Localization-Free and Energy-Efficient Hole Bypassing Techniques for Fault-Tolerant Sensor Networks

Onur Yilmaz[a,b], Orhan Dagdeviren[b], Kayhan Erciyes[c]

[a]New Jersey Institute of Technology,
Electrical and Computer Engineering, University Heights, Newark, NJ 07102, USA
[b]Ege University,
International Computer Institute, Bornova, Izmir 35100, Turkey
[c]Izmir University,
Computer Eng. Dept., Uckuyular, Izmir 35350, Turkey

Abstract

Nowadays, since wireless sensor networks (WSN)s are increasingly being used in challenged environments such as underground mines, tunnels, oceans and the outer space, fault-tolerance need has become a major requirement for routing protocols. So far, the proposed fault-tolerance methods or algorithms aim to recover the isolated failures which occur different parts of the network in different times. However, there is another type of failure for WSNs which is more destructive for the applications. By collapsing sensor nodes as a group at the same time, a hole can appear at the network which may cut the data delivery drastically. In literature, previous studies for bypassing holes are based on localization which may have significant energy and economic costs. In this paper, two localization-free and energy-efficient algorithms are proposed for bypassing the holes formed by group collapse. We realized that when holes are modeled with clusters, hole bypassing can be solved by cluster bypassing. Our algorithms, Intra-Cluster Bypass and Inter-Cluster Bypass aim to heal the corrupted communication links in the presence of holes. We show the operation of the algorithms, analyze them and provide extensive simulation results in ns-2 environment. We compare our proposed algorithms with the other approaches and show that our algorithms significantly improve the fault recovery percentages while consuming reasonable amount of energy

Email addresses: `oy6@njit.edu` (Onur Yilmaz), `orhan.dagdeviren@ege.edu.tr` (Orhan Dagdeviren), `kayhan.erciyes@izmir.edu.tr` (Kayhan Erciyes)

even in the presence of high collapse ratio.

Keywords: Wireless Sensor Networks, Fault-Tolerant Routing, Fault Tolerance, Bypass, Bypassing Holes, Resilience, Reliable Routing

1. Introduction

Advances in hardware and wireless network technologies have resulted in low-cost, low-power, multi-functional miniature sensor devices [1, 2, 3, 4, 5, 6]. These sensor nodes are small, with limited processing and computing resources, and they are inexpensive compared to traditional sensors. These tiny sensor motes can sense, measure, and gather information from the environment and, based on local decision processes, they can transmit the sensed data to a base station (sink). By communicating each other and conveying information which is gathered from the environment, they form a huge network which cannot be compared with the other type of networks. WSNs are generally large scale and distributed systems which are composed of ten to thousand of sensor motes that are communicating with each other. They are used in diverse applications such as military target tracking and surveillance, natural disaster relief and biomedical health monitoring [7]. Although WSNs are used for various purposes, they contain challenges such as limited energy and wireless communication problems.

WSNs are increasingly being used in challenged environments such as underground mines, tunnels, oceans and the outer space. Wireless communication in challenged environments have transmission failures, mainly as a consequence of direct impact of physical world. In addition to energy constraints and wireless communication problems, tiny sensor motes are prone to failures. In [8], sources of all fault types are proposed in detail. In many critical applications of WSNs, the communication in challenged environments has to be reliable and therefore this requirements bring a need for the faults to be detected and recovered timely.

In particular, in a sensor network which is deployed in an extreme environment, each node may individually fail or a group of geographically close located nodes may collapse which forms holes [9]. A hole may prune the communication links in a sensor network where data transfer from hole affected regions of the network becomes impossible without necessary topology recreation. In this case, sensor network applications have to suspend their communications and data delivery during construction of new paths. Thus, network recovery in the presence of holes is a very important problem. Although, this problem is addressed and studied by the researchers [9, 10, 11, 12, 13, 14], the main focus lies in node localization based approaches where sensor nodes should be equipped with a position tracker or localization algorithms should be executed priori on these nodes. Equipping nodes with a position tracker like a global positioning system (GPS) receiver may cause significant costs. On the other hand, executing complicated localization algorithms may exhaust the batteries of sensor nodes which may cause new faults. In either cases, localization may introduce a considerable cost to the sensor network.

In this study, we aim to design localization-free and energy efficient hole bypassing techniques for fault-tolerant sensor networks. Our idea is firstly to construct a cluster tree rooted at sink node where network is partitioned into multi-hop clusters. By applying this strategy, we aim to model holes with clusters. Afterwards, to recover the communication links in the network, we propose an intra-cluster energy-efficient solution in the first step and an inter-cluster robust solution in the second step. By applying these methods, we aim to avoid the cost of localization and network-wide topology recreation.

The rest of this paper is organized as follow. In Section 2, we review the related work on fault recovery techniques. We show the network model and hole problem formulation in Section 3. In Section 4, we introduce the proposed intra-cluster and inter-cluster based methods. We show the simulation results of the proposed methods and its performance comparison with the related work in Section 5. The conclusions are drawn in Section 6.

2. Related Works

Routing is an attractive research area in all networks. Like other networks, researchers have proposed various protocols and algorithms for conveying messages to sink and dealing with the challenges of WSNs.

The early studies on routing in WSNs dealt with the energy problem and they tried to optimize the energy consumption. One of the answers to this problem was data-centric routing mechanism. Sensor protocols for information via negotiation (SPIN) [15] is one of the earliest works to pursue data-centric routing mechanism in WSNs. Then, Directed Diffusion [16] was proposed which is an important milestone in data-centric routing mechanisms. After these studies, many approaches which pursue data-centric routing were proposed including the study in [17]. On the other hand, clustering-based protocols were proposed around the same time, in order to solve the energy problem. Low energy adaptive clustering hierarchy (LEACH) [18] forms clusters regarding received signal strength indicator (RSSI) of sensor nodes which is one of the earliest study in clustering-based protocols. However, LEACH is inefficient in terms of energy since it lacks multi-hop routing. Then, threshold-sensitive energy-efficient sensor network protocol (TEEN) [19] and adaptive periodic threshold-sensitive energy-efficient sensor network protocol (APTEEN) [20] were proposed which are designed to respond reactively to sudden energy changes efficiently.

Then, the focus of network community shifted to fault-tolerance because sensor nodes are prone to failures. Since the event packets are conveyed over the sensor nodes hop by hop towards sink, any fault of sensor nodes can cause event packet losses. In particular, this is a serious problem for critical systems. In [21], the periodic event-driven and query-based protocol (PEQ) and its variation clustering periodic event-driven query-based protocol (CPEQ) which are fault-tolerant and low-latency routing protocols are proposed. In [22], inter-cluster communication based energy aware and fault-tolerant protocol (ICE) is proposed which is a cluster based, energy aware and fault-tolerant routing protocol for WSNs. In [23], variable transmission range protocol (VTRP) is proposed for smart dust networks, a special type of WSN, which is an energy-efficient and fault-tolerant protocol using a variation of the transmission range. This is the first study for data propagation in literature which uses a varying transmission range technique. It is pointed out in this paper that additional knowledge, obtained by increasing the transmission power in a distributed manner improves the data propagation to sink in terms of fault tolerance. In [24], a fault-tolerant and efficient data propagation protocol for WSNs is proposed which uses the varying transmission range technique same as in [23]. In [25], the resilience of directed diffusion is increased by constructing disjoint and braided multipaths.

When the fault-tolerant routing algorithms and protocols in the literature are examined, they usually gather around the same fault-tolerant techniques including Disjoint and Braided Multipaths, Instant Recovery and Reconstructing the Routing Paths. Although these techniques provide reasonable recoveries, they do not intend to recover the holes which is formed by sudden group collapse. Although there are some studies intended to solve hole problem [9, 10, 11, 12, 13, 14], they are mainly based on bypassing holes using localization techniques. Since our concern is localization-free hole bypassing methods, we omit these localization based studies. The related methods are as follows:

- Disjoint Multipath Routing: In this routing scheme, the alternate paths are node disjoint with the primary path and with each other [25]. Because of this property, a node failure in the primary path does not affect the alternate paths and so on. Node disjoint paths can be constructed by executing localized algorithms like directed diffusion [16]. Although node disjoint paths are fault-tolerant, it may not always be possible to construct disjoint paths especially in sparse networks. Besides, node disjoint paths can be energy inefficient since alternate paths can be longer than the primary path.

- Braided Multipath Routing: Node disjointedness requirement is relaxed in braided multipath routing where alternate braided paths are partially disjoint from the primary path [25, 22]. Braided paths from a source node to the sink node can be constructed as follows: For each node v on the primary path, find the best possible path from source node to sink node that does not include node v. The alternate paths are expected to be geographically close to the primary path, thus the technique is energy-efficient intuitively. Alternate paths can be constructed with a localized algorithm similar to disjoint multipath routing. Although this technique provides a fault-tolerant infrastructure, it is bounded with the parent-child relationships which are constructed priori, and it may not reactively recover faults in the presence of holes.

- Instant Recovery: In this technique, when a node detects its parent's fault, it reactively tries to recover the fault by searching alternative parent [21]. In order to find a suitable alternative parent, the node m broadcasts a $Search(m.level)$ message to its neighbors. When the neighbor node v receives a $Search(level)$ message, it replies with an acknowledgment if $v.level \leq Search.level$ and it is not the child of the sender of $Search$ message. In this manner, the loops are prevented. PEQ and CPEQ protocols use this technique to recover faults [21]. Although this technique provides a fast and low cost recovery, it is bounded with the level of nodes and the event delivery percentage may not be adequate when many of nodes collapse.

- Re-executing Topology Construction Algorithms: Previously described methods can be inadequate to recover faults in challenged environments where holes are present and fault rate is high. In this situation, topology may be periodically reconstructed in order to continue data delivery. When topology reconstruction is applied, new links are chosen for data communication and parent-child relationships are updated with the level information. Fault-tolerant protocols like CPEQ and ICE use this method periodically to regenerate links [21, 22]. Many other topology construction algorithms offer this method for fault recovery [26, 27, 28]. Although this method may provide idealized recovery from the faults, it has important drawbacks. Firstly, in the duration re-execution is applied, nodes may not be able to transmit their data packets. This may cause too much delay in data delivery operation. Secondly, the energy cost of this operation can be very high. Assuming that each node at least sends one message during re-execution, the message cost of this operation is $\Omega(N)$ where N is the total node count.

