We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Assessing the Vulnerabilities of Mission-Critical Wireless Sensor Networks

Mohamed M. A. Azim * and Aly M. Al-Semary †

Additional information is available at the end of the chapter

1. Introduction

Wireless Sensor networks (WSNs) have recently received increased attractiveness driven by many mission-critical applications such as battlefield reconnaissance and homeland security monitoring. Mission critical here refers to networking for application domains whose infrastructure and operations are absolutely necessary for an organization to carry out its mission [1]. Therefore, the main feature that must be guaranteed by all networks running mission critical applications is the network continuity.

However, due to the nature of the deployment field, these networks are vulnerable to natural disasters such as earthquakes, tornadoes or floods. Moreover, they are also subject to physical attacks such as an Electro-Magnetic Pulse (EMP) attack and security breaches such as sinkhole and selective forwarding attacks [2]. Such real world events may happen in particular geographical areas and disrupt specific parts of the network. Therefore, the geographical layout of the network topology determines the impact of such events on the network's connectivity.

Several contributions in the literature have addressed the failure modeling and survivability problems. The authors in [3, 4] tackled the single link failure problem in the logical topology. The authors in [5] focus on the dual link failure assumption. Most of these studies are based on a common assumption that failures are independent of their locations and randomly distributed across the network, which fails to reflect several real scenarios. Such real-world events have geographical nature, and therefore, the geographical structure of the network affects the impact of these events. Under such region failure scenarios, several network components within a geographically correlated region may be simultaneously destroyed,

*Author's permanent address: Faculty of Industrial Education, Beni-Suef University, Egypt
†Author's permanent address: Systems and Comp. Eng. Dept., Faculty of Eng., Al-Azhar University, Cairo, Egypt. This research is supported by the deanship of scientific research, Taibah University under Grant 717/431.

INTECH
open science | open minds

resulting in network holes, cuts (partitions) or even breakdown of the overall network connectivity as shown in Fig. 1. Therefore, it is essential to assess the vulnerabilities of mission critical networks to such region-failures.

Some research has been conducted to understand the impact of region failures on wired backbone networks such as [11–18]. On the other hand, the cut detection problem has been investigated by [19] and [20].

Recently, few studies have tackled the region-failure problem in wireless networks. The authors in [7–9] investigated the region- based connectivity issue in wireless networks and demonstrated the effect of the transmitting power on maintaining a region-based connectivity in the presence of single and multiple region failures. The authors in [10] proposed a more general Probabilistic Region Failure (PRF) Model to capture the key features of geographically correlated region failures. They also developed a framework to apply the PRF model for the reliability assessment of wireless mesh networks.
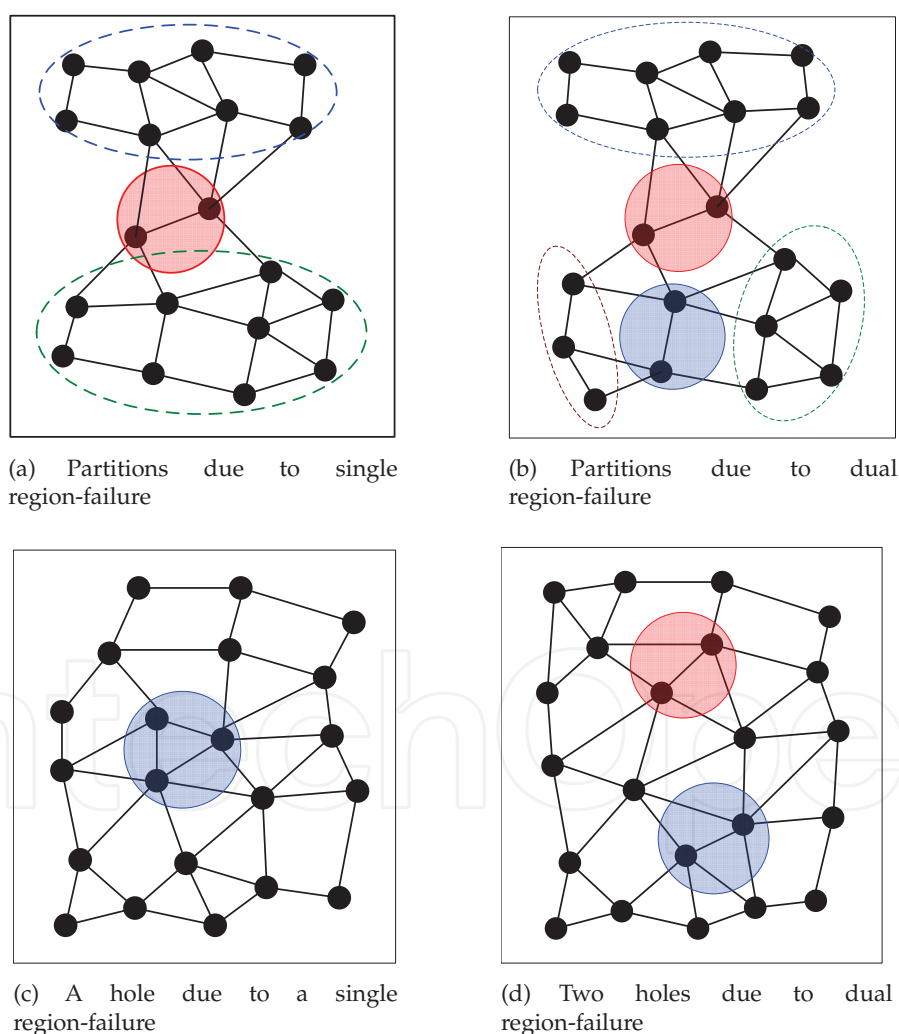

