A Clustered Response Mechanism to Discover the Topology of Wireless Sensor Networks with an Application to Network Management at Faulty and Operational State of Nodes

Mahdi Nasrullah Al-Ameen
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
mahdi.cse.buet03@gmail.com

Abstract— **To efficiently manage the sensor networks the topology of the entire network has to be discovered by the monitoring node. In this paper a topology discovery algorithm for sensor networks has been described. The algorithm finds a set of distinguished nodes, using whose neighborhood information the approximate topology of the network is constructed. Only these distinguished nodes reply back to the topology discovery probes. These nodes logically organize the network in the form of clusters comprising nodes in their neighborhood. Topology discovery algorithms form a tree of clusters rooted at the monitoring node, which initiates the topology discovery process. This organization is used for efficient data dissemination and aggregation, duty cycle assignment and fault tolerance of the network system. The unpredictable behaviors of sensor networks have made it a vital point that how the operational nodes will be managed when a node in the network fails. In this paper fault tolerance mechanisms for sensor networks have been described for clustered response approach on considering different scenarios that may come to consideration when a node fails; thus ensuring maximum connectivity among operational nodes after the failure of the node. The mechanism explains how the information packets transmitted to the faulty node can be cached by an operational node. After being repaired the faulty node is reinstalled to operational state and the mechanisms of getting the repaired node connected to the network have been described in this paper. Reverse traverse mechanism has been described in this paper as a part of fault tolerance mechanisms, which ensures that the number of clusters is not increased when a faulty node is repaired and re-connected to the network. The mechanisms described in this paper are distributed and highly scalable.**

Keywords-Sensor networks; topology discovery algorithm;5-color clustered response mechanism; data dissemination and aggregation;duty cycle assignment;fault tolerance mechanisms; reverse traverse mechanism.

I. INTRODUCTION

A sensor network consists of hundreds of tiny battery-powered sensing nodes communicating via radio. The sensors are deployed throughout the sensor field to collect information or data for a base station [4]. Deploying such a high number of nodes requires careful handling of topology maintenance. In fact sensor networks are distributed event-based systems that differ from traditional communication networks in several ways: sensor networks have severe energy constraints, redundant low-rate data, and many-to-one flows [5].

The behavior of sensor networks is highly unpredictable because of randomness in individual node state and network structure. Routing and data dissemination problem in sensor networks can be viewed as a special case of mobile ad-hoc networks. In this case for sensor networks a much simpler and more scalable solution is needed.

To efficiently manage sensor networks the topology of the network has to be known to the monitoring node [3]. In this paper 5-color clustered response mechanism has been described to discover the topology of the network and this mechanism is designed in a way so that the nodes in sensor networks get and store the required information to simply and efficiently implement the fault tolerance mechanism.

The mechanism to discover the topology of the network is the basis of describing fault tolerance mechanism for the network system. At any instant a node may fail, which is a child of a node (if it is not a monitoring node) and also may be the parent of a number of nodes. Thus the fault tolerance mechanism works on updating the topology of the network in the absence of a faulty node.

Duty cycle assignment has been discussed in this paper so that it is clearly defined how a packet of information is transmitted between a pair of clusters. A set of nodes in each cluster is selected for the communication between each pair of clusters. Data dissemination and aggregation has been covered within the context of this paper. And finally the paper gives a clear indication to the future work for the management of sensor networks.

II. 5-COLOR CLUSTERED RESPONSE MECHANISM

In clustered response approach the network is divided into set of cluster where each cluster is represented by one node, called the cluster head [2]. Each node is part of at least one cluster in this case and the response action is generated only by cluster heads. The clustered response approach gives the reachability map of the network where if all cluster heads are reachable then all other nodes are reachable from at least one cluster head [1].

A sensor network is considered to be an undirected graph G {V, E} with vertices V and Edges E. In clustered response approach nodes in the graph are divided into sets of nodes. A cluster head represents each set and the nodes in its set belong to its neighborhood [1].

Consider H as a set of cluster heads, and N_i as the neighborhood list of node i, where $i \in H$. In this case following conditions hold.

1. $N = \cup N_i$.

2. $\forall x \in N_i$, edge $(x, i) \in E$.

Communication overhead for clustered response approach is dependent on the number of clusters that are formed and the path length connecting the clusters. So for minimum communication overhead following problems are needed to be solved.

- Find minimum cardinality set of cluster heads which have to reply back.
- Form a minimal tree with the set of the cluster heads.

In fact these are combinatorial optimization problems. Moreover, for optimal solution one needs global information about the network whereas the nodes only have local information. So in this paper heuristics have been given that provide approximate solutions to the problems.

In the mechanism to be discussed, 5 different colors represent 5 possible states of a node. The meaning of each color in the event of representing the state of a node has been described below.

