A Failure Adapted, Load-balanced Distributed Routing for Wireless Ad-hoc Sensor Networks

Shahram Nourizadeh, Ye-Qiong Song, Jean-Pierre Thomesse

A Failure Adapted, Load-balanced Distributed Routing for Wireless Ad-hoc Sensor Networks

Shahram Nourizadeh, Y.Q. Song, J.P. Thomesse
LORIA research laboratory
National Polytechnic Institute of Lorraine (INPL)
Nancy, France
{Shahram.Nourizadeh, Song, Thomesse}@loria.fr

Abstract— this paper proposes a distributed routing protocol for ad hoc sensor networks which uses Fuzzy Logic. Each sensor uses a Fuzzy decision making process to find the best Cluster Head. Simulation shows that this protocol is able to dynamically adapt to sensors' mobility and failure. By a new load balancing method, it provides also stable clusters and so a cluster head have greater lifetime, which results minimum message exchange and so minimum energy consumption.

Keywords: Adaptive routing, Dynamic Clustering, Mobility management, Failure management, Load balancing, Fuzzy Logic

I. Introduction

In all ad-hoc wireless networks, like health monitoring systems, either the data is collected from the network periodically or on an occurrence of an event, in such systems, the data are highly vital to have a stable monitoring and have a minimum number of faulty alerts. Hence, none of them adapts completely themselves to the failure of the nodes and the temporal variations in data delivered by the sensor network. This necessitates the use of a routing protocol that readily adapts to the failures of the nodes and changes in the data delivery rate. One of the suitable solutions to manage the mobility of the nodes is dynamic clustering. However the existing clustering protocols use many assumptions which make them not able to address the needs of real application. Some algorithms are based on centralized control that makes them not to be scalable. Some algorithms use periodic rounds to change cluster head and elect a new one. The new cluster head will be fixed for one round, but in an ad-hoc network with dynamic topology, cluster topology may change during this period, and in this case a new cluster head must be elected. Therefore this type of algorithms will be good for networks with fixed or very low mobility nodes.

The zone routing protocol (ZRP) [2] is a hybrid strategy which attempts to balance the trade-off between proactive and reactive routing. In ZRP, each node maintains its own hop-count constrained routing zone; consequently, zones do not reflect a quantitative measure of stability, and the zone topology overlaps arbitrarily. LEACH [3] is an application-specific data dissemination protocol that uses clustering to prolong the network lifetime. LEACH clustering terminates in a constant number of iterations, but it does not guarantee good cluster head distribution and assumes uniform energy consumption for cluster heads. A fuzzy logic approach to cluster-head election is proposed in [4], based on three descriptors - energy, concentration and centrality. This technique is proposed to use in LEACH [3], but it cannot support the mobility of the node and in addition it is centralized algorithm and therefore it cannot be scalable.

In addition, network topology changes resulted by node mobility and node state transitions due to the use of power management or energy efficient schemes may be detected as node failures or wireless link failures. A highly dynamic network greatly increases the complexity of failure management. Also, with bandwidth limitation in a sensor network the failure detector must generate a minimum number of control messages. Marzullo[5] proposed a flexible control process program that tolerates individual sensor failures. Issues addressed include modifying specifications in order to accommodate uncertainty in sensor values and averaging sensor values in a fault-tolerant way. The authors in [6] developed an algorithm that guarantees reliable and fairly accurate output from a number of different types of sensors when at most *k* out of *n* sensors are faulty. The results of the scheme are applicable only to certain individual sensor faults and traditional networks. However, the traditional failure detectors and management systems assume that all of the nodes of the network are synonymous, that means there is no difference between a node that was crashed *n* times in *t* hours, with a node that was crashed *m* times (m>n) in the same period of time. Inadition, in the traditional failure detectors, when a node fails, it will be assumed as a dead node and we don't have a return of the node. That will be a restriction, for example, when a node is in maintenance.

