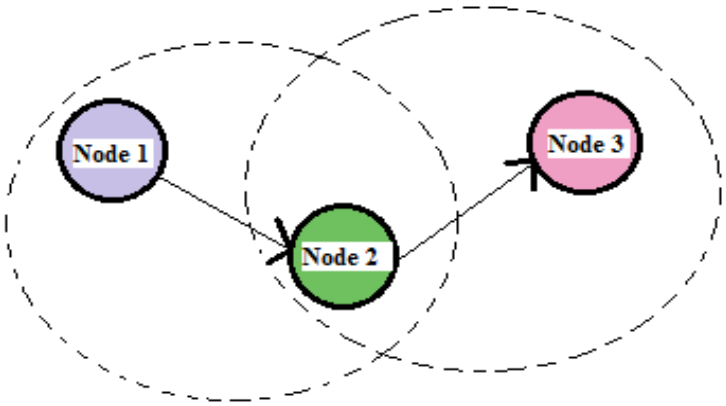

Fig 1 Example of mobile ad hoc network

II. CLUSTERING

Partitioning of network into small feasible groups of mobile nodes is called clustering. Clustering is the most popular approach that offers following advantages in MANET:

- Bettering routing and mobility.
- Supporting virtual framework for dynamic network.
- Preserving the network topology.
- Enforce more adequate resource allocation.

Clustering is used in hierarchical routing architecture to split the nodes of a self-organized network into a number of overlapping or disjoint clusters. Cluster-Based network categorize the nodes into following three types as shown in figure

- **Cluster Head:** The centre node of the cluster is known as Cluster Head (CH) that perform inter-cluster routing, data forwarding and many more operations.
- **Gateway Node:** The node that is the member of more than one cluster is known as Gateway node (GW). Thus it is used for forwarding packets between clusters i.e. in inter-cluster routing.
- **Normal or Ordinary Nodes:** The node that communicate only with their Cluster Heads and is the member of the cluster that neither act as Cluster Head nor as Gateway.

III.FAULT DIAGNOSIS IN MANET

Fault diagnosis in networks was first presented in the 1960s. In 1967, under the push of the NASA, the Office of Naval Research (ONR) took charge of the management of the Mechanical Failure Preventing Group (MFPG). With increase in complexity in spaceflight, manufacturing, navigation, nuclear industry and hospital, more faults appear. There are too many subsystems that need to assemble in large and complex system to work together well. Faults are unavoidable and often become very serious problems that we must face. Since the appearance of the computer networks, more and more application systems relied on networks to share information and acquire more effectiveness in production. The reliability of networks is an important issue. Fault diagnosis in networks prospers rapidly from 1970s, with the help of other fields. Earlier fault diagnosis in networks relies on professional knowledge and implementations.

Fault diagnosis methods for MANET must satisfy following properties:

1. Fault diagnosis should be robust to inbuilt the instability in the network.
2. Fault detection based on expert knowledge and understanding of the operations and maintenance of networks should be limited to avoid unwanted effect of overly scenarios.
3. Statistical approach should be used to capture the interactions and dependencies between measurements and states of ad hoc network elements

IV. RELATED WORK

Yang Qin at al The hierarchical routing protocols have been proposed to deal with the path search in wireless multi hop networks in various research works. Most existing designs of ad hoc network routing protocols are based on the assumption of non-adversarial environment, that every node in the network is cooperative and well behaved. However, such assumption usually does not hold in realistic environments. The performance of current routing protocols degrades significantly when misbehaving nodes exist in the network. an efficient and effective hierarchical algorithm for MANET, which is called Fault-tolerance Cluster Head based (FTCH) routing protocol. FTCH is proposed to provide a certain packet delivery fraction guarantee and low routing overhead in the presence of faulty nodes. The FTCH routing protocol is evaluated through both analysis and simulations compared with Max-Min Multi-Hop routing protocol (MMMH), AODV and DSR. The results show that FTCH greatly improves the ad hoc routing performance in the presence of misbehaving nodes.

FTCH is derived from the MMMH protocol. Hence the basic structure and entities of the FTCH is similar to the MMMH. The FTCH accounts for topology changes in the network when nodes move within a cluster (intra-cluster) or move to another cluster (inter-cluster). Intra-cluster movement causes changes in the topology to be reported to the cluster head where the link state information of the cluster is maintained. The cluster head needs to have exact link state information of the nodes of its cluster for efficient routing in the cluster.

Nigamanth Sridhar et al A failure detector is an important building block when constructing fault-tolerant distributed systems. In asynchronous distributed systems, failed processes are often indistinguishable from slow processes.