- White: Initially all nodes are white. The node, which has not received any topology request, is white.
- Yellow: When a white node accepts a topology request its color changes to yellow. A yellow node broadcasts topology request with its location information after getting a 'broadcast trigger' message from its red parent node and then starts a timer.
- Red: When the timer of a yellow node is stopped its color changes from yellow to red. When the timer is on, the yellow node accepts acknowledgements from the nodes that have received topology request from this node.
- Black: If a node is termed as a cluster head then its color changes from red to black.
- Gray: The color of a node is changed from red to gray when it is termed that the node is not a cluster head.

Now it is time to explain how this response mechanism works in a sensor network. Topology request contains the id and location information of the node from which it is broadcasted.

The topology request is begun to be broadcasted from a root node which is also termed as monitoring node. In this discussion node: 'a' is the parent node of node: 'b', if node: 'b' receives a topology request from node: 'a'. In this case node: 'b' is the child node of node: 'a'

At the beginning, the root node (monitoring node) is assigned yellow color and then the topology request is broadcasted (with its location information). On receiving the request a white node becomes yellow and sends an acknowledgement to the parent node carrying its location information. The parent node starts a timer as soon as it broadcasts the topology request and waits a period of time for acknowledgement from the children nodes.

As the timer is stopped, the color of the parent node changes from yellow to red and then establishes a priority queue of the nodes (that have sent acknowledgements) according to their location information. The closest child node from the red parent node gets the lowest priority and the farthest child node gets the highest priority in the priority queue.

A 'broadcast-trigger' message is sent to the first child node (the farthest node) in the priority queue and as the node gets the message it broadcasts the topology request. Other nodes in the queue wait for this message for broadcasting topology request.

The color of a yellow node changes to gray if it does not get any acknowledgement (with location information) within the time period when its timer is on. It means that there's no white node within its transmission region; that is either there's no node at all within its transmission region or the existing nodes within its transmission region have already been included to another cluster. When a node becomes gray, it informs its red parent node about its updated state (denoted by the color).

When all the children nodes (which are in the priority queue of the parent node) are denoted by either gray or black color, the color of their parent node changes from red to black.

When the color of a node changes from red to black it aggregates the topology information of its children black nodes with its own topology information and sends it to its red parent node.

When a red parent node gets a message from a child node (gray) about its updated state or receives topology information from a child node (black), it sends 'broadcast trigger' message to the next child node (next farthest child node) in the priority queue to broadcast topology request.

Here only the black nodes send topology information to their parent nodes (which ultimately become black) and in this course the root node (monitoring node) gets the reachability-map of the entire network and then it becomes black.

In this mechanism, a gray node has no child thus it cannot form a cluster (so it cannot be a cluster head), rather it is part of a cluster. But a black node has at least one child thus forming a cluster (being a cluster head). The monitoring node gets the complete reachability-map of the network.

When a white node receives a topology request it responds to the parent node with its location information and stores the id and location information of the parent node. But when a non-white node (node of any other color) gets a topology request it saves the node-id and location information of the node from which the request has been broadcasted but does not respond to the transmitting node; thus it is not included to the cluster of the transmitting node but stores the id of the node to whose transmission region the receiving node belongs to. This information will be required for fault tolerance mechanisms. Each node stores its cluster-id, the id of the cluster head of its cluster and the id of the nodes, which cover this node within their communication region.

Node: 1 (monitoring node) is red as it has already received acknowledgement with location information from node: 2 and node: 10. As node: b is nearer to node: 1 than node: 2, node: 2 is the first node of the priority queue of node: 1.

A. Description on 5-Color Clustered Response Mechanism from the Perspective of a Network Model

The state of the network at a particular moment of time has been shown in Fig. 1. Here node-id is representing the order of receiving 'broadcast trigger' message (the order of broadcasting topology request).

Node: 2 broadcasts the topology request and starts a timer. Within the time period when the timer is on it receives acknowledgement (with location information) from node: 9, node: 8 and node: 3. On considering the location information at first node: 3 receives the 'broadcast trigger' message. Node: 4 and node: 7 are the children of node: 3.

Node: 4 broadcasts topology request to node: 5 and node: 6. Node: 5 and node: 6 don't get acknowledgement from any node. So both of these nodes become gray and send messages to node: 4 about their updated state. Node: 4 then become black and sends a topology response to node: 3. Node: 3 then sends 'broadcast trigger' message to node: 7.