3. Problem Formulation

3.1. Network Representation

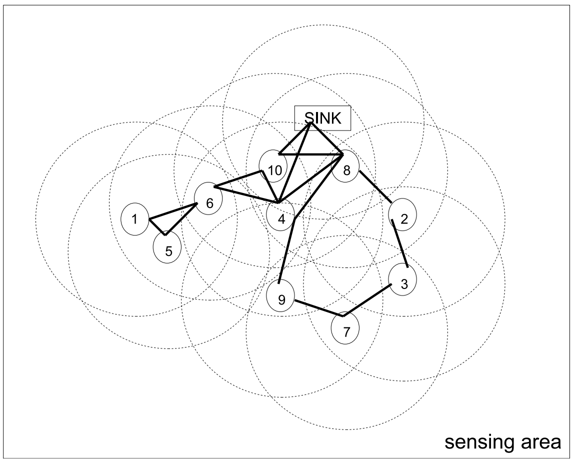


Figure 1: Network Model

The following assumptions are made about the network:

- Each node has distinct *node_id*.
- The nodes are stationary.
- Links between nodes are symmetric. Thus if there is a link from u to v, there exists a reverse link from v to u.

- Nodes do not know their positions. They are not equipped with a position tracker like a GPS receiver.

- All nodes are equal in terms of processing capabilities, radio, battery and memory.

- Each node knows its neighbors.

Based on these assumptions, the network may be modeled as an undirected graph $G(V, E)$ where V is the set of vertices, E is the set of the edges. An example undirected graph model is depicted in Fig. 1 where 10 ordinary sensor nodes and a sink node are located on the sensing area. Each node is labeled with its identifier, the transmission ranges of the nodes are shown with dotted circles, and the transmission edges between two nodes are shown with solid lines.

3.2. Hole Problem

Holes may occur in WSNs located in various environments having challenging conditions. A hole may stop the operation of the sensor network completely or may partition the network into disjoint parts which may significantly reduce the event collection. To recover faults in the presence of holes, we propose to construct a multi-hop cluster tree architecture. This architecture is generic, it can be an arbitrary multi-hop cluster tree rooted at the sink and can be constructed by topology generation algorithms such as given in [21, 22, 28, 29, 30, 31].

An example case is shown in Fig. 2 where the transmission edges between two nodes are shown with dashed lines, the border of clusters are shown with solid circles, the cluster edges are shown with directed solid edges. Each cluster has an identification (id) varying from A to M. Clusters are leveled from 0 to 4 by calculating their cluster count on the shortest path to sink node. The cluster levels can be found by a graph traversal algorithm like breadth-first search (BFS) where the level of sink node is 0, the levels of neighbor clusters of sink node are 1, and the levels of other clusters are their cluster count on the path to sink node after BFS is executed. A node is identified by concatenating its cluster id and node id, e.g H_5. Each cluster head (CH) has an id of 1 and is shown in black. Level of a node is its hop distance from its CH. For example, H_5's level is 2, since it can reach H_1 in 2 hops. The hole is shown with a bold closed spline. Hole includes all nodes in

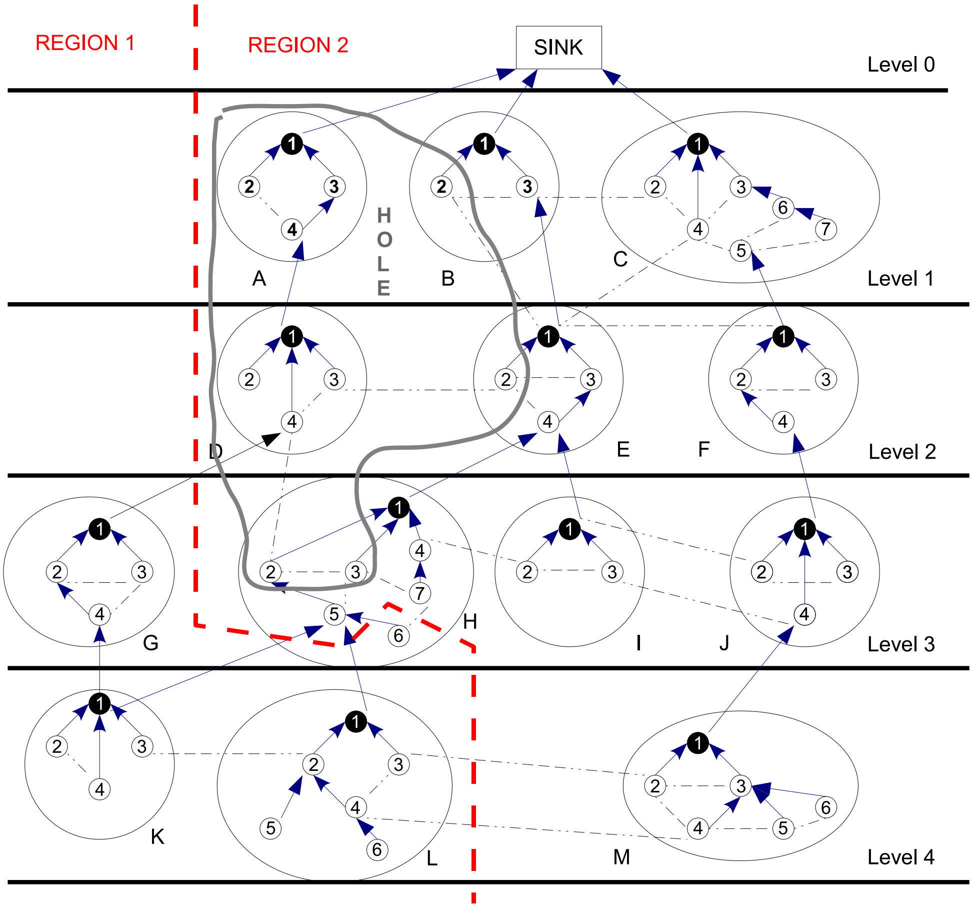


Figure 2: Hole Problem

cluster A and D, node H_2, node H_3, node B_2 and node E_2 as shown in Fig. 2.

After the formation of the hole, H_5 cannot transmit its packet since its parent H_2 has crashed. H_5 cannot recover its link by using instant recovery and multipath routing techniques because H_5's remaining neighbor is H_6 and its level is higher than H_5's level. Besides, the packets coming from cluster L and cluster K cannot be forwarded to the sink node, because H_5 is the parent of both L_1 and K_1. Although H_6 can recover its link and set its new parent as H_7, this healing operation will only benefit itself. Moreover, since D_4 is the parent of G_1 and it has crashed, packets coming from cluster G cannot be relayed to sink node. Instant recovery and multipath routing techniques cannot recover the link of G_1, since G_2 and G_3's levels are higher than G_1's level. We may state from this example that holes may significantly reduce the performance of sensor networks using instant recovery and multipath routing fault-tolerant techniques.

To heal all the corrupted links which cannot be recovered by instant recovery and multipath routing techniques, a network wide re-execution of the clustering protocol can be applied. On the other hand, nodes can redundantly consume their energy by executing this technique. In Fig. 2, network area is divided into two regions bounded by a bold dashed line. The Region 1 covers sensor nodes which cannot send their data packets to the sink node. The Region 2 consists of the hole and the other working nodes which can send their data packets to sink. Although the links of nodes in Region 1 are recovered by the re-execution, links of nodes in Region 2 remain the same. Since Region 2 is much more greater than Region 1, we may state re-execution may inefficiently consume the energy of sensor nodes. In addition, the nodes in Region 2 cannot send event packet to the sink node during reconstruction phase. Our example hole scenario depicted in Fig. 2 which is explained so far may occur in various forms in sensor networks and these problems can be generalized. In this manner, our problem is localization-free and is an energy-efficient hole bypassing techniques for multi-hop clustered and fault-tolerant sensor networks.

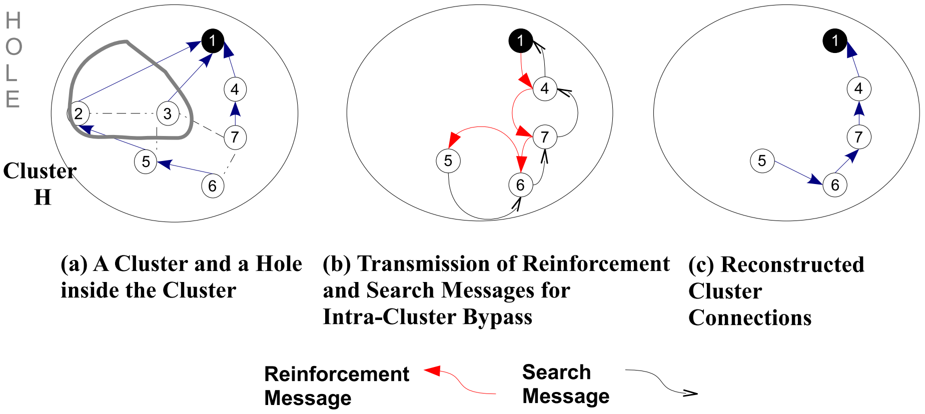
4. Proposed Methods

In this section, we will describe two localization-free and energy efficient methods that we propose for fault tolerance in multi-hop clustered sensor networks.

4.1. Intra-Cluster Bypass

We propose intra-cluster bypass technique which is designed to heal the corrupted intra-cluster links. Our aim is to recover the communication paths in a corrupted cluster by applying local operations residing in the same cluster. Our method is both fault tolerant and energy-efficient in this manner.

In some cases, a hole may span just a partition of a cluster as depicted with cluster H in Fig. 2. Instant recovery cannot recover the link of a node whose neighbors have higher cluster level than this node as shown with node H_5 in Fig. 2. Our goal is to reconstruct intra-cluster paths in these cases.



(a) A Cluster and a Hole inside the Cluster **(b) Transmission of Reinforcement and Search Messages for Intra-Cluster Bypass** **(c) Reconstructed Cluster Connections**

Figure 3: Intra-Cluster Bypass Example

When a node (node v) cannot find an alternate parent, it initiates intra-cluster bypass operation by broadcasting an intra bypass *search* message to its neighbors. The neighbors of node v which are at the same cluster forward this message just once to their neighbors by broadcasting where this message is relayed to node v's CH and the nodes residing in different clusters do not respond to this message. When the CH of node v firstly receives this message, it unicasts a $reinforcement$ message back to the forwarder of $search$ message where this message is relayed to the node v by unicasting in backward direction. In this way, the path is reconstructed. The pseudocodes of the procedures for this algorithm are given in Alg. 1 and 2. The features of the proposed intra-cluster bypass technique are listed below:

- The method recovers links locally in the corrupted cluster without disturbing the operation on the other clusters.
- The method provides cycle-free routes and it is resource-efficient as proved in Theorem 1 and Theorem 2.