Finally, because of restricted energy resources, load balancing is another important challenge in sensor networks. To balance the load in the network, most of the clustering protocols use different parameters to choose cluster-heads. Cluster ID [7], connectivity degree [8, 9] and periodical cluster heads election [3] are used in order to share the load among all the nodes of the network. By applying cluster ID or highest connectivity methods, the same node may be chose as cluster-head every time, and that will result resulting in this sensor to drain its energy very fast. Changing the cluster head in the cluster, connectivity degree or periodical choosing, changes the topology of clusters frequently and this will impose huge overhead since all other cluster-heads have to be notified about the change.

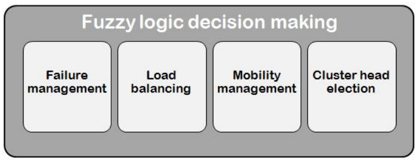

Figure 1. Protocol's architecture

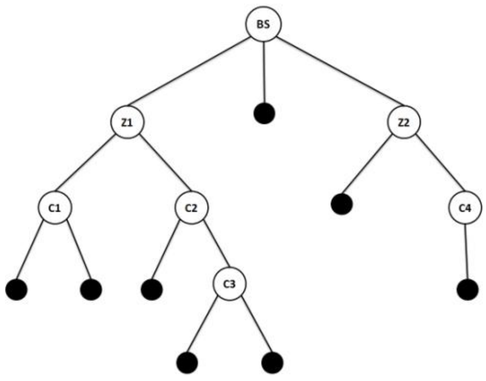

Figure 2. A sample network tree

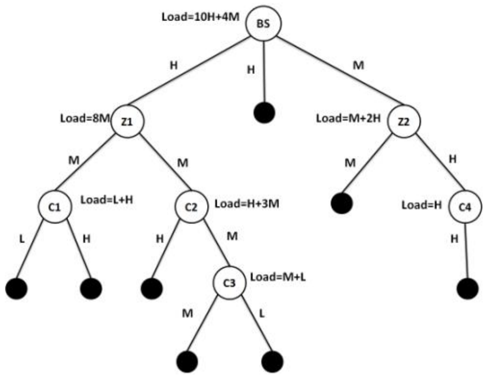

Figure 3. Load tree

The remainder of this paper is structured as follows: section II explains our proposal. The functionality of our proposal is described by using some exaples in section III, while section IV describes our simulations and section V provides concluding remarks.

II. OUR PROPOSAL

In order to address Mobility and Failure management and Load balancing, our approach has 5 main parts: Fuzzy logic decision making, Clustering (Cluster-head election), Mobility management, Load balancing and Failure management (Figure 1). To make it scalable the protocol is totally distributed, it has also a load balancing part. Fuzzy decision making is the basic part of our proposal and other parts of the protocol, use fuzzy logic to make decision or to process an event. In our proposal, we use fuzzy logic because it is capable of making real time decisions, even with incomplete information. Conventional control systems rely on an accurate representation of the environment, which generally does not exist in reality. Moreover fuzzy logic can be used for context by blending different parameters – rules combined together to produce the suitable result. In the next sections we will explain role of fuzzy decision making in different parts of our protocols.

Our protocol uses a cluster hierarchical architecture. The root node of the network tree is Base Station (BS) which can be a central computer or a receiver. In the second level, all mobile or stationary nodes that communicate directly with BS are Zone-Heads (ZH); each ZH constructs a Zone (set of one or more cluster). In the third level of the network tree, we have Cluster-Heads (CH) which is the nodes (mobile or stationary) that can communicate with one or more ZH or a node with some children that can communicate with other CHs. Finally the end level of the tree is Leaf-Nodes (LN). LN is a node without child. Figure 2 shows a sample network tree. In this figure, BS is the base station, Z1 and Z2 are zone heads, C1...C4 are cluster heads and the black nodes are the leaf nodes.

We have 7 different messages in this protocol. *Ok, invie, hello, find, join* and *join-other* are the messages used in our protocol. *Invite*t is a message, between the nodes to exchange the information. This message is used by a ZH or CH to invite the other nodes to join them. *Hello* is used by a node to announce a change or event to its neighbors. *Find* will be used by nodes to find a new parent. *Join* is used by a node to answer an Invite message. If the node has just one possible candidate to choose as its parent, it will indicate that, in this message. *Quit* is sent by a node to its parent node to advertize leaving it. Finally, *join_Other* is a message that a ZH or a CH sends to one of its child to ask him to find another parent to reduce its load by reducing number of its child. This will be when the ZH or CH, received a new request of join from a node with no other possible parent, and the admission condition is not satisfied. This message will be used also when the ZH or CH is in a low level energy state.