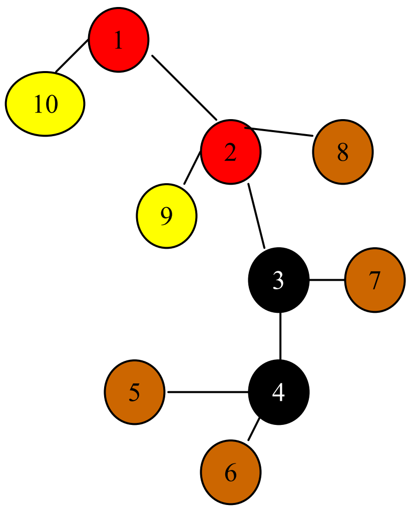

Figure 1. State of the sensor network at a particular instant during topology discovery phase

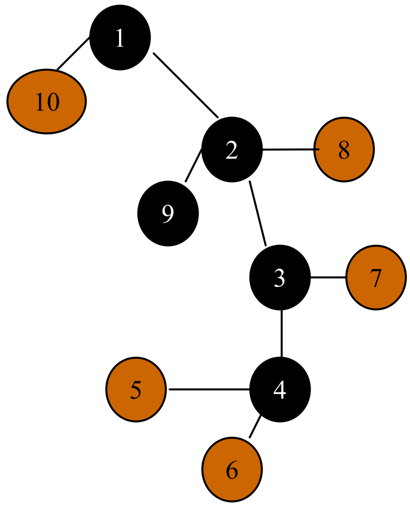

Figure 2. State of the sensor network after topology discovery phase.

On getting a message from node: 7 about its updated state (gray), node: 3 becomes black and aggregates the topology information of node: 4 with its own topology information and sends the aggregated topology response to node: 2. In this way node: 1 gets the complete reachability map of the network and it becomes black. Resulting state of the network after topology discovery phase is shown in Fig. 2.

B. Heuristics Behind 5-Color Clustered Response Mechanism

The heuristic behind giving the farthest node the highest priority in the priority queue is explained as follows: The coverage region of each node is the circular area centered at the node with radius equal to its communication range. Then the number of nodes covered by a single node could be proportional to its coverage area times the local node density [1]. The number of new nodes covered by a forwarding node is proportional to its coverage area minus the already covered area. This is illustrated in Fig. 3.

Coverage region of node: A is colored blue. New area (outside the coverage are of node: A) covered by node: B is colored light green and the new area (outside the coverage are of node: A) covered by node: C is colored green.

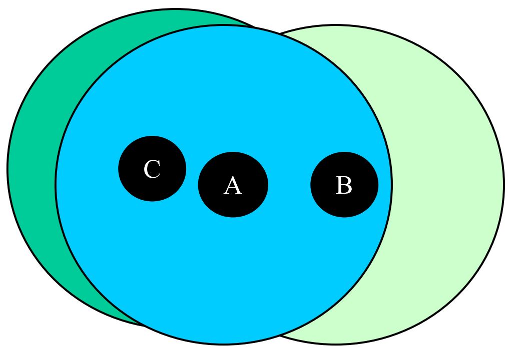

Figure 3. Illustration for distance heuristic of 5-color clustered response mechanism

In Fig. 3, the distance of node: B from node: A is greater than the distance of node: C from node: A. Thus node: B covers larger new area than the new area covered by node: C.

III. IDENTIFYING A FAULTY NODE

A watchdog timer is incorporated in every node. The timer of a node tracks an individual signal from each node that is connected to it. This signal is sent at regular interval from each node. In fact this signal is an indication that the sending node is alive and remains in operational state. But if a node fails to send the signal to the watchdog timer of a node (let us say this node: v) at the predefined frequency for a several (it is a pre-calculated fixed value) number of times, it is identified as a faulty node by node: v

When a node fails, the node that has identified the faulty node sends the 'failure message' to the monitoring node with the node-id of the faulty node and required steps are taken to repair and reinstall the faulty node to the operational state. At this point it is necessary to update the topology of the network so that the regular connectivity is maintained among operational nodes with maximum efficiency and availability. Details on fault tolerance mechanisms from the perspective of faulty and operational state of a node have been discussed in this paper.

IV. FAULT TOLERANCE MECHANISMS

In sensor networks fault tolerance mechanism is discussed depending upon different states of a node.

- Faulty State: When a node fails.
- Operational State: When the faulty node is repaired and reinstalled to operation.

The information packet transmitted to the faulty is needed to be cached by an operational node so that the rate of packet (transmitted to the faulty node) drop is decreased. When the faulty node is reinstalled to operation the cached packet is transmitted to it and it responds. Consequently the response rate of the nodes in the network system is increased.

A. Mechanism of Caching Packets Transmitted to a Faulty Node

When a node (let us say this node: i) fails, it cannot receive the information packet transmitted to it. The node (let us say this node: h) that has to forward the packet directly to node: i caches the information packet for a specific period of time. In this case, node: h is definitely a neighbor node of the node: i and thus it is well-informed about the faulty state of node: i and thus caches the information packet, to be transmitted to node: i. Node: h starts a timer when it caches the packet of the node: i. Now the packet can be transmitted to node: i from node: h in two ways.