(a) Partitions due to single region-failure

(b) Partitions due to dual region-failure

(c) A hole due to a single region-failure

(d) Two holes due to dual region-failure

Figure 1. Example of Network Partitions and holes

All of the aforementioned studies about regional-failures consider a worst-case cut as the cut which maximizes or minimizes certain performance metric (such as capacity) of the

intersected links. However, such a definition is inadequate to capture many realistic situations where the faulty nodes may influence larger number of nodes rather than larger number of link-cuts. we claim that, using the number of failed links as the main criteria for defining the worst-case region cut underestimates the impact of a region failure on the overall network performance.

Therefore, in this chapter, we first introduce a new definition for for a worst-case cut (partition) due to failure regions. Then, we identify the location of a disaster that would have the maximum impact on a test network using both definitions. Finally, we conduct a deeper analysis to understand the behavior of a single and dual region-failures on the network performance with mission-critical nodes. Our simulation results indicate that, current studies in regional failures under estimate the impact of the worst-case cut due to their dependence on a relaxed definition for the worst-case region-cut.

The rest of this chapter is organized as follows. Section 2 presents the problem investigation. Section 3 demonstrates our proposed scheme for identifying the worst-case cut under single and dual region-failures. Section 4 presents our experimental results. Finally, Section 5 concludes our chapter.

2. Problem investigation

Most of the available studies [21–23] in the literature consider that the failure probability of a node is independent of its location in the deployment area. Few studies [6–10] addressed the region failure problem of spatially correlated network nodes in the physical topology. Available studies focusing on region failures consider link-cuts due to a region-failure as the main criteria for identifying the worst-case region-failure and can be interpreted in terms of the network capacity and throughput. However, such fault scenario is inadequate to capture many realistic situations where the failure region may influence larger number of nodes rather than larger number of link-cuts as shown in Fig.2.
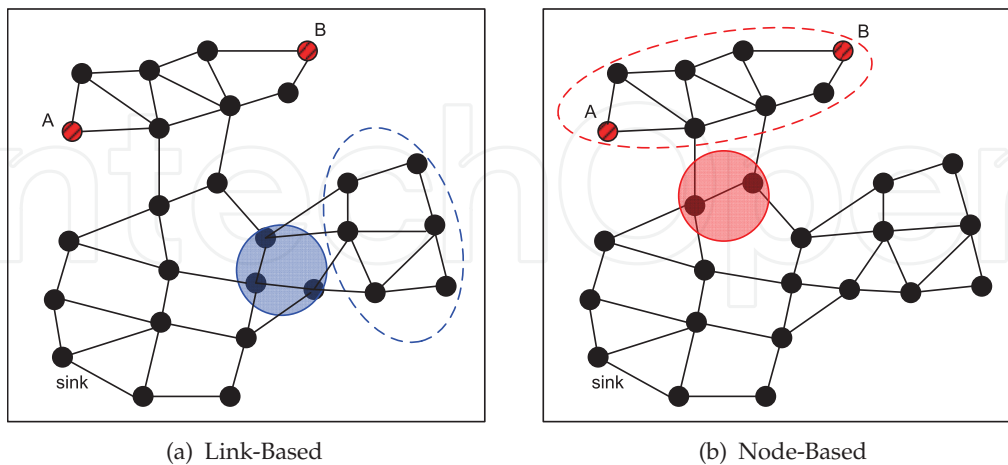