A failure detector is an oracle that can intelligently suspect processes to have failed. Different classes of failure detectors have been proposed to solve different kinds of problems. Almost all of this work is focussed on global failure detection, and moreover, in systems that do not contain mobile nodes or include dynamic topologies. Failure detectors are important building blocks for constructing fault-tolerant distributed systemsa solution to the problem of failure detection in the presence of mobility in distributed systems. Most research on failure detectors so far has been targeted at global failure detection—each process p keeps track of the health of every other process q in the system. However, in some deployment contexts, such as in wireless sensor networks, global failure detection is too resource intensive, and is hence not practical. Although there has been some research in local failure detection, all of this work ignores mobility in systems. They all assume static communication graphs.
Mourad Elhadef et al Dependable mobile ad-hoc networks are being designed to provide reliable and continuous service despite the failure of some of their components. One of the basic building blocks that has been identified for such fault tolerant systems is the failure detection service which aims at providing some information on which hosts have crashed. a failure detection service for wireless ad-hoc and sensor systems that is based on an adaptation of a gossip-style failure detection protocol and the heartbeat failure detector. We show that our failure detector is eventually perfect−That is, it satisfies both properties: strong completeness and eventual strong accuracy. Strong completeness means that there is a time after which every faulty mobile is permanently suspected by every fault-free host. While, eventual strong accuracy refers to the fact that no host will be suspected before it crashes. The failure model we consider can be described as follows. Faults are assumed to be of type crash faults, also known as hard faults. A host that crashes will not be able to communicate with its neighborhood (no sending, nor receiving). Faults are assumed to be permanent. That is, faulty mobiles will remain so until they are repaired and/or replaced. the failure detectors by two main properties: completeness and accuracy. The completeness of a failure detector refers to its capability of suspecting every faulty node permanently. While, the accuracy refers to its capability of not suspecting fault-free ones. The classical heartbeat approach suffers from two main weaknesses. The first one is that the detection time depends on the last heartbeat. This weakness may have a negative impact on the accuracy of the failure detector since premature timeouts may occur. The second weakness is that it relies on a fixed timeout delay that does not take into account the network and system's load. That is, a node may be mistakenly suspected as faulty if it slows down due to heavy workload or if the network suffers from links failure that may delay the delivery of ―I Am Alive‖ messages .

V. PROPOSED WORK

System and Fault Model

We assume that the wireless ad hoc network is a large connected network in which there are totally N sensor nodes denoted by 1, 2, 3… N. The nodes are distributed randomly in some physical domain and become stationary after deployment. The transmission range for each node is fixed and link between two hosts is bi-directional. If host u is in the transmission range of another host v, then there must be a link between the two. The system can be modeled as a communication graph G = {V, E}, where V= {1, 2….N}, and E= {(v1, v2): v1 is in transmission range of v2 and vice versa}.
A cluster is a unit disk with a radius equal to the center node's transmission range. As a result, any non-center nodes in a cluster are one-hop neighbors of the center node. The center node is called the cluster head (CH), while a node that is a one hop neighbor of the CHs of two different clusters can become the gateway (GW) node (see Figure 1) . After the autonomous cluster formation, only CH and GW node, which are elected in a fully distributed fashion, participate in the inter-cluster communication (see Figure 1(b)), while ordinary members (OMs) in each cluster talk only to their CHs (and to other members when necessary).
The proposed system is not fully distributed. The total number of nodes is equally divided into a number of clusters. Each cluster has a CH and there is a GW node between two clusters to forward the message from one cluster to another. The cluster is controlled by the CH. The fault is detected by the CH in each cluster and the

message is forwarded to all nodes of the cluster and also forwarded to other CH . All the clusters are operating simultaneously.

Intra-Cluster and Inter-Cluster Communication

In the fault detection of wireless sensor networks, we assume that all the sensor nodes have the same transmission range. Sensor nodes can be randomly deployed or placed in predetermined locations. Nodes with faulty sensors and permanent communication faults are to be identified. Sensor nodes which generate incorrect sensing data or fail in communication intermittently are treated as usable nodes, and thus are diagnosed as fault-free. Sensor nodes with malfunctioning sensors could participate in the network operation since they are still capable of routing information. Only those sensor nodes with a permanent fault in communication (including lack of power) are detected and this information is disseminated throughout the network and removed from the network.

Algorithm for cluster formation

This section describes the algorithm for cluster formation in the proposed system model. The algorithm is given in a table.

The system model uses an existing method FIND (Faulty Node Detection) to detect nodes with data faults [7]. After the nodes in a network detect a natural event, FIND ranks the nodes based on their sensing readings as well as their physical distances from the event. A node is considered faulty if there is a significant mismatch between the sensor data rank and the distance rank.

Table 1 Algorithm for Cluster formation For any unselected node v

```
{
If ((node v is an indispensable node) || (node v is the only node with highest quality Qv among
unselected neighbor) || (among unselected neighbor with same quality node v is with the
smallest ID)
{
Update status to selected;
Regard itself as a CH;
Send an invite packet, invite (v) to all neighbors ;
}
On receiving an invite packet from neighboring node v
If (node u is an indispensable node)
Discard this packet;
Else
{
Regards itself as an ordinary node;
Updates status to selected;
Sends a join packet, join (u,v) to join the cluster constructed by v;
If (more than one such packets are received)
Join the one with smallest ID;
Else
Joins sender with largest logical degree;
Regards itself as a gateway node;
}
```

On receiving a join packet sent from neighboring node u decreases the logical degree by 1;
}
}