The first one is event-driven approach. In this approach when node: i comes to operational state it informs each node within its communication region by sending a message and if node: h receives this message within the period of time when its timer remains on, the timer is interrupted and stopped and it sends the cached packet to node: i. In this approach node: h need not cache the packet any more as soon as the faulty node is reinstalled to operation.

In time-driven approach node: h transmits the cached packet to node: i when its timer is elapsed. If node: i is reinstalled to operational state within the period of time when the timer of node: h remains on, node: i receives the information packet, otherwise the information packet is dropped. In this approach node: i need not transmit any message to node: h to acknowledge that it has entered the operational state.

B. Discussion on Fault Tolerance Mechanisms at Faulty State of a Node

A node that has failed either may be a gray node or a black node. Different fault tolerance mechanisms are implemented depending on the color (representing node-status) of the node.

If the faulty node is a gray node it is detached from its parent node. If the faulty node is the only child of its parent then the color of its parent is changed to gray and the updated topology information is passed to the upper level of black parent nodes. But if the parent of the faulty node has at least one more child except the faulty node, then the parent updates its topology information and transmits to its upper level.

In the case of the failure of a black node its children become orphan and are re-connected to the network system through the nodes within their communication region. When a node identifies itself as an orphan, its color changes to blue but its previous state (black/gray) is also stored. If its previous state was black then it broadcasts a 'faulty state' message to the nodes of the tree rooted at itself and on getting this message the color of a node is changed to blue. In fact the color: 'blue' represents that the corresponding node is part of a tree rooted at the faulty node and is under consideration of the fault tolerance mechanism where it may be needed to update the topology of this tree.

The child (let's say this node: a) of the faulty black node sends a 'parent find' message to each node within its communication region. Node: a starts a timer after sending the message. Nodes that are blue ignore the message so as to minimize the complexity of the mechanism and as a result the rate of message-transmission is also minimized in this case. A non-blue node replies to the message with its current state (black/gray).

On the basis of replies to the 'parent find' message one of the following 3- scenarios may come to consideration:

- Scenario 1: Node: a has received reply from at least one black node.
- Scenario 2: Node: a has received reply from no black node but from at least one gray node.
- Scenario 3: Node: a has received reply from no node.

Fault tolerance mechanisms have been described from the perspective of the mentioned scenarios.

1) Scenario 1: In this case node: a receives reply (to its 'parent find' message) from at least one black node. If multiple black nodes reply, node: a selects the black node (Let us say this node: s) with minimum distance (location information of each node within its communication region is stored during topology discovery phase) after its timer is elapsed and sends a 'parent selection' message to node: s. It then receives a confirmation message from node: s and thus gets included to its cluster. Topology of node: s is updated and passed to its upper level.

The color of node: a is changed from blue to its previous color (gray/black) when it is connected to node: s. Now if the color of node: a is black the 'parent found' message is sent to the nodes of the tree rooted at node: a. On getting the 'parent found' message the color of a node is changed from blue to its previous color.

2) *Scenario 2:* This scenario comes to consideration when node: a receives reply from no black node but from at least one gray node. One of the two approaches can be taken in this case to ensure connectivity among all operational nodes.

a) *Approach 1:* According to the first approach node: a is connected to the gray node (let us say this node: t) with minimum distance from it and thus the color of node: t changes to black and its topology information is transmitted to the upper level of black parent nodes. The color of node: a is changed to its previous color when it gets connected to node: t. If the color of node: a is black the 'parent found' message is sent to the nodes of the tree rooted at node: a and on getting this message the color of a node is changed from blue to its previous color.

The number of clusters is increased in this approach but it is simple and minimum message-transmission is required to update the topology of the network under the consideration of this scenario.

b) *Approach 2:* In the second approach, fault tolerance mechanism is quite different and complicated but it tries to ensure 'not to increase' the number of clusters. This approach is applicable only when node: a is a black node. In this approach, node: a broadcasts the 'search parent' message to the nodes of the tree rooted at node: a and starts a timer. This message is sent to every node of this tree. On getting the message each node broadcasts the 'parent find' message and starts a timer. As the timer elapses, if the node (let us say this node: c) receives reply (to the 'parent find' message) from at least one black node it sends the 'black parent discovered' message to node: a. In case of multiple black nodes that have replied, black node with minimum distance is selected by node: c. On getting the 'black parent discovered' message from any node (in this case it is node: c) the timer of the node: a is interrupted and immediately stopped. Then node: a sends an acknowledgement to node: c and on getting the acknowledgement node: c sends a 'parent selection' message to the selected black node (let us say this node: e) and gets connected to its cluster. Node: e sends the updated topology information to its upper level of black parent nodes.