An example operation is shown in Fig. 3. The corrupted cluster H from Fig. 2 is depicted in Fig. 3. Node H_2 and H_3 are corrupted which causes that node H_5 and node H_6 cannot send their packets to sink node as shown in Fig. 3.a. Because of this, node H_5 initiates sending $search$ message and this message is relayed to the node H_1. $reinforcement$ message originated from Node H_1 is sent backward along the path of $search$ message. These

Algorithm 1: Intra Bypass - Search

```
upon node $r$ receives an intra bypass $search$ message from node $s$;
if $search.clusterID = clusterID$ then
    if search message with search.searchID has firstly received then
        store the sender and $search$.searchID as pair in tableIntraSearchMessages;
        if the node is cluster head then
            unicast a $reinforcement$ message to node $s$;
        else
            broadcast the same $search$ message;
        end
    end
end
end upon
```

Algorithm 2: Intra Bypass - Reinforcement

```
upon node $r$ receives an intra bypass $reinforcement$ message from node $s$;
set node $s$ as new parent;
if node $r$ is the initiator of the intra bypass then
    continue to relay events because the path is repaired;
else
    find the sender of intra bypass $search$ message from tableIntraSearchMessages;
    unicast $reinforcement$ to the sender of intra bypass $search$ message;
end
end upon
```

message transfers are shown in Fig. 3.b. Finally, the path is recovered as H_5 → H_6 → H_7 → H_4 → H_1 as shown in Fig. 3.c.

Theorem 1. *Intra-cluster bypass operation is free from cycle formation.*

Proof. In an intra-cluster operation, each node sends *search* message once. The *reinforcement* message can only be originated by a CH once for each *search* message. Thus the corrupted path cannot be recovered without the CH. Since a CH's parent is not in the same cluster and CH's level is smaller than its cluster members, formation of a cycle is not possible.

Theorem 2. *Assume that C_m is maximum node count in a cluster, D_m is the maximum cluster diameter, the message and the time complexities of an intra-cluster bypass operation are $O(C_m)$ and $O(D_m)$ respectively. The message size is $O(log_2(N))$ and space complexity is $O(C_m)$ where N is the number of nodes.*

Proof. All nodes in the corrupted cluster broadcast *search* message and nodes along the recovered path unicast $reinforcement$ message. Assume that corrupted node count in cluster is f and count of nodes along the recovered path is r, total number of messages in this case is O(C_m+r-f) $\in$ O(C_m) when $0 < r < C_m$ and $0 < f < C_m$. At most, propagation of *search* message in the corrupted cluster takes O(D_m) time.

Since all fields in *search* and $reinforcement$ messages may be in (0,N) interval, the message size is O($log_2(N)$). Each node should store a table including (sender,ID) pair where this table can have a size of at most O(C_m).

4.2. Inter-Cluster Bypass

When a fault occurred in a cluster, the nodes in this cluster firstly try to recover it by in-cluster fault-tolerant techniques. However, it may not be possible to recover the fault in some cases especially when a hole may span a whole cluster or major part of it. For example, node G_1 cannot send its packets directly to a cluster at the upper level since all nodes in cluster D have failed as shown in Fig. 4.

In the multi-hop clustered sensor network architectures, every cluster except sink node's cluster has an up cluster (parent cluster). These clusters are responsible to send their data as well as to relay the packets of their down clusters to their up clusters. When a cluster's up cluster has failed, it can send its packet to sink node and it cannot use intra-cluster bypass to recover, as mentioned in the previous paragraph. To overcome this situation, we propose inter-cluster bypass technique. Our aim is to provide the down clusters keep relaying message to sink. In our technique, the CH(s) of down cluster(s) search for an alternate CH for constructing a path towards to it and relaying its messages to sink with bypassing its up cluster. By relaying the messages over the alternate CH, the paths of down clusters can pass its up cluster.

When a CH with cluster level (*cl*) cannot find an alternate parent, it initiates inter-cluster bypass operation by broadcasting an inter-cluster bypass *search* message to its neighbors. The CH's neighbors residing in different clusters forward this message just once to their neighbors. This message is forwarded once by the other nodes if their cluster level (*nl*) is *cl*+*m* $\geq$ *nl* $>$ *cl*-*m* where m is a previously defined constant. When a CH with cluster level (*bl*) receives a *search* message, it originates a $reinforcement$ message once to the forwarder of *search* message, if *cl* $\geq$ *bl* $\geq$ *cl*-*m*. Also this CH should not use a path constructed with inter-cluster bypass to avoid cycle

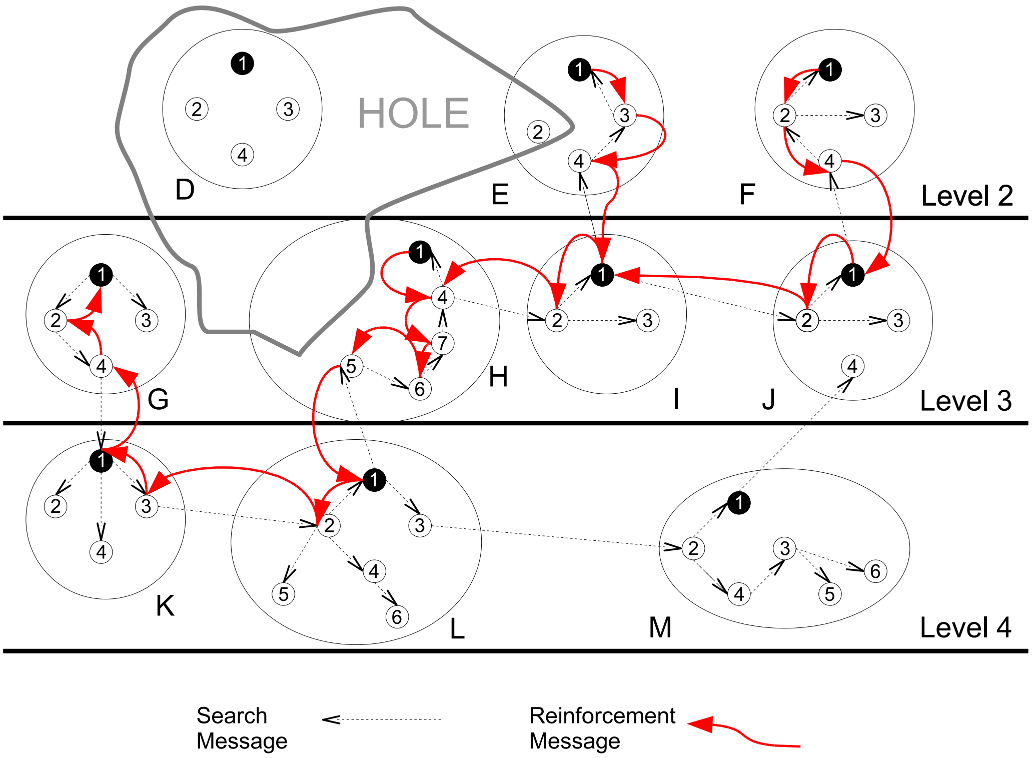


Figure 4: An Inter-Cluster Bypass Example

formation. The *reinforcement* message sent by CH is relayed to the *search* originator CH by unicasting it from backward direction where the path is recovered. There may be more than one CH sending a *reinforcement* message, in this case the first one constructs the path, operation of other CHs are omitted. The pseudocodes of the procedures of this algorithm are given in Alg. 3 and 4. The features of the proposed inter-cluster bypass algorithm are summarized below:

- The proposed method provides a recovery technique when a hole spans more than one cluster.

- The proposed method is cycle free as proved in Theorem 3. The time complexity depends on the diameter of the network, the message complexity depends on the m constant, maximum cluster count at the same level (k), the maximum node count in a cluster (C_m) and the network diameter as proved in Theorem 4. When k, C_m and $D \in$ O($\sqrt{N}$), the message complexity is linear.

- In the worst case, the count of fault recovery alternatives produced by the inter-cluster bypass technique is at least the count of fault recovery alternatives produced by other methods as proved in Theorem 5.

Algorithm 3: Inter Bypass - Search

```
upon node $r$ receives an inter bypass $search$ message from node $s$;
if $search$ message with $search$.searchID has firstly received then
    store the sender and $search$.searchID as pair in tableInterSearchMessages;
    if $search.clusterLevel \geq clusterLevel$ and ($search.clusterLevel - m) < clusterLevel$ then
        if node $r$ is cluster head and does not use inter bypass then
            unicast a $reinforcement$ message to node $s$;
        else
            broadcast the same $search$ message;
        end
    else if $search.clusterLevel < clusterLevel$ and ($search.clusterLevel + m) \geq clusterLevel$ then
        broadcast the same $search$ message;
    end
end
end upon
```

An example inter-cluster bypass operation with m=1 is depicted in Fig. 4 where clusters from Fig. 2 are shown. The hole in the figure spans cluster D and node E_2. Since cluster G's up cluster is cluster D, node G_1 cannot send its cluster's packet to sink node. So node G_1 starts inter-cluster bypass operation by broadcasting a *search* message to its neighbors and this message is relayed to all nodes in clusters at level 2, level 3 and level 4. Node H_1 which used intra-cluster bypass to recover its residual cluster links, forwards a $reinforcement$ message along the path of received *search* message. All CHs in clusters with level 2 and level 3 send $reinforcement$ messages same as node H_1. Since node H_1's $reinforcement$ message is received before than the other CHs' messages, node H_1 constructs the inter-cluster bypass path which is $G_1 \rightarrow G_2 \rightarrow G_4 \rightarrow K_1 \rightarrow K_3 \rightarrow L_2 \rightarrow L_1 \rightarrow H_5 \rightarrow H_6 \rightarrow H_7 \rightarrow H_4 \rightarrow H_1$.

Theorem 3. *Inter-cluster bypass operation is free from cycle formation.*

Algorithm 4: Inter Bypass - Reinforcement

```
upon node $r$ receives an $reinforcement$ message from node $s$;
if $reinforcement$ message with the search id $reinforcement$.searchID has firstly
received then
    store the sender and $reinforcement$.searchID as pair in tableByPassParents
    for bypass path;
    if node $r$ is the initiator of the inter bypass then
        continue to relay events through the new path because the fault is
        recovered;
    else
        find the sender of inter bypass search message from
        tableInterSearchMessages;
        unicast $reinforcement$ to the sender of inter bypass search message;
    end
end
end upon
```

Proof. In an inter-cluster bypass operation, each node sends *search* message once. The $reinforcement$ message can be originated by CHs with equal or smaller cluster levels, once for each *search* message. Also these CH's should not have used inter-cluster bypass previously. By applying these constraints, formation of a cycle becomes impossible.