(a) Link-Based   (b) Node-Based

Figure 2. Selection of the Worst-case region cut

For the network shown in Fig. 2, suppose that all links have the same capacity. Then, it can be easily seen that the region failure in Fig. 2a leads to 10 link-cuts while the region failure shown in Fig. 2b leads to only 6 link-cuts. Based on the current studies, the region with the

maximum number of link-cuts is identified as the worst-case cut (i.e the region in Fig. 2a). On the other hand, the failure region depicted in Fig. 2b disconnects the dashed region (8 nodes) from the rest of the network which has a greater impact on the network than the case shown in Fig. 2a which has only 6 disconnected nodes.

Therefore, we first propose a new model for the worst-case region-cut considering disconnected nodes (Node-Based) due to a region-failure as the major criteria for identifying the worst-case region-cut. In our proposed model, we may select a region with less number of link-cuts as the worst-case cut if it isolates a larger number of nodes. Moreover, in our proposed model mission-critical nodes are given more weight during the worst-case analysis process to reflect the importance of its service continuity.

3. Proposed model

The network model we consider here is composed of a base station, a sink node, and a set of distributed wireless sensor nodes. The base station can be inserted in any suitable place whether in the field or somewhere else. It is directly connected to the sink node through a wire or wireless link. The sink node is a wireless sensor with high capability in memory, processing, power, and wireless coverage. It works as an intermediate node between the base station and the other sensors. It receives commands from the base station and then conveys them to the deployed sensors. In addition, it collects data from the sensors and sends it to the base station. The other sensor nodes are categorized into mission-critical (MC) sensor nodes and regular sensor nodes. A mission-critical sensor node is a node that is responsible for sensing or reading mission-critical information such as a sensor node in the battle field. while the regular sensor node is any other node. These sensors which have limited capabilities in their battery-powers, memory, and processing are distributed all over the area of interest in such a way that any deployed node has at least one path to the sink node.

In the rest of this section we present the problem formulation of our proposed model in section 3.1. Then, in section 3.2 we introduce the routing process based on the above mentioned network model. Finally, we demonstrate the region failure analysis under single and dual region-failures in section 3.3.

3.1. Routing process

Nodes in a routing table are classified into three categories: 1) parent node, 2) sibling node and 3) child node. A parent node is a node in the transmission range of another sending node and having a hop count one less than the sending node. A sibling node is a node in the transmission range of another sending node and having the same hop count as the sending node. A child node is a node in the transmission range of another sending node and having a hop count one more than the sending node. After deploying sensor nodes into the network field, the routing tables of the underlying sensor nodes are established. To illustrate the process, a network is constructed in Fig. 3. The network has eight sensors: a sink node, two mission-critical sensors (A and F), and five regular sensors (B, C, D, E, and G).

The routing process is started when the sink node broadcasts a setup packet to all nodes within its transmission range. The setup packet contains several parameters including the number
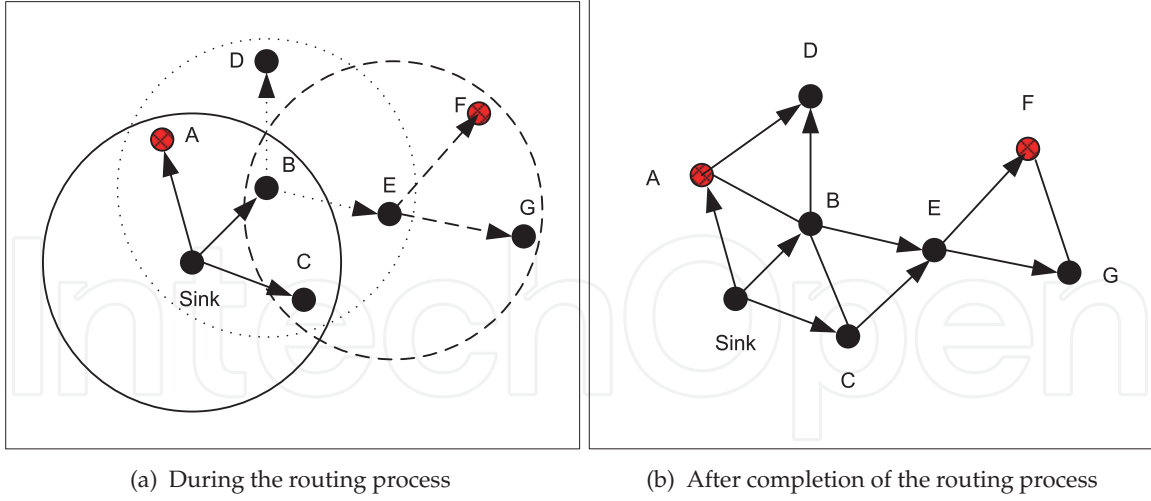
(a) During the routing process   (b) After completion of the routing process