But if the timer of node: a elapses, it means no black node has replied to its 'parent find' message within this period when the timer remains on. In this case node: a selects the nearest gray node (let us say this node: m) as its parent that has replied to its 'parent find' message and sends a 'parent selection' message to node: m. The color of node: m then changes from gray to black and sends a confirmation message to node: a. Node: a is now connected to node: m which sends its updated topology to the upper level.

If node: c gets connected to a black node (node: e), the tree rooted at node: a (let us say this tree: T1) is now connected to the rest of the network through node: e. So the topology of the nodes of this tree has to be updated. In this case, topology update mechanism is described as follows.

5-color clustered response mechanism can be applied in this case where node: e is the monitoring node that begins broadcasting the special topology discovery request (the request message is different from the usual topology discovery message that is broadcasted to discover the topology of the entire network) Before broadcasting the topology discovery request by node: e, the color of node: a is changed from blue to violet and then it broadcasts a 'change state' message to each node of T1 and on getting this message the color of a node changes from blue to violet. The non-violet node ignores this special topology discovery request. Violet nodes are considered as white nodes in this case of applying 5-color clustered response mechanism.

Our goal is 'not to increase' the number of clusters, which become uncertain on applying this mechanism. To resolve the problem reverse traverse mechanism has been described as an expected solution of not increasing the number of clusters when the topology of the network is needed to be updated after the failure of a node.

According to the reverse traverse mechanism:

- If the previous color of node: c was gray its color is changed from blue to black and it sends a 'topology change' message to its current parent.
- On getting this message the color of any node changes to gray if it has no other node connected to it except the node that has sent the message. But the color of a node changes to black if it has at least one other node connected to it.
- If a node receives the message from its child, parent-child relationship is swapped between the two nodes and as a result the receiving node becomes the child of the transmitting node. But parent-child relationship remains unchanged when a child node receives the message from its parent.
- Each node forwards the message to the nodes, connected to it after its color is changed to gray/black and parent-child relationship is defined with the node from which it has received the 'topology change' message.

In Fig. 4 the tree (rooted at node: 3b) is connected to the rest of the network (not shown in the figure) through node: A. Here node: 1 is representing node: c (as discussed before) and node: A represents node: e. Arrowhead represents the direction of the transmission of 'topology change' message.

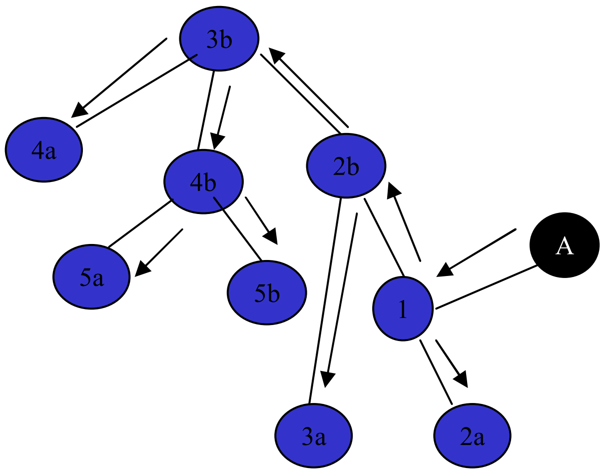

Figure 4. Illustration before applying reverse traverse mechanism

In Fig.4, node-id stands for the sequence of receiving this message. Here node: 1 sends the message simultaneously to node: 2a and node: 2b and any of these two nodes can receive the message before another one. After applying the reverse traverse mechanism the updated topology of the tree in Fig. 4 has been represented by Fig. 5.

In this example the number of clusters remains unchanged. In fact the increase/decrease of the number of clusters can be defined by 4 conditions. Conditions are described considering Fig. 4.

- Number of clusters decreases if the previous color of node: 1 was black and node: 3b has no other child except the one (here it is node: 2b) from which it has received the 'topology change' message.
- Number of clusters remains unchanged if the previous color of node: 1 was black and node: 3b has at least one child except node: 2b.
- Number of clusters remains unchanged if the previous color of node: 1 was gray and node: 3b has no child except node: 2b.