Theorem 4. *The message complexity of an inter-cluster bypass operation is O($mk(C_m$+$D)$) and the time complexity is O(D) where k is the maximum number of clusters at the same level and D is the network's diameter. The message size is O($log_2(N)$) and the space complexity is O(C_m).*

Proof. When a CH with cluster level(l_b) tries to bypass its upper cluster, it sends *search* message where this message is propagated to the clusters with level(l_o) providing l_b+m $\geq$ l_o $\leq$ l_b-m. Thus, the message is propagated to $2m$+1 levels of clusters excluding the faulty cluster. The number of messages for this operation is mkC_m. A mk-1 number of CHs may send *reinforcement* messages, so (mk-1)D *reinforcement* messages may be transferred including routing operations. The number of total messages is mkC_m+(mk-1)D $\in$ O(mk(C_m+D)). The time complexity for the dissemination of *search* messages is O(D) and similarly transmission of *reinforcement* messages takes O(D) time, thus total time complexity is O(D). All fields in messages may be in (0,N) interval, thus the message size is O(log_2(N)). Each node should store a table including (sender,ID) pair where this table can be at most of size O(C_m), same as intra-cluster bypass.

Definition 1. *A cluster border node has at least one neighbor residing in another cluster. Other nodes are cluster non-border nodes.*

Theorem 5. *Assume that the count of node v's alternate parent that can be chosen by the instant recovery and the braided multipath technique is i, by the intra-cluster bypass is a and by applying the inter-cluster bypass technique is b. Also assume that l_h is the count of inter-cluster links which connect node v to higher level nodes. For cluster border nodes with (Δ-l_h) $> l_h$ and for all cluster non-border nodes, the fault recovery order is: $0 \leq i \leq a \leq b \leq \Delta$. For the other nodes the fault recovery order is $0 \leq a \leq i \leq b \leq \Delta$.*

Proof. The crashed parent can be the only neighbor of fault detecting node, thus lower bound is 0. In all techniques the new parent should be chosen in the non-faulty neighbor set, thus upper bound is Δ. The non faulty neighbor nodes with equal or smaller level than the level of the fault detecting node can be chosen as the new parent in instant recovery and braided multipath techniques. The non faulty neighbor nodes in the same cluster can be chosen as the new parent in intra-cluster bypass. When (Δ-l_h) $> l_h$, as a result of lower level intra-cluster recovery alternatives being higher, the order is $i \leq a$. Obviously, this case is true for all cluster non-border nodes. All non faulty neighbors without any restriction are the candidates for the new parent in inter-cluster bypass technique, thus $a \leq b$. The final order when (Δ-l_h) $> l_h$ is: $0 \leq i \leq a \leq b \leq \Delta$. When ($\Delta$-$l_h$) $< l_h$ is true, as the higher level inter-cluster alternatives are more than other alternatives, the order becomes: 0 $\leq a \leq i \leq b \leq \Delta$.

5. Performance Evaluations

In order to evaluate the performance of proposed two fault-tolerant techniques in the presence of group collapse, we have carried out an extensive simulation study using ns-2 simulator. We have conducted two simulation studies over two different protocols including a generic protocol and CPEQ [21]. In the simulation study with the generic protocol, we compared the proposed techniques with well-known fault-tolerant techniques including instant recovery (search), braided-multipath and global re-execution in terms of event delivery percentage, energy consumption and delay. When the well-known fault-tolerant routing protocols [21, 22, 23, 24, 25] are examined deeply, it can be clearly seen that these techniques are mainly used for fault recovery purposes. Besides, these techniques are utilized in various protocols by employing either only one of them or as a combination of multiple techniques. On the other hand, in the simulation study with CPEQ protocol, we extended CPEQ protocol with the proposed fault-tolerant techniques and evaluate its performance in terms of event delivery percentages and energy consumptions. CPEQ protocol [21] is one of the widely accepted cluster-based fault-tolerant routing algorithms in literature. The aim of this simulation study is also to show how the proposed techniques can be implemented to a well-known cluster based protocol.

We designated the simulation parameters by taking into consideration the parameters in CPEQ and Directed Diffusion [16] for both simulation studies. The initial energy of sensor nodes was set to 100 joule. Random generated topologies were used in different node counts ranging from 100 to 500 nodes for average node degrees varying between 5 and 23. The position of sink was also set randomly. IEEE 802.15.4 radio and medium access control (MAC) layer standards readily available in *ns-2* simulator were chosen for lower layer protocols. The transmission range of nodes were set to 20 m. The transmission, receiving and idle powers were set to 0.660 w, 0.395 w and 0.035 w respectively. The simulation parameters are summarized in Table 1.

Table 1: Experimental Study Parameters

Simulation Time (s)	1000
Number of Nodes	100 - 500
Sink Position	Randomly placed in the area
Percentage of Source Nodes (%)	2
Source data rate (eventMsgs/s)	2
Radio range (m)	20
MAC	802.15.4
Transmit Energy (w)	0.660
Receive Energy (w)	0.395
Dissipation in Idle (w)	0.035
Initial Energy (j)	100
Node degrees	5-23
Percentage of Node failure (%)	10, 20 and 30

5.1. Performance Evaluations on Generic Protocol

In this simulation study, we use a generic protocol in order to asses the performance of two proposed fault-tolerant techniques in the presence of group collapses. We call it as generic protocol since many researchers used these type of topologies for data delivery [29, 22, 26, 30, 31]. Besides implementing the generic protocol, we evaluated the performance of instant recovery, braided-multipath and global re-execution, and compared with the two proposed techniques. The reason to choose these techniques is almost all localization-free fault-tolerant routing protocols just use these techniques and their very similar versions for fault recovery as mentioned in Section 2.

The paths are constructed towards the sink with the shortest hop algorithm in the generic protocol for relaying the messages energy efficiently. The algorithm is based on flooding the network which is initiated by the sink with a hop value. The neighbor nodes of the sink store the received hop value, increment it and transmit it to its neighbor nodes and so on until the whole sensor network is configured with different levels of hops. In order to confine the message diffusion, a node does not always transmit a new hop message after it receives a hop message. When a node receives a hop message from its neighbor, it checks this value against its local hop value. If the local hop value is greater than the received one, the node updates its hop, increment this value and retransmit it to its neighbors. Each node sets the sender of minimum hop message as a parent node which is used for forwarding the messages towards sink. After the algorithm terminates, a shortest hop tree is formed whose root is the sink.

Since the proposed techniques are for cluster based protocols, clusters have to be formed in the generic protocol. Therefore, we used the method in [29] for forming clusters over a tree. This technique is very simple and it does not have any overhead. A constant depth value is defined in this method and if (hop level *mod* depth = 0) then, the node becomes the cluster head, otherwise it is an ordinary node in the cluster. This technique can also be used as a tree based protocol in order to form simply clusters. Therefore, although the proposed techniques are for cluster based protocols, they can be implemented to tree based protocols by simply forming clusters with this method. In order to asses the performance of the proposed techniques for different cluster sizes, the simulation study is conducted by setting the depth value to 3, 5 and 7 in order.

The delay, energy consumption and resilience of instant recovery, braided-multipath, global re-execution and proposed techniques are evaluated over a sample WSN application which is same in the [21] and [16]. Two percent of the nodes of the network are selected randomly per 0.5 second in order to generate a random traffic. The selected nodes generate event packets and these packets are relayed to sink through the shortest hop tree. After a node transmits a packet to its parent in the tree, it waits an acknowledgement from its parent for reliable communication. If a sender node cannot receive an acknowledgement in a constant time, it retransmits the packet. Otherwise, if it cannot receive any acknowledgement after ten retransmissions, the sender node detects a fault and performs a recovery algorithm.

During the simulation, group collapse is simulated by turning off the fixed fraction of nodes. At a random time in the simulation, randomly selected nodes and its neighbors are turned off until fixed fraction of nodes are reached, and thereby, this type of faults form big and small holes in the network.

The implemented methods are utilized regarding their fault recovery merits and energy consumptions in various routing protocols. Generally, the cheapest method in terms of energy consumption is firstly performed. However, it usually offers lower fault recovery rate. Other costly methods are performed adaptively if they are required. Thus, in order to show how the proposed methods can be used coherently with these methods and to provide a comparison, we set up twelve different simulation series. The series can be divided into two groups. In the first group of series, each technique is executed solely in order to measure its performance. We implemented no-fault tolerance algorithm, braided multipath, instant recovery, intra-cluster bypass and global re-execution techniques. Since inter-cluster bypass is designed for joint use with another technique, we did not implement it alone. The series of the first group is as follows:

- *No Fault-Tolerance Algorithm (No Tolerance):* In this simulation setup, only the generic protocol is used which does not perform any fault-tolerance algorithm in the presence of a fault. Therefore, *No Tolerance* is expected to show the worst event delivery percentage.

- *Braided Multipath (BM):* Braided multipath fault-tolerant technique is only used in the generic protocol. Braided multipaths are constructed while the shortest hop algorithm is performing. In the presence of a fault, a node tries to recover the fault via using another path in the braided multipath.

- *Instant Recovery (search) (IR):* Instant recovery fault-tolerant technique is used in the generic protocol. The node discovers a new node from lower layer instantly in order to recover the fault and keep relaying event messages.

- *Intra-Cluster Bypass (IntraBp):* Intra-cluster bypass fault-tolerant technique is used by a cluster member node in the presence of a fault.
- *Global Re-execution (GR):* Shortest hop algorithm is re-performed per 10 seconds for fault recovery. This technique is optimal in terms of fault recovery but it consumes high energy as well as all nodes in the network have to wait for termination of the algorithm in order to keep relaying events to the sink.

The second group includes integrated techniques for in-cluster and network-wide fault tolerance. Braided multipath, instant recovery and intra-cluster bypass are used for in-cluster fault tolerance whereas inter-cluster bypass with m=1 and global re-execution are implemented to provide network-wide fault tolerance.