In this protocol we proposed a new parameter named *Mobility*. This parameter shows frequency of parent, level or zone change of a node. (Number of CH or level change of a node in his life time). Therefore each time that the node changes its CH or his level, it must increment value of a variable named *Change* and divide it to his lifetime to find the *Mobility*. It is clear that the mobility of a fix node can be greater than zero, because of the mobility of his parent.

Our protocol has also a load balancing strategy. It considers the cumulative load of data traffic from child nodes in a load tree on their parent nodes. We use *Load tree* and *admission condition* for load balancing. Figure 3 shows a sample *load tree*. The *load tree* is rooted in the base station. The load of child sensor nodes adds to the load of each upstream parent in the tree. Hence, the sensor nodes nearest the base station will be the most heavily loaded. The goal of this load balancing technique is to evenly distribute packet traffic generated by sensor nodes across the different branches of the Load tree. But here Load has a special definition. Load is the sum of the QoL (see next section) between a node and his children. It is a new definition that can be used as a new parameter in QoS. In a load tree, the weight of each link, in load tree is QoL between each node and his parent, and load of each node is the sum of the QoLs between the node and his child. In order to balance the load between nodes of the network, we use *admission condition* to accepte a new child node. The condition is, to accept a new child node, the QoL of the parent node must be greater than its *load*.

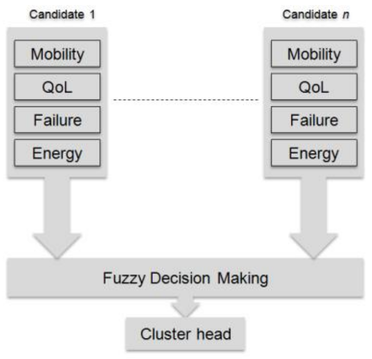

Figure 4. Cluster head election

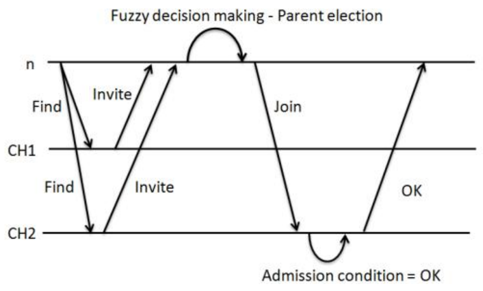

Figure 5. Example1

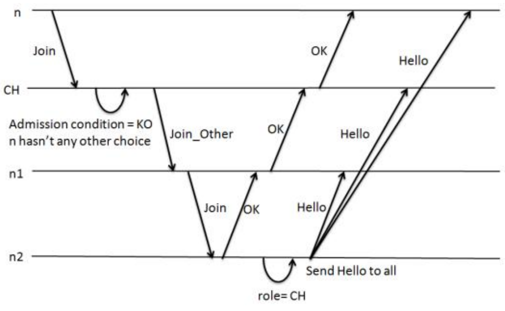

Figure 6. Example 2

In this approach we have also a parameter named *failure* shows the failure history of a sensor. This parameter will be computed by using the number of sensor's failures during its lifetime. Like the other used parameters, it is also a Fuzzy variable that has 3 levels: *High*, *Medium* and *Low*. That is not a static parameter, that means, for each node it can change from *Low* to *High* and also *High* to *Low*. To manage the failure of nodes or links, each time that the node detects a failure in a neighbor, it updates failure parameter for this node, by using: *failure = fuzzy (n / L);* where *fuzzy* is a function to convert decimal value to fuzzy value, *n* is number of neighbor's failures and *L* is our life time, therefore the *failure* parameter has different value for each node in the other nodes. In each network, due to the mobility or failure frequency of the nodes, BS will defines a update period, in which each node will update the *failure table*, therefore *failure* and *Reliability* parameters are really dynamic parameters that can change not only from *Low* to *High* but also from *High* to *Low*.