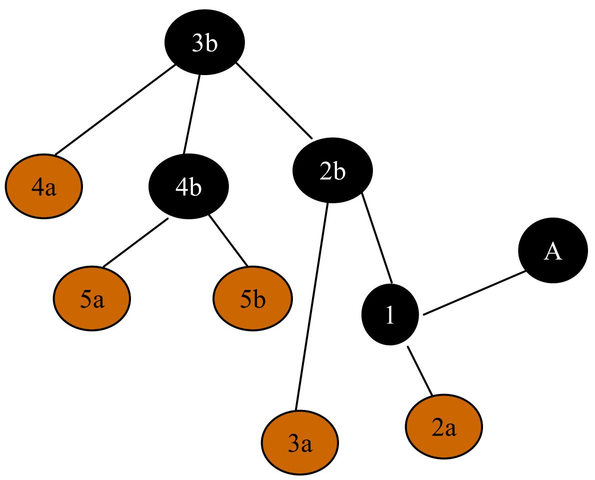

Figure 5. Illustration after applying reverse traverse mechanism

- Number of clusters is increased if the previous color of node: 1 was gray and node: 3b has at least one child except node: 2b.

Here under 3 conditions (out of 4) the number of clusters is not increased and under 1 condition the number of clusters is decreased. So a higher probability exists of not increasing the number of clusters on applying the reverse traverse mechanism

3) Scenario 3: In this scenario node: a receives reply (to its 'parent find' message) from no node. So if node: a has no child, it is just disconnected from the rest of the network. But if it has child it broadcasts 'search parent' message to the nodes of the tree rooted at its own and starts a timer as described before in approach 2 of scenario 2.

The method of processing 'black parent discovered' message has been already described. But if the timer of node: a elapses, it means no black node has replied to the 'parent find' message within this period when the timer remains on. In fact if a node (let us say this node: n) finds no black node replying to the 'parent find' message, it selects the nearest gray node that has replied and sends the 'gray parent discovered' message to node: a with the distance information (the distance is between node: n and the selected gray node). Node: a stores this information (only when node: a receives reply to its 'parent find' message from no node) and when the timer is elapsed it selects the gray node with minimum distance-information. Consider, this gray node (let us say this node: p1) is selected by node: q as its nearest gray node. Node: a then sends an acknowledgement to node: q and on getting the acknowledgement node: q sends a 'parent selection' message to node: p1 whose color is then changed to black. It then sends a confirmation message to node: q that node: q has been added to its cluster and sends the updated topology information to its upper level.

In this case the node is changed through which the tree: T1 is connected to the rest of the network. So the topology of the nodes of this tree can be updated by the reverse traverse mechanism that has been already described.

If node: a receives no 'black parent discovered' message or 'gray parent discovered' message within the period of time the timer of node: a remains on, the tree: T1 is disconnected from the rest of the network.

4) Discussion on Fault Tolerance Mechanism at the Faulty State of a Node from the Perspective of a Network Model: A network model has been illustrated in Fig. 6 to explain the fault tolerance mechanisms under the consideration of different scenarios that have been already described. A specific state of the network at a particular instant of time is illustrated in this figure.

In Fig. 6 node: F has become faulty and before it was faulty node: s1, s2a1 s2a2, s3a and s3b were connected to it as children nodes but now these nodes have become orphan. The color of each orphan node has been changed to blue. Node: n1, n2, n3, n4, n5 and n6 are connected to the rest of the network (not shown in this figure).

Here node: s1 gets reply (to its 'parent find' message) from two black nodes and will be connected to node: n1 (the nearest black node) following scenario 1.

Node: s2a1 gets reply (to its 'parent find' message) from no black node but from gray nodes and n5 is the nearest gray node in this case to which node: s2a1 will be connected if approach 1 is followed in this scenario 2. If approach 2 is followed then node: s2a1 will be connected to the same node (node: n5) because no black node has replied to the 'parent find' message of its children. Color of n5 will be changed to black in this case. But in case of node: s2a2, a black node (node: n4) has replied to the 'parent find' message of node: 3 and according to approach 2 of this scenario 2 the tree rooted at node: s2a2 will be connected to the rest of the network through node: n4. Reverse traverse mechanism is used to update the topology of this tree.

Following scenario 3 node: s3a is disconnected from the network because no node has replied to the 'parent find' message of this node and of its children. Node: s3b gets reply (to its 'parent find' message) from no node and its child node: 9 gets reply to its 'parent find' message from no black node but from gray nodes and node: n6 is the nearest gray node in this case through which the tree rooted at node: s3b will be connected to the rest of the network. The topology of this tree can be updated by reverse traverse mechanism.

When the faulty node enters the operational mode it is re-connected to the network through different mechanisms; discussed in the next section.