- *Braided Multipath + Inter-Cluster Bypass (InterBp):* When a cluster member node encounters a fault, it tries to recover the fault by using a braided multipath. On the other hand, when a CH is informed or encounters a fault and it cannot recover with a braided multipath, it uses the inter-cluster bypass technique.
- *Braided Multipath + Intra-Cluster Bypass + Inter-Cluster Bypass:* This setup is same with the previous setup except that a cluster member node first tries to recover by braided multipath then it uses intra-cluster bypass.
- *Braided Multipath + Global Re-execution:* In the presence of a fault, a node first tries to recover the fault via using braided multipath. Besides, shortest hop algorithm is re-performed per 20 seconds for fault recovery. The difference between this technique and GR is that this technique provides fast recovery with a braided multipath. Otherwise, a node can wait 20 seconds for recovery and this can cause problems in emergency applications.
- *Instant Recovery + Inter-Cluster Bypass:* When a node encounters a fault, it tries to recover the fault by using braided multipath. When a CH cannot recover with braided multipath, it uses inter-cluster bypass.

- *Instant Recovery + Intra-Cluster Bypass + Inter-Cluster Bypass:* This setup is same with the previous setup except that a cluster member node first tries to recover by instant recovery then it uses intra-cluster bypass.

- *Instant Recovery + Global Re-execution:* Instant recovery is used for in-cluster fault tolerance. For global re-execution, shortest hop algorithm is re-performed for each 20 seconds.

- *Intra-Cluster Bypass + Inter-Cluster Bypass:* Only two proposed methods in this paper are implemented together.

5.2. Event Delivery Percentages

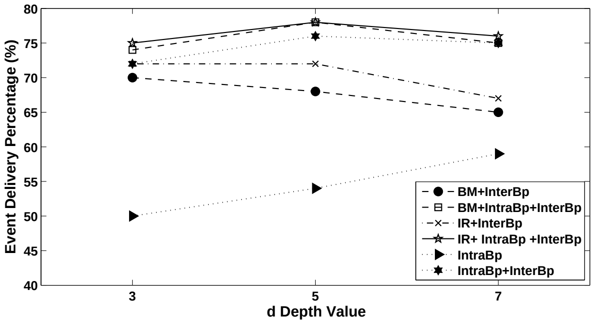


Figure 5: Affect of d cluster depth to event delivery percentage

Firstly, we measured the event delivery percentages of the algorithms against various conditions in order to obtain fault tolerance quality of them. Fig. 5 displays the affect of d cluster depth to event delivery percentage at 10% node collapse rate for 500 nodes. In fact, d parameter corresponds to the cluster size in the generic protocol because clusters and CHs are determined regarding the d parameter. According to Fig. 5, event delivery percentage of IntraBp is increasing when the cluster size is increased because the node which wants to bypass the hole in the cluster has more chance to find alternative nodes. On the other hand, performance of InterBp with BM or IR is decreasing when the cluster size increases because the number of alternative CHs for bypassing the holes decreases. When the IntraBp+InterBp with BM or IR are examined, it is shown that the results are quite different. Since cluster size has different effects on IntraBp and InterBp, the highest event

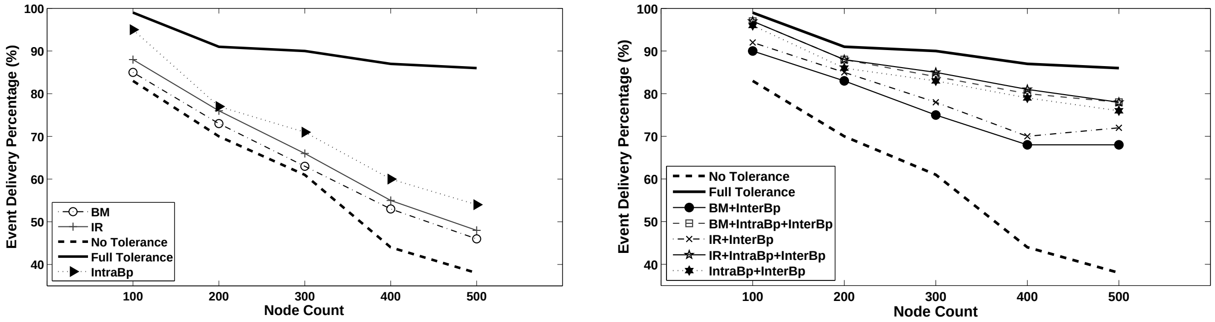


Figure 6: (a) Event delivery percentage at 10% node collapse (b) Event delivery percentage at 10% node collapse (cont.)

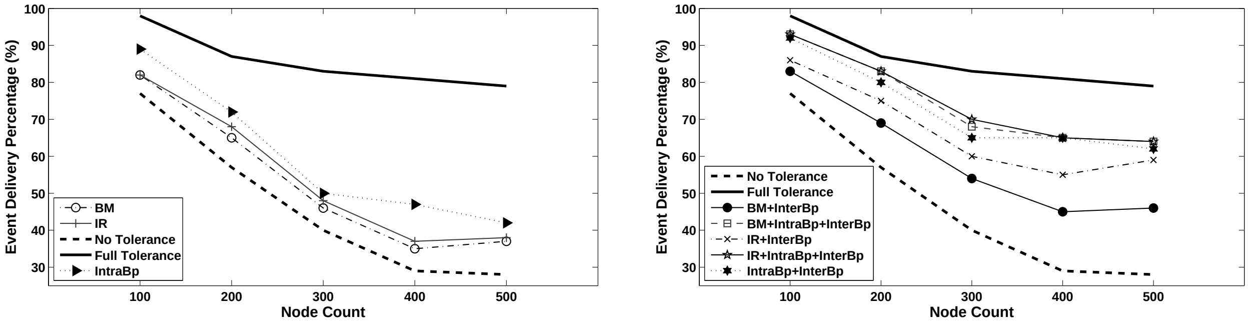


Figure 7: (a) Event delivery percentage at 20% node collapse (b) Event delivery percentage at 20% node collapse (cont.)

delivery percentage was measured in the middle of depth values. Thus, the cluster size should be selected as 5 for the case when IntraBp and InterBp are used together.

Fig. 6.a and Fig. 6.b display the event delivery percentage of fault recovery techniques at 10% node collapse percentage and 5 as cluster depth for different node counts ranging from 100 to 500 nodes. To compare these techniques with the ideal situation, we implemented an ideal fault tolerance method called *Full Tolerance*. This method heals a corrupted path if it is possible, by applying a central tree construction algorithm on the whole graph in an unrealistic manner. In Fig. 6.a and Fig. 6.b, the event delivery percentages of all fault-tolerant techniques are decreasing while the node count of network is increasing. The reason is that the size of formed holes increases with the node count in the network since a fixed percentage of the nodes have collapsed. Thus, larger hole sizes cause more disrupted communication paths. The lowest event delivery percentage which belongs to No Tolerance which is decreasing from 83% to 38% whereas the highest, Full Tolerance which is decreasing from 99% to 86%. Fig. 6.a shows us that IR and BM provide nearly 3% more event delivery percentage than No Tolerance. It is clearly shown that IntraBp achieves better performance than BM and IR, and provides from 7% to 15% performance increase regarding No Tolerance. When the InterBp is performed with BM, IR and IntraBp, event delivery percentage significantly increases as shown in Fig. 6.b. In particular, IR+IntraBp+InterBp provides 97% - 78% which is very close to the performance of Full Tolerance.

Fig. 7.a and Fig. 7.b display the event delivery percentages of fault recovery techniques at 20% node collapse percentage and 5 cluster depth for different node counts ranging from 100 to 500 nodes. Fig. 7.a shows us that the event delivery percentages of all techniques decrease compared with the results in Fig. 6.a due to increase of node collapse percentage. Fig. 7.a shows us that the event delivery percentage of No Tolerance significantly decreases compared with the results in Fig. 6.a. The event delivery percentage of No Tolerance is 77% - 28% whereas, the highest, Full Tolerance, is decreasing from 98% to 79%. BM and IR provide 4% - 5% more event delivery percentages than No Tolerance. IntraBp also achieves better performance than BM and IR at 20% node collapse percentage which provides up to 15% performance increase regarding No Tolerance. On the other hand, when the Fig. 7.b is examined, it is shown that InterBp technique has more contribution to the protocol in terms of event delivery percentage at this collapse percentage. Similarly in the Fig. 6.b, IR+IntraBp+InterBp provides the highest event delivery percentage with 93% to 64% among fault-tolerant techniques except Full Tolerance.

Fig. 8.a and Fig. 8.b display the event delivery percentage of fault recovery techniques at 30% node collapse percentage and 5 cluster depth for different node counts ranging from 100 to 500 nodes. No Tolerance has 69% -25% event delivery percentage whereas, Full Tolerance has 95% to 61%. BM and IR provides 3% - 5% more event delivery percentages than No Tolerance. IntraBp also achieves better performance than BM and IR which is 10% - 14% more than No Tolerance. When the node collapse percentage is 30%, the effectiveness of InterBp is significantly increasing. Fig. 8.b shows us that

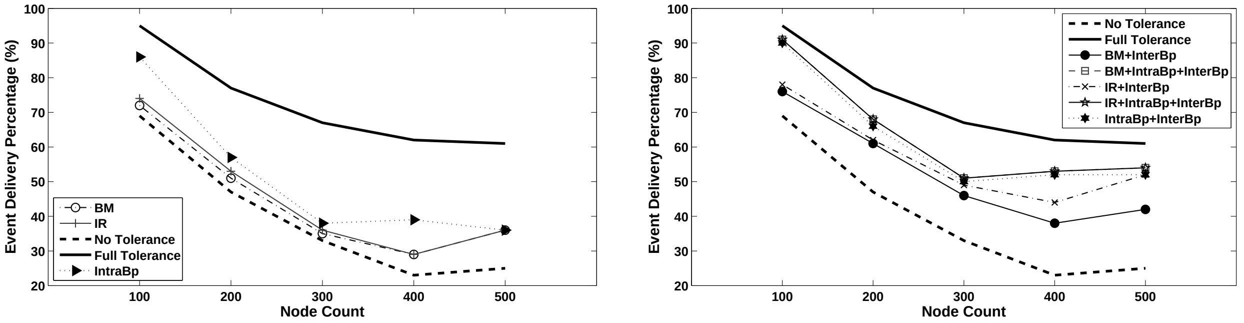


Figure 8: (a) Event delivery percentage at 30% node collapse (b) Event delivery percentage at 30% node collapse (cont.)