Figure 3. Routing process configuration

of hops h between the sending node and the sink node. Each sensor node receiving the setup packet from the sink node sets its value of h to the incremented value of the received h and marks the sink node as a parent node. For the network shown in Fig. 3a, each of the nodes A, B, and C in the solid circle marks the sink node as its parent. Then, every node with hop count of 1 broadcasts a newly constructed setup packet. For example, suppose that the node B sends its setup packet. In this case, the sink node (belongs to the third category), the sensor nodes A and C (belong to the second category), and the sensor nodes D and E (belong to the first category) mark the node B as a child, sibling, and parent, respectively. Each node in the first category (in this case, D and E) sets its value of h to 2 (i.e., the value of received $h + 1$). Each of the nodes D and E constructs its setup packets and broadcasts it to its neighbors. This broadcasting process is repeated hop-by-hop by the nodes of the first category until all the deployed sensor nodes establish their routing tables. This will end up with the Fig. 3b, where a headed-arrow represents a link between a parent node and a child and it starts from the parent and heads into its child. A line with no arrow-head refers to a link between two sibling nodes.

3.2. Problem formulation

When considering circular cuts, we assume that a disaster results in a cut of radius r, which is centered at [x,y]. We define the worst-case region-cut of a network topology nt as $WCC(nt)$ which can be evaluated as follows.

$$WCC(nt) = \max_{\forall x,y}(rw_{x,y}) \tag{1}$$

Where $rw_{x,y}$ is the weight function of the region-cut centered at (x,y) and can be evaluated as follows.

$$rw_{x,y} = c \cdot \underbrace{\sum_{s_i \in ds_{x,y}} nw(s_i)}_{\text{Disconnected nodes}} + \underbrace{\sum_{s_j \in fs_{x,y}} pw(s_j)}_{\text{Failed nodes}} + \underbrace{\sum_{b_i \in br_{x,y}} bw(b_i)}_{\text{Failed links}} \tag{2}$$

Where c is a constant that gives the disconnected nodes higher weight than link-cuts, $ds_{x,y}$ is the set of disconnected sensor nodes due to the region-cut centered at (x,y), and $nw(s_i)$ is the weight of the sensor node s_i that differentiates between a regular node and a mission-critical node.

$fs_{x,y}$ is the set of failed sensor nodes within the region-failure centered at (x,y), and $pw(s_j)$ is the weight of the sensor node s_j on the path from a source node S to the sink node. The $pw(s_j)$ is used to distinguish between a node on a regular path or a mission-critical path. $br_{x,y}$ is the set of paths (bridges) connecting a disconnected network partition (due to the region-cut centered at (x,y)) with a connected network section. $bw(b_i)$ is the weight of the bridge b_i.

Equation 1 indicates that, the worst-case cut can be located by evaluating a weight function $rw_{x,y}$ for each region cut and the region-cut with maximum weight is considered the worst-case region-cut.

Equation 2 evaluates the weight of a region cut centered at [x,y] by estimating the importance of three main factors: disconnected nodes, failed nodes, and link-cuts as described in Eq. 2.

The first term $\sum_{s_i \in ds_{x,y}} nw(s_i)$ is the summation of the weights of all disconnected nodes due to the region-failure. Through the node weight $nw(s_i)$, we provide more weight to mission-critical nodes than regular nodes as indicated by Equation 3.

$$nw(s_i) = \begin{cases} 1 & \text{if } s_i \in RS, \\ 2^c & \text{if } s_i \in MS. \end{cases} \tag{3}$$

Where RS is the set of all regular sensor nodes in the network field and MS is the set of all mission-critical sensor nodes.

The second term $\sum_{s_j \in rs_{x,y}} pw(s_j)$ represents the summation of all path weights of the failed nodes (all nodes within the region cut).