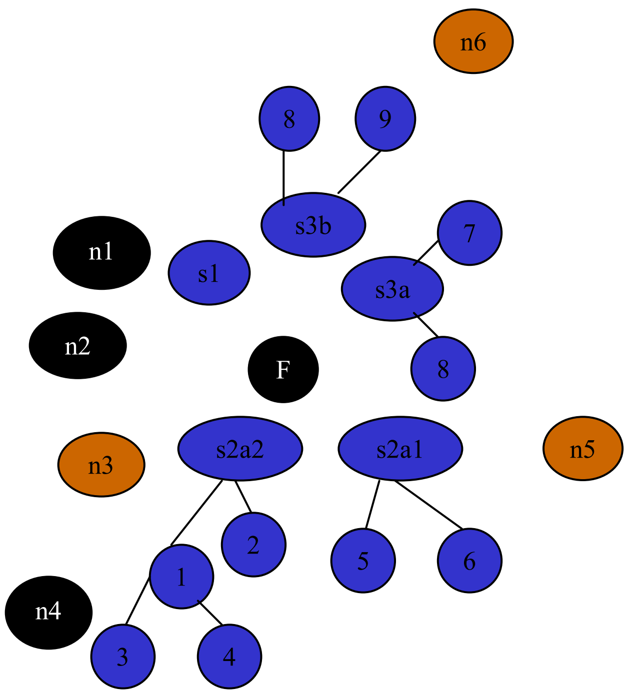

Figure 6. Network Model illustrating fault tolerance mechanism

C. Discussion on Fault Tolerance Mechanism at the Operatial State of the Faulty Node

When a faulty node (let us say this node: w) is repaired and reinstalled to operation it has to be connected to the network. The topology of the network may have been changed after the failure of node: w. When node: w is reinstalled to operation it sends 'get state' message to the nodes within its communication region. On getting the message a node replies with its current state (gray/black). At this point different approaches can be followed for node: w to be connected to the network. In this case if node: w selects a node (let us say this node: p7) as its parent it sends a 'parent selection after repair' message to node: p7. If the color of node: p7 is gray it is changed to black. Node: p7 then sends a confirmation message to node: w and the updated topology information is sent to the upper level of black parent nodes by node: p7.

- Approach 1. If the current state of the previous (before node: w was faulty) parent node (let us say this node: p3) of node: w is black, node: w is connected to it as a gray node. The number of clusters remains unchanged in this approach.
- Approach 2. Node: w selects the black node with minimum distance as its parent and sends it the 'parent selection after repair' message to get connected. In this case number of clusters is not increased
- Approach 3. In this approach node: w is always connected to node: p3 even if the color of node: p3 is gray. The number of clusters is increased in this case if the color node: p3 is gray.

Method of resetting the states of the nodes with the updated states is as follows.

If approach 1 or approach 3 is applied after node: w is repaired, it is possible to get the nodes to their previous (before node: w became faulty) states (gray/black). When the topology of the network is updated after a node becomes faulty, each node whose topology is updated, stores its previous state. When node: w is connected to node: p3, node: w broadcasts 'reset state' message and on getting the message each node gets back to its previous state and forwards the message to other nodes.

D. Fault Tolerance Mechanisms When a Node Fails Multiple Times

Each node stores its updated topology information after a node fails (let us say this f2) and if node: f2 fails for the second time, 'restore state' message is broadcasted (with the node-id of the node: f2) by the node that has identified node: f2 as a faulty node and on getting this message a node restores its state to that state that has been stored when node: f2 became faulty for the first time.

A considerable amount of message transmission is required for the fault tolerance mechanisms when a node becomes faulty for the first time. But it can be reduced when a node fails multiple times by following this mechanism.

V. DATA DISSEMINATION AND AGGREGATION

In sensor networks all information flow would be from sensor to monitoring node with some control information being transmitted from monitoring node to sensors. The topology discovery process sets up a treC rooted at the initiating node. Thus any data flow from a sensor to monitoring node has to flow up the TreC [1].

Each cluster has a minimal number of nodes, which are active to transfer packets between a parent-child cluster pair. Mechanism of assigning duty cycles of active nodes is described later. Whenever a sensor needs to send some data to the monitoring node, it can just wake up and broadcast. The duty cycle assignment mechanism ensures that there is at least one node, which is active and responsible for forwarding the data to the next cluster [1]. Also there is at least one node in the next cluster active to receive this packet.

Each black node covers a region given by its communication range. The parent black node, logically also covers the area covered by its children black nodes. Thus the area covered propagates up the tree and monitor covers the whole field. The area covered by each black node is cached during the topology response phase. The parent black node gets such areas from its children and in turn makes larger area to approximate its logical coverage region [1]. Region based queries from the monitoring node can be channeled to the appropriate region by the black nodes using their coverage information. At the return path the data may be aggregated at the black nodes.

VI. DUTY CYCLE ASSIGNMENT

Each black node has the location information of its children black nodes and children gray nodes. Each parent black node selects sets of children gray nodes to communicate with the cluster of its children black nodes. A set of children gray nodes is selected for each of the children black node. The number of gray nodes in a set depends on the total number of children gray nodes and the total number of children black nodes of this parent black node.