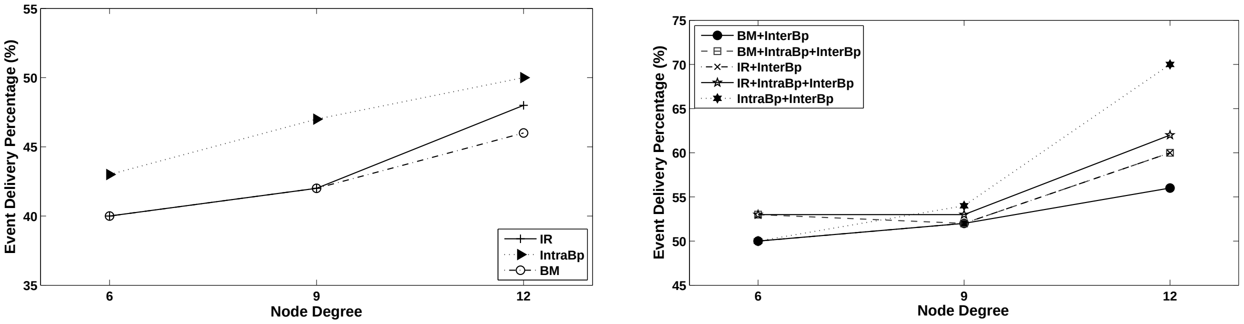


Figure 9: (a) Affect of node degree to event delivery percentage (b) Effect of node degree to event delivery percentage (cont.)

IR+IntraBp+InterBp provides the highest event delivery percentage with 91% - 54% after Full Tolerance.

Fig. 9.a and Fig. 9.b display the affect of node degree to event delivery percentage at 20% node collapse percentage and 5 cluster depth for 500 nodes. Since a node can find more alternative nodes for recovering the fault at high node degrees, event delivery of all fault-tolerant increases with the node degree. In particular, increase of IntraBp is more significant than the others. Therefore, the performance of IntraBp regarding to BM and IR increases much more while node degree is increasing. The best performance is achieved when IntraBp is used with InterBp as shown in Fig. 9.b.

The most important obtained result is that the well-known fault-tolerant techniques (IR and BM) are not capable to recover the faults formed by group collapses when the node collapse percentage increases. Since the IntraBp and InterBp techniques can reach more alternative nodes in the case of group collapses, they outperform the IR and BM in terms of event delivery percentages. Although IntraBp achieves better performance than IR and BM, it is inadequate at high node counts and collapse percentages. At this point, InterBp provides network wide fault-tolerance and reaches nearly Full Tolerance in terms of the event delivery percentage. In addition, the event delivery percentage results conform to the Theorem 5.

5.3. Energy Consumptions

Energy efficiency is an important metric for WSNs. We measured the energy consumptions of proposed algorithms, which occur mostly by message transfers, and previous approaches. In order to accurately obtain the energy consumption of fault-tolerant techniques, energy was measured for a time period starting right after the holes are formed to until each node relays an event to sink. All energy figures display the energy consumption per node when the corresponding fault-tolerant technique is performed. The measurements in the figures related to energy consumption include both the energy consumption of fault-tolerant techniques and event relaying to the sink.

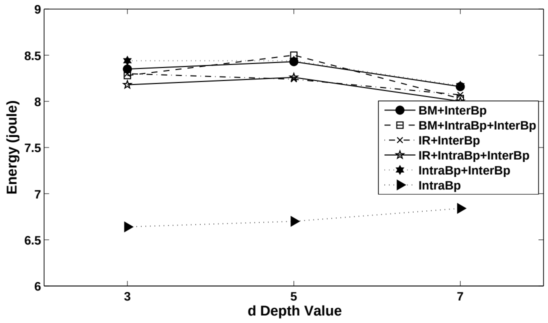


Figure 10: Affect of d cluster depth to energy consumption

Fig. 10 displays the effect of d cluster depth to energy consumption per node when the corresponding fault recovery technique is performed at 20% node collapse percentage for 500 nodes. Fig. 10 shows us that energy consumption of all fault-tolerant techniques except IntraBp increase from 3 to

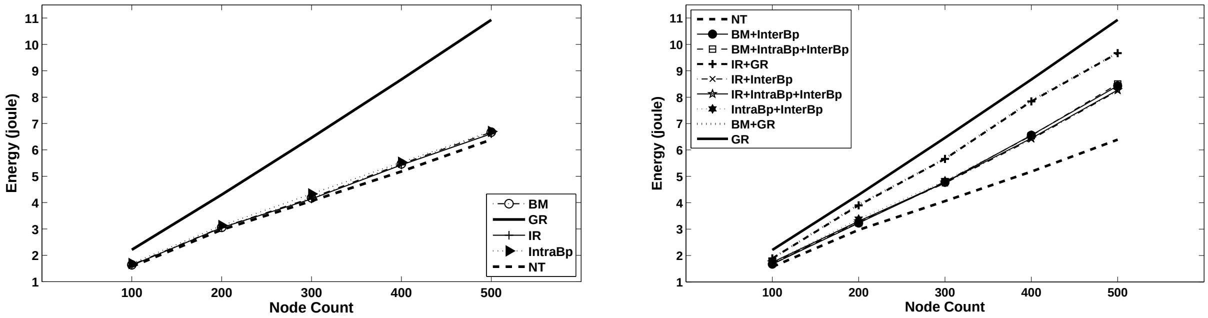


Figure 11: (a) Affect of node count to energy consumption (b) Affect of node count to energy consumption (cont.)

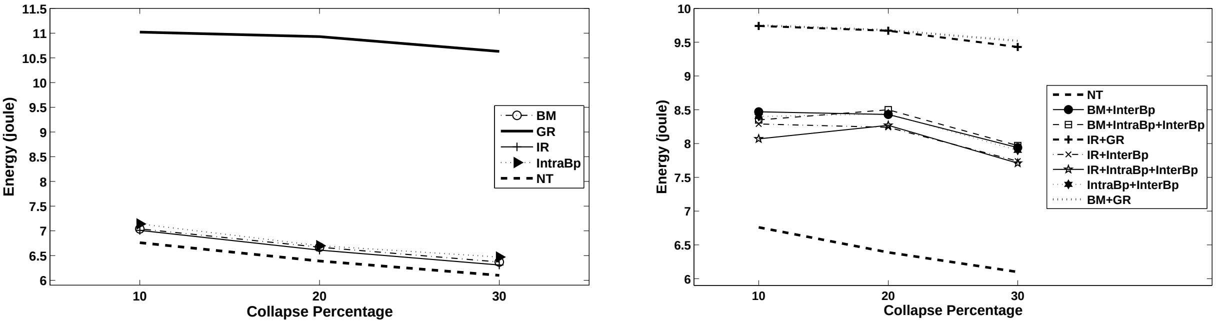


Figure 12: (a) Affect of node collapse percentage to energy consumption (b) Affect of node collapse percentage to energy consumption (cont.)

5 and then they decrease from 5 to 7 cluster depth values. These energy consumptions are related both to the event delivery percentage and cluster size. From 3 to 5 cluster depth value, both the number of delivered event and cluster size increase, thereby, the energy consumption increases. On the other hand, although the cluster depth value increases from 5 to 7, since the event delivery percentage decreases, the energy consumption also decreases. The energy consumption of IntraBp slightly increases when the cluster depth value is increased because both the event delivery percentage and the cluster size increase.

Fig. 11.a and Fig. 11.b display the effect of node count to energy consump-

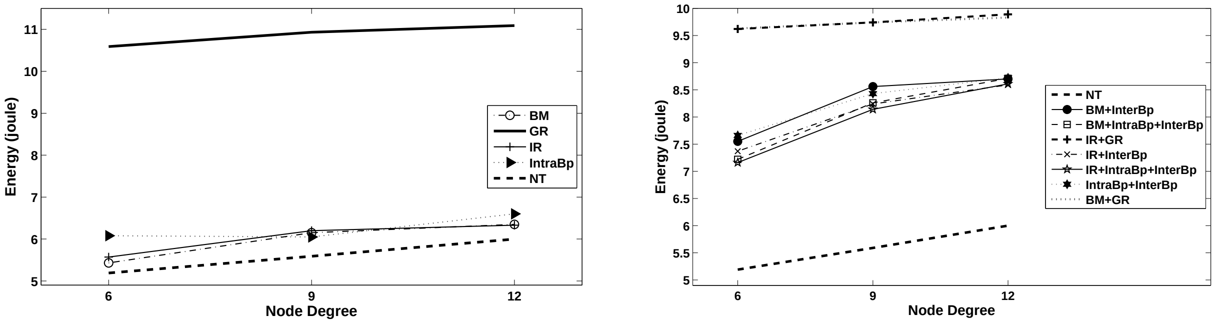


Figure 13: (a) Effect of node degree to energy consumption (b) Effect of node degree to energy consumption (cont.)

tion per node when the corresponding fault recovery technique is performed at 5 cluster depth for different node counts ranging from 100 to 500 nodes. The lowest energy consumption which belongs to No Tolerance increases from 1.57J to 6.39J whereas the highest, GR, increases from 2.21J to 10.93J. Fig. 11.a shows that BM and IR bring a few overhead in terms of energy consumption regarding No Tolerance which is between 0.05J and 0.3J. IntraBp consumes slightly more energy than BM and IR as expected because the *search* messages of the method are flooded inside the cluster. Fig. 11.b shows that since the fault-tolerant techniques are performed adaptively, the energy consumption of the techniques which are combination of InterBp consume nearly the same energy ranging from 1.72J to 8.44J. Therefore, these techniques bring from 0.25J to 2.0J overhead per node. On the other hand, BM and IR with GR consume from 1.9J to 9.68J energy and bring from 0.42J to 3.5J energy overhead per node. Therefore, although BM and IR with GR, and only GR provide the highest event delivery percentage, their energy consumption is far from applicable. In addition, techniques with GR constantly dissipate energy while the WSN is operating but, on the other hand, the other techniques consume energy only when they are performed in order to recover the faults.

Fig. 12.a and Fig. 12.b display the affect of node collapse percentage to energy consumption per node when the corresponding fault recovery technique is performed at 5 cluster depth and 500 nodes. Fig. 12.a and Fig. 12.b show that collapse percentage increase does not much affect energy consumption but, there is still a decrease when it is increased. Since the event delivery percentage is decreasing by the collapse percentage, the energy consumption of delivered messages decreases. These figures show that our proposed techniques consume far less energy than GR and consume similar energy with IR and BM.

Fig. 13.a and Fig. 13.b display the effect of node degree to energy consumption per node when the corresponding fault recovery technique is performed at 20% node collapse percentage, 5 cluster depth and 500 nodes. Energy consumption of all fault-tolerant techniques increase with node degree since a message is received by all neighbors of a sender node. We may conclude as before, for Fig. 13.a and Fig. 13.b that energy consumption performance of our techniques outperforms GR and they have similar performance with IR and BM.