This weight function $pw(s_j)$ is used to grant higher weights to the failed nodes within the region failure that are used in forwarding mission-critical information as described in Eq. 4.

$$pw(s_j) = \begin{cases} 0 & \text{if } s_j \in RS \text{ \& } s_j \notin MS, \\ 2^c & \text{if } s_j \in MS, \\ \sum_{s_k \in SC(s_j)} \frac{pw(s_k)}{pn_k} & \text{if } s_j \in RS \text{ \& } s_j \in MP. \end{cases} \tag{4}$$

Where MP is the set of mission-paths which we refer to as paths carrying mission-critical information, SC is set of children of the sensor s_j and pn_k is the number of parents of the sensor s_k.

In Eq.4, the path weight $pw(s_j)$ is assigned a value of zero if a failed node is a regular node that is not located on a mission-path (i.e., it is not used to forward mission-critical information). on the other hand, if the failed node is mission-critical node it receives higher path weight. Finally, if the failed node s_j is a regular node that is located on a mission-critical path, then the path weight is evaluated recursively by the summation of the ratio of path weights $pw(s_k)$ of the child node k to the number of parent nodes pn_k of the child node k.

The last term $bw(b_i)$ is the weight of the bridge b_i. This weight function reflects the importance of a failed link due to a region-cut. Here, a failed link can participate in more than one bridge. Hence, we are interested in calculating the the For example, if all links on the failed bridge are not located on mission-path it is given a weight of 1.

on the other hand,if all links on the failed bridge are located on mission-path it is given the highest weight.

Finally, if not all of the links on the failed bridge are located on a mission-path, its weight is a percentage of the highest weight. This percentage can be evaluated by the ratio of the number of links carrying mission-critical information of the bridge ML_{b_i} to its total number of links TL_{b_i} as described in Equation 5

$$
bw(b_i) = \begin{cases} 1 & \text{if all sensors in } b_i \notin MP, \\ c & \text{if all sensors in } b_i \in MP, \\ \frac{ML_{b_i}}{TL_{b_i}} \cdot c & \text{otherwise.} \end{cases} \tag{5}
$$

where b_i is a linking path (bridge) connecting a disconnected node with a connected nodes and passing through failed nodes. Each of the connected and disconnected nodes should have a direct link with a failed node. The bridge direction should be from a child toward a parent (i.e. any bridge linking two sibling nodes is excluded). ML_{b_i} is the number of mission-critical links located on the bridge b_i, and TL_{b_i} is the total number of links of the bridge b_i.

To further illustrate the evaluation of the worst-case region-failure, we provide the following example shown in Fig. 4. In this example compare two different region failures as shown in Fig. 4c and Fig. 4d. Herefater, we also provide the corresponding analytical calculations.

For the network shown in Fig. 4a, to find the worst-case region-failure we have to apply the routing process discussed earlier to configure the routing tables by the category (parent, sibling, child) of each neighbor. The result of routing phase for the given network topology is depicted in Fig. 4b. As described earlier in this section, the headed-arrow represents a child node toward the head and a parent node toward the tail while a line without heads represents a sibling relationship. Hereafter in the following example, we assume that the value of the constant c is equal to 5.