The protocol uses four parameters: *Energy level* of the node (Battery charge), *Mobility*, *Quality of Link - QoL* (*Reliability* between a node and his parent) and the *failure*, to evaluate a node that is candidate to be a ZH or CH. These parameters will be the *Fuzzy Logic Descriptors* and each of them has three possible values: *low, medium, high*. Therefore we have 81 rules to evaluate a node. The result of the rules will be *Reliability* with five possible levels: *Very Low, Low, Medium, High and Very High* (See figure 4).

In each *Invite* message the node will send necessary information to be evaluated by the other nodes, as like as: *Energy level* and *QoL*, and the node will compute the QoL of the connection between candidate and itself. The QoL of a node is *Reliability* parameter that he was calculated for his parent. This parameter helps us to choose the best parent node, a node with maximum energy, maximum stability, and higher reliability of connection. By finding the *Reliability* of a candidate we must evaluate the chance of the candidate to be a parent. To restrict depth of network's tree when a node receives more than one Advertisement, it will choose the node with smaller level, therefore we use: *Chance = Reliability / Level*

III. HOW DOES IT WORK?

In this section we will explain our protocol with some examples. Figure 5 shows thet node *n* search a parent. It sends *find* message, CH1 and CH2 receive its message and to answer the *find* message they send a *invite* message to *n*. By receiving 2 *invite* message, *n* start a *fuzzy decision making* function to find the best cluster head to join. In this example CH2 is the best one, therefore *n* sends a *join* message to it. CH2 verifies the *admission condition*, and as it is OK, it sends a *ok* message to *n* and *n* joins CH2.

In figure 6 we have a different scenario as the second example. In this example node *n* sends a *join* message to CH. The *admission condition* in CH is not OK, and *n* has just one posible cluster head to join (CH). Therefor CH sends a *Join-other* message to its children n1. By receiving this message n1 sends a*o* message to its neighbeur, n2. As n2 is a leaf node, it doesn't need to verify *admission condition*, therefore it sends a *ok* message to n1, and then changes its role to cluster head. By receiving *ok* message from n2, n1 send a *ok* message to CH and join n2. CH send a *ok* message to *n* and *n* joins CH. In this example, the role of n2 has changed, therefor it sends *hello* message to its neighbeurs to announce this change.

IV. EVALUATION

In this section evaluated performance of our proposition will be presented and will be compared with ZRP [2]. In our simulation we focus in load of the zone heads and average QoL in each zone and network's data delivery ratio as performance metrics. Table I, shows our simulation parameters.

TABLE I. SIMULATION PARAMETERS

Node Number	50
Surface	100m x 100m
Transmission range	15m
Data transmission rate	15 packet/sec
Failure model	Random
Packet size	128 bytes
Initial Energy	5J
Energy consumption (Calculation, receive and send)	10 nJ/bit

We used random waypoint model [10] in our simulations. The Network consists of several low mobility wireless nodes, just 20% of the nodes are mobile and their speed is 0.5 m/s. Each node is initially placed at a random position within in the simulation area. In adition, to focus on the assessment of the performance of the proposed algorithm, we do not generate any user data traffic during a simulation. We assume that all the nodes are able to detect correctly the failure of the other nodes and when a node failed or crashed, it is not dead, it will be return to the network after a variable time, $t\neq\infty$. In our simulation, a round is the period of time in which all the mobile nodes change their zone.