From obtained measurements, we can firstly state that the energy consumptions of IntraBp and InterBp methods are scalable and stable against node count and degree. The reason for this observation is that their total message transfer is bounded by the cluster size and the count of clusters in the same level. Since IntraBp and InterBp find more alternative paths than IR and BM as shown in the previous section, they consume slightly more energy than IR and BM. On the other hand, they consume far less energy than GR because they diffuse their messages relatively to the smaller parts of the network whereas GR may cause depletion of energy in all nodes of the network.

5.4. Delays

One of the important criteria in sensor network applications is the event delivery delay, so we measure the delay values of the algorithms. The figures on delay include only the base fault-tolerant techniques except BM because it is usually formed with the paths. Since BM does not need a time for recovery, it is not needed to be displayed. Fig. 14.a displays the effect of d cluster depth to delay of fault recovery techniques at 20% node collapse percentage for 500 nodes. Fig. 14.a shows us that IR is not affected by the cluster size, whereas IntraBp and InterBp are affected by cluster size. Since the cluster size increases while the depth value is increasing, IntraBp spends more time in order to find the clusterhead. Similarly, InterBp spends more time, since *search* messages disseminate over more number of nodes and the clusterhead count decreases. The delay increase in IntraBp and InterBp is linear so we may claim that the proposed approaches are scalable in terms of delay against varying depth values.

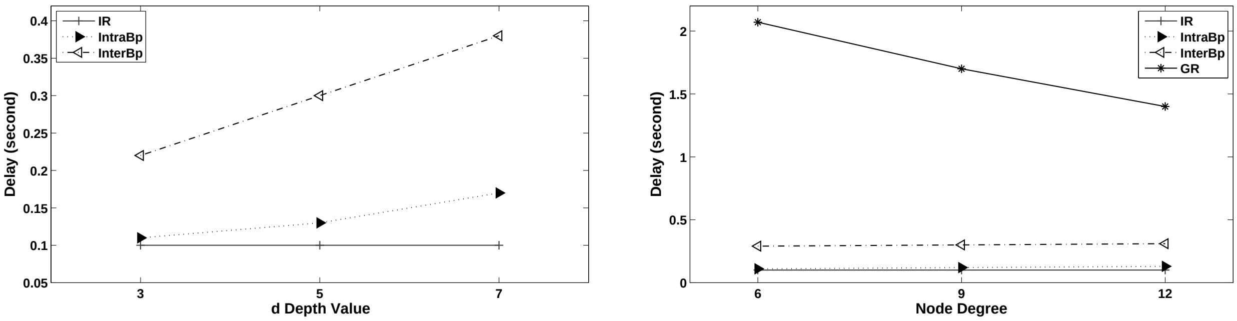


Figure 14: (a) Affect of d cluster depth to delay (b) Affect of node degree to delay

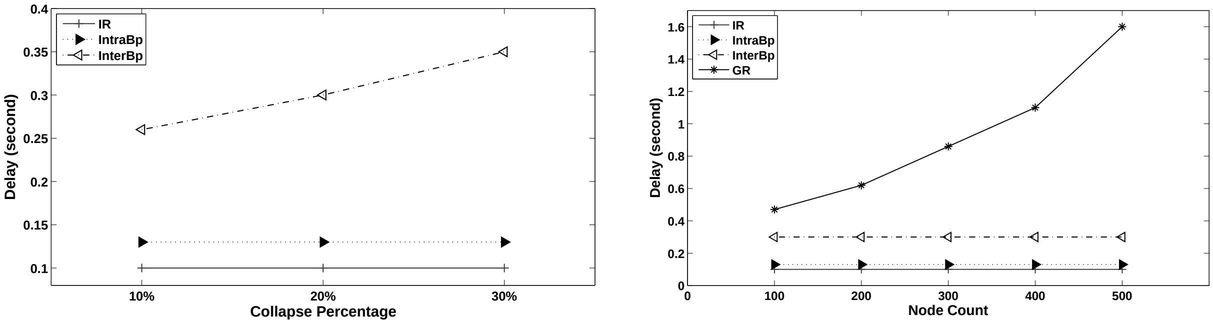


Figure 15: (a) Affect of node collapse percentage to delay (b) Delay of fault-tolerant techniques at different node counts

Fig. 14.b displays the effect of node degree to delay of fault recovery techniques at 20% node collapse percentage for 500 nodes. The delay of IntraBp and InterBp are not affected by node degree as shown in Fig. 14.b. On the other hand, time consumption of GR is very high especially for the sparsely connected networks. Fig. 15.a displays the effect of node collapse percentage to delay of fault recovery techniques at 20% node collapse percentage for 500 nodes. In Fig. 15.a, the delay of IR and IntraBp is stable against varying node collapse percentages because these techniques are performed in a designated area. On the other hand, delay of InterBp is increasing while node collapse percentage increases because probability of finding a close alternative cluster is decreasing with the node collapse percentage.

Fig. 15.b displays delay of fault recovery techniques for different node counts ranging from 100 to 500 nodes. Node count metric shows the scalability of fault-tolerant techniques for different node counts. Fig. 15.b shows us that the delays of IR, IntraBp and InterBp are nearly constant against node count whereas the delay of GR is increasing with the node count since GR needs network-wide execution.

The results measured in this section show us that IntraBp and InterBp are scalable in terms of event delivery delay against varying node degrees, node counts, cluster counts and cluster sizes which conforms to Theorem 2 and Theorem 4. The time consumptions of the IntraBp are slightly more than those of IR since all cluster links are regenerated in IntraBp. InterBp performs far more better than GR because when GR technique is used, whole network has to wait since all the paths are regenerated. But, InterBp is only performed only in a part of network and other nodes can continue to relay messages to sink.

5.5. Performance Evaluations on CPEQ Protocol

In this simulation study, we extended the CPEQ protocol with proposed InterBp fault-tolerant technique. Then, we compare the CPEQ protocol with the extended version of it which is denoted by CPEQ+InterBp. CPEQ does not need IntraBp because clusters are periodically re-formed. With this simulation study, we show that how these proposed techniques can be easily be inserted into any clusterbased routing protocol in literature.

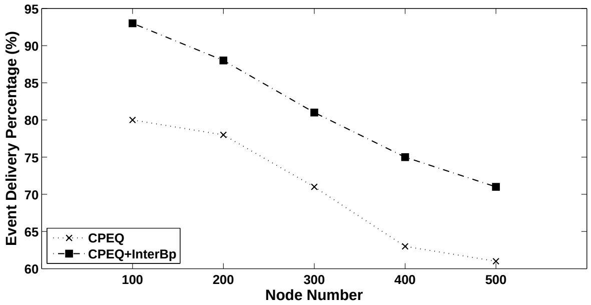


Figure 16: Event delivery percentage comparison of CPEQ and CPEQ+InterBp at 10% collapse

CPEQ is a periodic, event-driven and query-based protocol which includes fault-tolerant and low-latency algorithms that meet sensor network

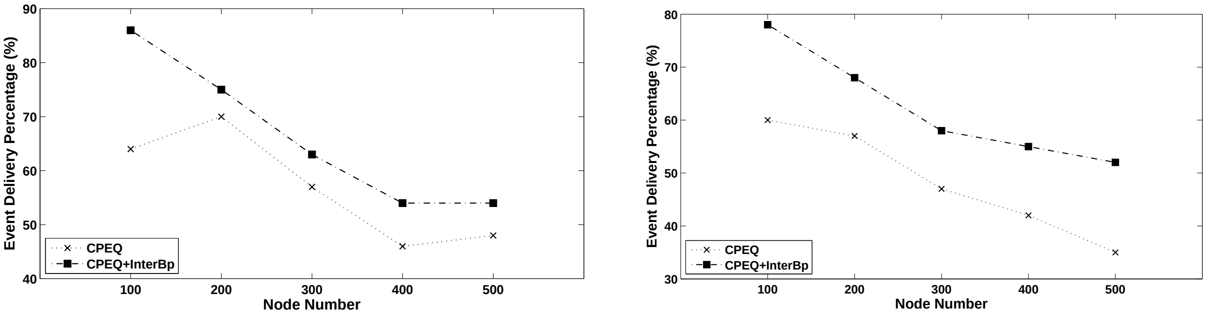


Figure 17: (a) Event delivery percentage comparison of CPEQ and CPEQ+InterBp at 20% collapse (b) Event delivery percentage comparison of CPEQ and CPEQ+InterBp at 30% collapse

requirements for critical conditions. It is a cluster-based routing protocol that groups sensor nodes to efficiently relay the sensed data to the sink by uniformly distributing energy dissipation among the nodes. CPEQ uses the publish/subscribe paradigm to disseminate requests across the network.

CPEQ starts with building the hop tree which is a flooding based algorithm. This algorithm is similar to the shortest hop algorithm used in Generic Protocol. The only difference is that the algorithm in CPEQ does not construct any path towards sink. Its aim is to configure nodes in the network by assigning hop levels. Then, the sinks in the network subscribe to the nodes by disseminating subscription messages. While the subscription message is disseminated in the network, paths towards sink are constructed. Each node maintains a subscription table and it forwards the events by looking at this table.

Clusters are re-formed per fixed time period in CPEQ protocol. The clustering algorithm starts by designating aggregator nodes. In order to designate the aggregator nodes at each clustering period, each node generates a random number between 0 and 1. The nodes which generate a number lower than 0.05 will request the energy level from its immediate neighbors. The nodes then select the neighbor with more energy as aggregator and inform it. The newly selected aggregator node is responsible for forming the cluster. The cluster configuration is performed through the broadcasting of a notification packet. In order to limit the size of a cluster, the notification packet carries a time to live (ttl) field. We set the value of ttl value as 3 in our simulations.

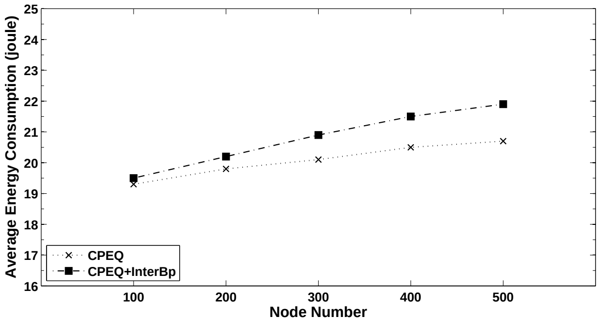


Figure 18: Energy consumption (per node) comparison of CPEQ and CPEQ+InterBp during 10% collapse

When a node detects an event in the environment, the sensed data is sent to the cluster aggregator node. Then, the aggregator node relays the event towards the sink using the path constructed in the subscription phase. Timer and acknowledgement mechanisms are enabled after a message is transmitted for reliable communication. The percentage of aggregator nodes is selected as 5 % in our simulations.