For the failure $Region1$ shown in Fig. 4c, to find the worst-case region-cut, we apply Eq. 2 of our proposed model to get the weight of this region-failure as follows. For the calculation of the node weight $nw(s_i)$ based on Eq. 3, each node in the disconnected partition of the topology is assigned a value of 1. Hence, $\sum_{s_i \in ds_{Region1}} nw(s_i) = 6$. Note that, there is no mission-critical nodes in the disconnected partition. For the calculation of the path weight $pw(s_j)$ we use Eq. 4. As all the failed nodes within the region-failure are not carrying mission critical information, then $\sum_{s_j \in fs_{Region1}} pw(s_j) = 0$. Finally, for the bridge weight $bw(b_i)$ we use Eq. 5. Here we have 6 different paths (bridges) crossing the region-failure and none of these bridges is carrying mission-critical informations. Hence, each bridge will be given a weight of 1. Consequently, $\sum_{b_i \in br_{x,y}} bw(b_i) = 6$. Finally, since the constant c is assumed to have the value of 5, we have $rw_{Region1} = 5 * 6 + 0 + 6 = 36$. Note that, any bridge that has a link connecting two sibling nodes will be execluded.
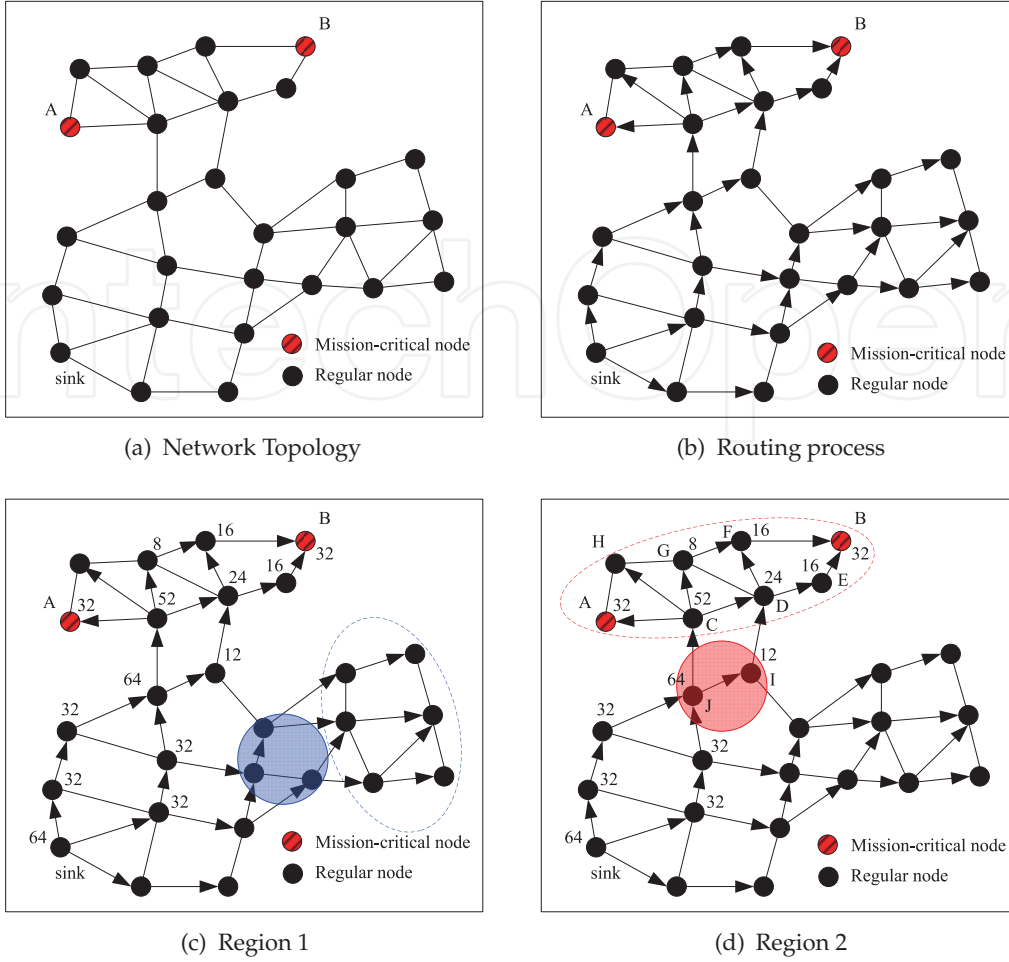
(a) Network Topology
(b) Routing process
(c) Region 1
(d) Region 2

Figure 4. Calculations of the worst-case region-cut

Similarly, For the failure $Region2$ shown in Fig. 4d, we apply Eq. 2 to get the weight of this region-failure as follows. For the calculation of the node weight $nw(s_i)$ based on Eq. 3, each node in the disconnected partition of the topology is assigned a value of 1 except each of the mission-critical nodes A and B will have the value of 32. Hence, $\sum_{s_i \in ds_{x,y}} nw(s_i) = 64 + 6 = 70$.

For the calculation of the path weight $pw(s_j)$ we use Eq. 4. As the failed nodes within the region-failure are carrying mission critical information of node A and B, we have to find the weight of each node on a mission-path to the sink node. The estimation of the $pw(s_j)$ is performed as follows. first, each mission-critical node such as A and B receives a weight of 32 based on equation 4. Since the node B has only two parent nodes (node F and node E), each of them will receive half of the weight of the node B (i.e 16). Node D will receive its $pw(D)$ weight of 24 which is the sum of 8 (half of the $pw(F)$ as node F has two parent nodes) and 16 (the $pw(E)$ as node E has only one parent node) received from its child nodes F and E, respectively. In the same way, Node G will have a $pw(G)$ weight of 8 (half of the $pw(F)$ of its child node F), Node C will have a $pw(C)$ weight of 52 (the sum of $pw(G)$, $pw(A)$, and half of $pw(D)$), Node H will receive a $pw(H)$ of zero since the node H is not on a mission-path, Node I will receive a $pw(I)$ weight of 12 which is half of the $pw(D)$ weight of its child node D, Node J will receive a $pw(J)$ of 64 that is the sum of $pw(I)$ and $pw(C)$. This process continues in the same way up to the sink node as shown in Fig. 4d. Then,

$\sum_{s_j \in fs_{Region2}} pw(s_j) = pw(I) + pw(J) = 76$. Note that, In Fig. 4c and Fig. 4d, a node z without a $pw(z)$ weight indicates that this node is not on a mission-path.