Self-diagnosis Phase

When a set of sensor nodes is queried, each sensor in the queried set performs a self-diagnosis procedure to verify whether its current reading vector is faulty or not. Once the reading vector of a sensor node is determined as normal, the sensor node does not need to enter the neighbor-diagnosis phase. To execute a self-diagnosis, each sensor *si* only maintains two reading vectors: i) the current reading vector at the current time *t* (denoted as *bi* (*t*)); and ii) the last correct reading vector at a previous time *tp* (expressed by *bi* (*tp*)). *bi* (*tp*) records a series of readings occurred in the previous time and is used for checking whether the current reading behavior is faulty or not. If these two reading vectors are not similar, *bi* (*t*) is viewed as an unusual reading vector. Once a sensor node is detected an unusual reading vector, this sensor node will enter the neighbor-diagnosis phase to decide whether the unusual reading behavior is faulty or not. Note that when *bi* (*t*) is identified as a normal vector through the neighbor-diagnosis, *bi* (*tp*) is updated so as to react the current monitoring state.

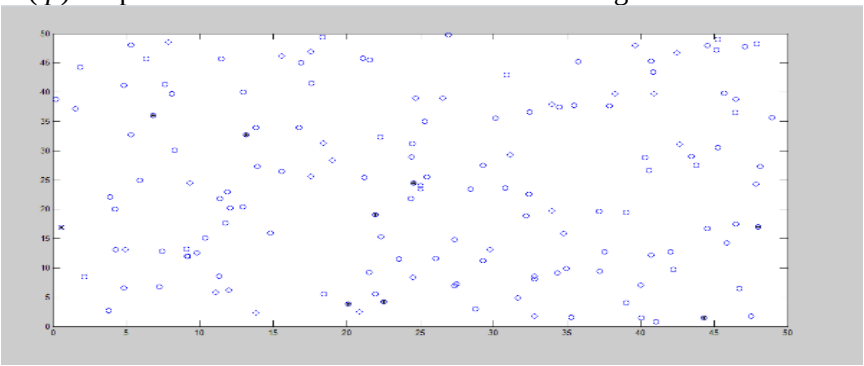

Fig 2 Fault Detection Rate

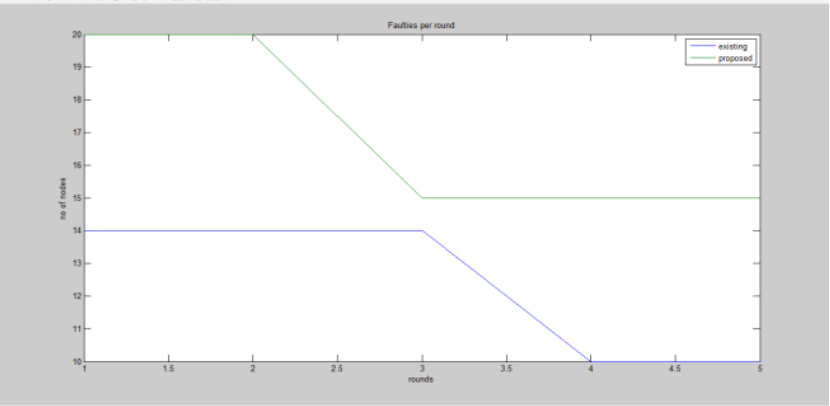

Fig 4 Faulty per round

VI. CONCLUSIONS

The failure detection algorithm coupled with suitable clustering algorithm make a very efficient failure detection service for wireless ad-hoc networks. Clustering divides whole network into two level communication architecture namely intra-cluster and inter-cluster. Two types of message overheads are required to maintain such as intra-cluster and inter-cluster. The disadvantage of the clustering approach is that CH itself may fail, hence it becomes necessary that the presence of leader is also need to be monitored and in case of its failure another node takes over the CH.

REFERENCES

[1] Elhadef, M. and Boukerche, A., ―A Failure Detection Service for Large-Scale Dependable Wireless Ad-Hoc and Sensor Networks‖, The Second international Conference on Availability Reliability, and Security, pp. 182-189, 2007.

[2] Qin, Y. and Pang, K. L. ‖ A Fault-tolerance Cluster Head Based Routing Protocol for Ad Hoc Networks.‖ Appeared in vehicular technology conference, 2008.VTC spring 2008, IEEE, pp-2472-2476.

[3] Chessa, S. and Santi,P. ―Comparison-Based System-Level Fault Diagnosis in Ad Hoc Networks‖ appeared in reliable distributed Systems,2001.procedings.20th IEEE symposium, pp.257-266, 2001.

[4] Nigamanth Sridhar ―Decentralized Local Failure Detection in Dynamic Distributed Systems‖IEEE 2006

[5] Weigang Wu, Jiannong Cao ―Eventual Clusterer: A Modular Approach to Designing Hierarchical Consensus Protocols in MANETs ―IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 6, JUNE 2009.

[6] Elhadef, M., Boukerche, A., And Elkadiki, H. ‖ Self-diagnosing wireless mesh and adhoc networks using an adaptable comparison-based approach.‖In *Proceedings of the 2ndInternational Conference on Availability, Reliability and Security*, pp.983– 990,2007.