Although CPEQ protocol includes fault-tolerant and low-latency algorithms, its effectiveness decreases in the presence of holes caused by group collapses because it uses instant recovery in the protocol. Therefore, we extended the CPEQ protocol by inserting the proposed techniques in this paper. Fig. 16, Fig. 17.a and Fig. 17.b display the event delivery percentage comparison of CPEQ and CPEQ+InterBp during 10%, 20% and 30% node collapse percentages respectively. Three figures definitely show that InterBp fault-tolerant technique increases the event delivery percentage of CPEQ protocol. InterBp provides 7%-18% event delivery percentage increase in each node collapse percentage. Fig. 18, Fig. 19.a and Fig. 19.b display the energy consumption of per node comparison of CPEQ and CPEQ+InterBp during 10%, 20% and 30% node collapse percentages respectively.

These figures show us that InterBp technique increases the energy consumption per node 4% in the worst case, 2% in the average case. In fact, since CPEQ+InterBp recovers more fault and provides significant increase in event delivery, more events are relayed towards sink. Therefore, in order to relay more events, more energy is consumed. However the consumed energy is not significant when compared to the increase in event delivery. The reason

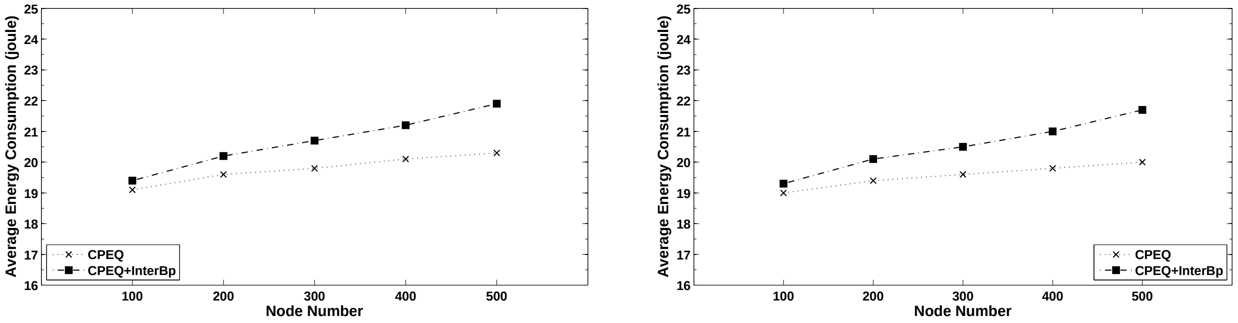


Figure 19: (a) Energy consumption (per node) comparison of CPEQ and CPEQ+InterBp during 20% collapse (b) Energy consumption (per node) comparison of CPEQ and CPEQ+InterBp during 30% collapse

of this observation is InterBp tries to reconstruct a new path only within a few levels of clusters and reconstructs a new path immediately if it is found.

6. Conclusions

In this paper, we first model holes as clusters to convert the problem of hole bypassing to cluster bypassing. Intra-cluster and Inter-cluster bypass algorithms are proposed for bypassing the holes inside the cluster and holes which cover more than one cluster. The algorithms are analyzed, evaluated with extensive simulations and compared with well-known fault-tolerant algorithms. In the performance evaluations, two simulation studies over a generic protocol and CPEQ [21] have been conducted in order to show the effectiveness and applicability of algorithms, and how the algorithms increase the event delivery percentage of CPEQ. We also showed from the simulations that the event delivery percentages and resource usages of proposed methods conform to the theoretical analysis.

Although fault-tolerant routing techniques has been studied in literature, localization-free and energy-efficient hole bypassing has not been studied so far. Disjoint and braided multipaths, and instant recovery algorithms aim to recovery the isolated failures. Unfortunately, these algorithms cannot recover the faults occurring by group collapses. [9, 10, 11, 12, 13, 14] proposed algorithms and protocols for bypassing the holes which are formed by group collapses, deployment etc. But, the algorithms in these studies use the localization techniques. Therefore, they are not applicable to many WSN applications.

The simulations results show that Intra-cluster and Inter-cluster bypass algorithms provide very close performance to Full Tolerance in terms of fault-tolerance while they consume significantly lower energy than Full Tolerance. When they are compared with the well-known fault-tolerant techniques, they provide up to 25% better fault-tolerance as well as reasonable delay and energy consumptions.

In addition, Intra-cluster and Inter-cluster bypass algorithms can be easily integrated into many protocols in literature. Since they can be performed adaptively, they increase the fault-tolerance with very low energy consumption. In the simulation study on CPEQ protocol, it is obviously seen that the Inter-cluster bypass algorithm increases the event delivery percentage of CPEQ protocol with insignificant energy consumption.

References

[1] M. Tubaishat, S. Madria, Sensor networks: an overview, Potentials, IEEE 22 (2) (2003) 20–23.

[2] L. D. Mendes, J. J. Rodrigues, A survey on cross-layer solutions for wireless sensor networks, Journal of Network and Computer Applications 34 (2) (2011) 523 – 534, efficient and Robust Security and Services of Wireless Mesh Networks.

[3] K. Lin, Research on adaptive target tracking in vehicle sensor networks, Journal of Network and Computer Applications (0) (2012) –.

[4] A. Hadjidj, M. Souil, A. Bouabdallah, Y. Challal, H. Owen, Wireless sensor networks for rehabilitation applications: Challenges and opportunities, Journal of Network and Computer Applications 36 (1) (2013) 1 – 15.

[5] R. P. Pantoni, D. Brando, A gradient based routing scheme for street lighting wireless sensor networks, Journal of Network and Computer Applications 36 (1) (2013) 77 – 90.

[6] H. Dai, Z. min Zhu, X.-F. Gu, Multi-target indoor localization and tracking on video monitoring system in a wireless sensor network, Journal of Network and Computer Applications 36 (1) (2013) 228 – 234.

[7] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Comput. Netw. 52 (2008) 2292–2330.

[8] L. Paradis, Q. Han, A survey of fault management in wireless sensor networks, J. Netw. Syst. Manage. 15 (2007) 171–190.

[9] Q. Fang, J. Gao, L. J. Guibas, Locating and bypassing holes in sensor networks, Mob. Netw. Appl. 11.

[10] F. Yu, E. lee, Y. Choi, S. Park, D. Lee, Y. tian, S.-H. Kim, A modeling for hole problem in wireless sensor networks, in: Proceedings of the 2007 international conference on Wireless communications and mobile computing, IWCMC '07, ACM, New York, NY, USA, 2007, pp. 370–375.

[11] F. Yu, S. Park, Y. Tian, M. Jin, S.-H. Kim, Efficient hole detour scheme for geographic routing in wireless sensor networks, in: Proceedings of the 67th IEEE Vehicular Technology Conference, VTC Spring 2008, 11-14 May 2008, Singapore, IEEE, 2008, pp. 153–157.

[12] F. Yu, S. Park, E. Lee, S.-H. Kim, Hole modeling and detour scheme for geographic routing in wireless sensor networks, Journal of Communications and Networks (2009) 327–336.

[13] J. You, D. Lieckfeldt, F. Reichenbach, D. Timmermann, Context-aware geographic routing for sensor networks with routing holes, in: Proceedings of the 2009 IEEE conference on Wireless Communications & Networking Conference, WCNC'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 2589–2594.

[14] H. Shiow-Fen, Y. Chia-Hsuan, S. Yi-Yu, D. Chyi-Ren, Energy efficient hole bypassing routing in wireless sensor networks, in: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, ICCSIT'10, IEEE Press, Piscataway, NJ, USA, 2010, pp. 576 – 580.

[15] W. R. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99, ACM, New York, NY, USA, 1999, pp. 174–185.

[16] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, Directed diffusion for wireless sensor networking, IEEE/ACM Trans. Netw. 11 (2003) 2–16.

[17] D. Braginsky, D. Estrin, Rumor routing algorthim for sensor networks, in: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02, ACM, New York, NY, USA, 2002, pp. 22–31.

[18] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8, HICSS '00, IEEE Computer Society, Washington, DC, USA, 2000, pp. 8020–.

[19] A. Manjeshwar, D. P. Agrawal, Teen: A routing protocol for enhanced efficiency in wireless sensor networks, Parallel and Distributed Processing Symposium, International 3 (2001) 30189a+.

[20] A. Manjeshwar, D. Agrawal, Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in: Proceedings 16th International Parallel and Distributed Processing Symposium, Vol. 13, IEEE Comput. Soc, 2002, p. 195.

[21] A. Boukerche, R. Werner Nelem Pazzi, R. Borges Araujo, Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments, J. Parallel Distrib. Comput. 66 (2006) 586–599.

[22] A. Boukerche, A. Martirosyan, R. Pazzi, An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks, Mob. Netw. Appl. 13 (2008) 614–626.

[23] A. Boukerche, I. Chatzigiannakis, S. Nikoletseas, A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range, Comput. Commun. 29 (2006) 477–489.

[24] I. Chatzigiannakis, A. Kinalis, S. Nikoletseas, Fault-tolerant and efficient data propagation in wireless sensor networks using local, additional network information, J. Parallel Distrib. Comput. 67 (2007) 456–473.

[25] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, Highly-resilient, energy-efficient multipath routing in wireless sensor networks, in: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '01, ACM, New York, NY, USA, 2001, pp. 251–254.

[26] H. Karl, A. Willig, Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons, 2005.

[27] O. Dagdeviren, K. Erciyes, Graph matching-based distributed clustering and backbone formation algorithms for sensor networks, Comput. J. 53 (2010) 1553–1575.

[28] K. Akkaya, M. F. Younis, A survey on routing protocols for wireless sensor networks., Ad Hoc Networks (2005) 325–349.

[29] K. Erciyes, D. Ozsoyeller, O. Dagdeviren, Distributed algorithms to form cluster based spanning trees in wireless sensor networks, in: Proceedings of the 8th international conference on Computational Science, Part I, ICCS '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 519–528.

[30] D. Wagner, R. Wattenhofer, Algorithms for sensor and ad hoc networks: advanced lectures, Springer-Verlag, Berlin, Heidelberg, 2007.

[31] D. Peleg, Distributed computing: a locality-sensitive approach, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.