Finally, we calculate the bridge weight $bw(b_i)$ for each bridge using Eq. 5.In Fig. 4d, we have 4 different paths (bridges) crossing the region-failure and all of them carrying mission-critical informations. Hence, each bridge will be given a weight of 5 according to Equation 4. Consequently, $\sum_{b_i \in br_{Region2}} bw(b_i) = 20$. Finally, we have $rw_{Region2} = 5 * 70 + 76 + 20 = 446$.

Based on Eq. 1 of the proposed model, the region-failure with maximum weight will be chosen as the worst-case region-failure ($Region2$ in this example).

3.3. Region failure algorithms

In this section we introduce the main algorithms used to find out the worst-case region-failure under the single and dual region-failure scenarios in Section 3.3.1 and Section 3.3.2, respectively.

Hereafter, we present the main algorithms used in our model. The algorithm shown in Fig.5 finds all sensors within a region-failure. The algorithm depicted in Fig. 6 removes the failed sensors from the network topology (nt). The algorithm shown in Fig. 7 finds all paths between a given source/destination pair.

```
findSensorsWithinRegionAlgorithm (i, j, r, nt)
1       swr ←{ } // swr is the set of sensors within the region
                 // centered at (i, j) with the radius r.
2       foreach sensor s ∈ nt
                 // calculate the distance from the region center (i, j)
                 //  to the position of the sensor s (xs, ys).
3               d ← SQRT ((xs – i)² + (ys - j)²)
4               if  d ≤ r
                        // s is within the region so its ID is added to the set swr.
5                       swr ← swr U ids
6               end
7       end
8       return  swr  // return the set of sensors found within the region.
end
```

Figure 5. Find sensors within a region-failure Algorithm

The algorithm shown in Fig. 5 demonstrates how to find sensor nodes located within the region-failure. The input parameters to this algorithm are the network topology nt, center (i,j) and radius r of the region-failure. In this algorithm, we first calculate the distance d from the center of region failure to the sensor node. If d is less than the radius of region-failure, then the sensor is located within the failed region.

The algorithm shown in Fig. 6 presents our strategy for removing a sensor node from the routing table (rt) of all sensors in the network field. The input parameters to this algorithm are the network topology nt_1 and the list rs of sensor nodes to be removed. In this algorithm we perform the following steps. First, for each sensor node in the removed sensor list (rs) we search for that sensor id. If the sensor ID. is found in any routing table of a sensor node that

```
removefailedSensorsAlgorithm (nt₁, swr)
1         foreach sensor s ∈ swr
2              foreach sensor sn in nt₁ & sn ∈ rt of s
                         // rt refers to a routing table
3                   nt₁ ← remove s from rt of sn in nt₁
4              end
5              nt₁ ← nt₁ ∩ s  // remove s from the nt₁
6         end
7         return nt₁  // return the updated network topology
end
```

Figure 6. Remove sensors from the routing tables algorithm

belongs to the network topology, then it is removed from the routing table and the network topology is updated with the changes made.

```
findAllPathsAlgorithm (paths, p, s, d)
1         if s = d  // A path p is found
2              paths ← paths U p  // the path p is added to the set of paths
3              return paths
4         else
5              foreach sensor sn ∈ parent or sibling of s
6                   if sn not ∈ p          // to prevent cycle
7                        findAllPathsAlgorithm (paths, p U sn, sn, d)
8                   end
9              end
10        end
end
```

Figure 7. Find All Paths from s to d Algorithm

The algorithm shown in Fig. 7 presents how to check if a sensor node s is still connected or not after a region-failure happens. In other words, we need to find all paths p from the given node s to the destination node d and add these paths to the empty list *paths* of paths. This is accomplished by checking first if the path already exists in the paths list *paths*. If it does not exist, a path search is initiated from each parent of the node s. If a path is found, the search is terminated, the found path is added to the paths list and the node s is considered a connected node. On the other hand, if no path exists from the parent node of node s, then a path search is initiated from the sibling node of node s. Finally, if no path is found from both the parent and the sibling nodes of node s the node is marked as disconnected node.