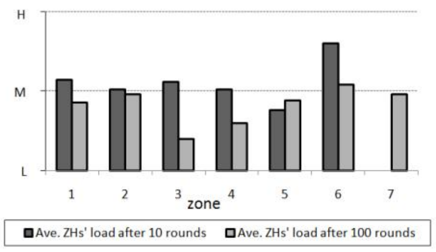

Figure 7. Network's ZHs' load after 10 and 100 rounds

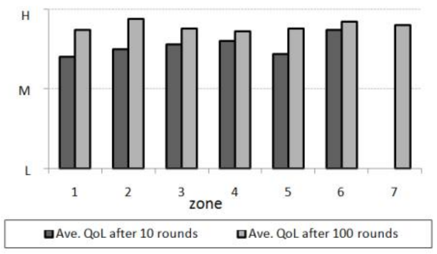

Figure 8. Average QoL in zones

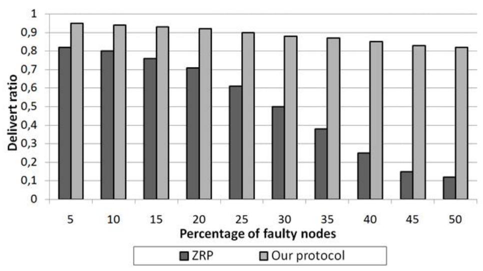

Figure 9. Delivery ratio

Figure 7 shows load in ZHs of the simulated network. We find in this figure that after 10 rounds, network has 6 zones and load of 4 ZHs are medium, one between medium and high, and one between medium and low. After 100 rounds network has 7 zones and the load of 6 ZHs is between medium and low and load of one of them is medium. The average of load in ZHs after 10 rounds is medium and after 100 rounds is between medium and low. These results show that our protocol can balance correctly the load between ZHs and CHs. A QoL with value of high shows a good connectivity between the nodes and a low QoL shows unstable connection between the nodes. Figure 8 shows average QoL in the zones of the simulated network. We find in this figure that after 10 rounds, network has 6 zones and average QoL in the zones in between medium and high, and after 100 rounds network has 7 zones with average QoL near to high. These results show the efficiency of our protocol to establish reliable and stable connections between the nodes.

In our simulation, we focus also on the network delivery ratio (the total received packets to the total sent packets in the sensor network). We compared this metrics in ZRP routing protocol and our protocol. As we said 20% of the nodes of the network are mobile and for the simulation, in each step, we change the number of faulty nodes from 5 to 50 percent. We can find the simulation results in figure 9. The simulation shows that our protocol increases the data delivery in the network and greatly adapts mobility and failure of the nodes.

V. CONCLUSION

In this paper, we have presented a distributed, load balanced routing for mobile wireless sensor networks which can adapt to mobility and failure of the nodes. This approach uses fuzzy logic to select the cluster heads. It can be applied to the design of sensor network protocols that require energy efficiency, scalability and mobility adaptation. This protocol is especially effective in networks that use sensor nodes to data aggregation and in which the data delivery ratio is important and the nodes are mobile, like health monitoring sensor networks. In such networks health events and information is sensed by several nodes and therefore, this protocol can help the network to deliver sensed events and avoid of data loss in the network. Through simulations we showed the effectiveness of our protocol for these applications.

REFERENCES

[1] C. R. Lin and M. Gerla, 'Adaptive clustering for mobile wireless networks', IEEE Journal on Selected Areas in Communications 15(7), pp 1265-1275, 1997

[2] Z. J. Haas and M. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks," IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, Oct 2002

[4] I. Gupta, D. Riordan and S. Sampalli: Cluster-head Election using Fuzzy Logic for Wireless Sensor Networks. Faculty of Computer Science, Dalhousie University. IEEE Communication Networks and Services Research Conference, 2005

[5] K. Marzullo, "Tolerating failures of continuousvalued sensors," ACM Transactions on Computer Systems, vol. 8, no.4, pp. 284-304, November 1990

[6] D.N. Jayasimha, "Fault tolerance in multi-sensor networks," IEEE Transactions on Reliability, vol. 45, no.2, pp. 308-15, June 1996

[7] D.J Baker and A. Ephremides, "A Distributed algorithm for Organizing Mobile Radio Telecommunication Networks", in the Proceedings of the 2nd International Conference in Distributed Computer Systems, April 1981

[8] M. Gerla and J.T.C Tsai, .Multicluster, mobile, multimedia radio network,. ACM/Baltzer Journal of Wireless networks, Vol. 1, No. 3, pp. 255-265, 1995

[9] A.K. Parekh, "Selecting Routers in Ad-Hoc Wireless Networks", Proceedings of the SBT/IEEE InternationalTelecommunications Symposium, August 1994

[10] Guolong Lin et al. "Mobility Models for Ad hoc Network Simulation",IEEE INFOCOM 2004