The set of gray nodes is formed according to their location information and the location information of the child black node (to communicate with whose cluster the set is being formed). The gray nodes with minimum distance to the child black node (for whose cluster the set is being formed) get preferences to be selected for the set.

After forming the set of gray node to communicate with the child black node's cluster, the parent black node sends the location information of the gray nodes of the set to that child black node. The child black node then selects gray nodes (which are nearer to the selected gray nodes of the parent black node) from its cluster to communicate with the parent black node's cluster.

The child black node then sends a message to the selected gray nodes of its cluster to let them know the cluster (cluster of the parent black node) with which they will communicate. The child black node then sends an acknowledgement to the parent black node that the set of gray nodes has been formed.

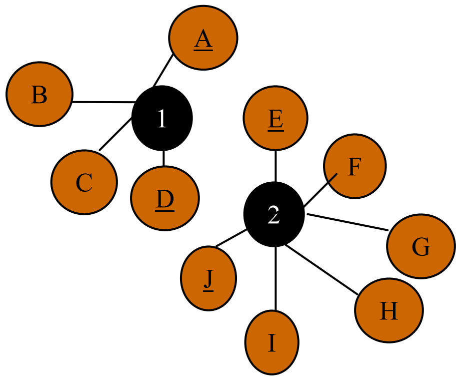

Figure 7. Illustration for duty cycle assignment

The parent black node then sends a message to the selected gray nodes of its cluster to let them know the cluster (cluster of the child black node) with which they will communicate.

A black node selects a gray node from each set to be in active mode and other nodes of a set will be in sleep mode for this cluster pair but they may be in active mode for any other cluster pair. The 'mode set' message is sent from the parent black node to the children gray nodes.

A gray node may give up its active mode for a cluster pair when it has spent a certain amount of energy. It sends a signal to the parent black node and goes to sleep mode. The parent black node then selects another gray node from this set to be in active mode.

The black node listens to all packets while these are being forwarded. The black node forwards a packet if the sending node is out of range of the active forwarding node.

In Fig.: 7 node: 1 selects node: A and node: D to communicate with the cluster of node: 2 and node: 2 selects node: E and node: J to communicate with the cluster of node: 1.

Node: 1 sets node: A to be in active mode and node: D to be in sleep mode for the cluster pair of node: 1 and node: 2. Node: 2 selects node: E as the active node for this cluster pair.

If node: C sends a packet to the cluster of node: 2, node: 1 forwards this packet to the active node: A, because forwarding node: A is out of the range of the sending node: C. Node: E (Active node of the cluster of node: 2) receives this packet.

Gray nodes selected by the child black node must be within the communication range of all the gray nodes selected by the parent black node for this cluster pair. Child black node may need to send a 'node-set update' message to the parent black node to update the set of gray nodes so that the child black node can have a minimal number of gray nodes in the node-set for this cluster pair. A number of message passing between parent-child black nodes may be required in this case.

VII. CONCLUSION AND FUTURE WORK

The topology discovery and fault tolerance mechanisms; described in this paper are distributed and highly scalable and thus can be efficiently applied to sensor networks. By applying these fault tolerance mechanisms connectivity among operational nodes can be ensured with maximum availability when a node fails.

The period of time of caching the packet (transmitted to a faulty node) by a node depends on its available energy, the total number of cached packets (to be transmitted to the faulty nodes) and the probability of a faulty node to be reinstalled to operational state within a specific period of time. Constructing a mathematical model to calculate the expected time of caching a packet by a node will be the future field of research to upgrade the fault tolerance mechanisms for sensor networks.

REFERENCES

[1] "A Topology Discovery for Sensor Networks with Application to Network Management", http://www.cs.rutgers.edu/dataman/papers/TopDisc.pdf.

[2] Iavor Georgeff, "A Distributed Topology Discovery Algorithm for Wireless Sensor Networks", http://www.csse.uwa.edu.au/~rachel/AdHocProjects/iavor.georgeff.hons 2004.dissertation.pdf.

[3] Hui Tian, "Network Topology Discovery and Its Application", http://www.jaist.ac.jp/library/thesis/is-doctor-2006/paper/hui-t/paper.pdf.

[4] Ranveer Chandra, Christof Fetzer, Karin Hogstedt, "Adaptive Topology Discovery in Hybrid Wireless Network", http://research.microsoft.com/~ranveer/docs/topology.pdf

[5] Krishnamachari, L.; Estrin, D.; Wicker, S, "The impact of data aggregation in wireless sensor networks", Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on Volume, Issue, 2002 Page(s): 575 – 578.