3.3.1. Single region-failure

The algorithm shown in Fig. 8 demonstrates how to estimate the number of disconnected nodes due to a single region-failure. The input parameters to this algorithm are the network topology nt, the radius of the failure region r, the increment value Δr for the radius r, the threshold of disconnected nodes ns_{th}, and the network field's length nfl and width nfw. In this algorithm, we first generate a failure region with radius r. then we find all sensors located within the region by the algorithm shown in Fig. 5. Sensors located within the failure region

are then removed from the routing tables of all nodes in the network topology by executing the algorithm shown in Fig. 6. For the rest of the remaining sensors, we investigate the availability of a path from each node to the sink node by carrying out the algorithm depicted in Fig. 7. If a path is found the node is marked as connected node and disconnected otherwise. then we calculate the number of disconnected nodes due to the failure region. The above mentioned scenario is repeated by incrementing the coordinates of center of the region-failure until th whole topology is scanned by region-failures. Finally, Equation 1 is applied to estimate the worst case region-failure.

---

singleRegionFailureAlgorithm (r, Δr, nt, nfl, nfw, ns_{th})
      // i and j are the center of a region with the radius r
1   *regions* ← { }
2   **for** i ← r to nfl, i incremented by Δr
3   **for** j ← r to nfw, j incremented by Δr
4   id_r ← i || : || j // id_r is the ID of the region
5   *swr* ← { } // the set of sensors within the region id_r
6   *swr* ← *findSensorsWithinRegionAlgorithm* (i, j, r, nt)
7   nt_1 ← *removeFailedSensorsAlgorithm* (nt, *swr*)
8   *dis* ← { } // the disconnected sensors within the region id_r
9   **foreach** sensor $s \in nt_1$
10   *paths* ← *findAllPaths* ({ }, {s}, nt_1, s, *sink*)
11   **if** *paths* is empty // no path exists
12   *dis* ← *dis* ∪ s // the node s is disconnected
13   **end**
14   **end**
               // number of disconnected nodes are above certain threshold
15   **if** *length* (*dis*) > ns_{th}
16   *regions* ← *regions* ∪ {id_r, *dis*}
17   **end**
18   **end**
19   **end**
      // return the set of failed regions with their associated disconnected nodes ns_{th}
20   return *regions*
end

Figure 8. Algorithm for Finding the number of disconnected nodes under single region-failure

3.3.2. Dual region-failure

Under dual region-failure scenario, we propose the following algorithm shown in Fig. 9 to estimate the number of disconnected nodes within the given network topology.

The algorithm shown in Fig. 9 demonstrates our strategy to determine the number of disconnected nodes under dual region-failures scenario.The input parameters to this algorithm are the same as that of a single region failure shown in Fig. 8. In this algorithm, at the beginning, we follow similar steps to that used to find the number of disconnected nodes of a single region failure where we first generate a failure region with radius r. then we

```
twoRegionsFailureAlgorithm (r, Δr,  nt, nfl, nfw, ns_th)
            // i and j are the center of one region with the radius r
1           regions ← { }
2           for  i ← r to nfl, i incremented by Δr
3                   for  j ← r to nfw, j incremented by Δr
4                           id_r1 ← i || : || j    // id_r1 is the ID of the first region
5                           swr_1 ← { } // the set of sensors within the region id_r1
6                           swr_1 ← findSensorsWithinRegionAlgorithm (i, j, r, nt)
7                           nt_1 ← removeFailedSensorsAlgorithm (nt, swr_1)
                            // k and l are the center of the second region with the radius r
8                           for  k ← i to nfl, k incremented by Δr
9                                   for  l ← r to nfw, l incremented by Δr
10                                          id_r2 ← k || : || l  // id_r2 is the ID of the second region
                                            // twoReg_id represents an ID for the underlying two regions
11                                          twoReg_id ← id_r1 || id_r2
                                            // the dis is the set of the disconnected nodes associated with
                                            // the two regions identified by the ID twoReg_id
12                                          dis ← { }
13                                          swr_2 ← { } // the set of sensors within the region id_r2
14                                          swr_2 ← findSensorsWithinRegionAlgorithm (k, l, r, nt_1)
15                                          nt_2 ← removeFailedSensorsAlgorithm (nt_1, swr_2)
16                                          foreach sensor s ∈ nt_2
17                                                  paths ← findAllPaths ({ }, {s}, nt_2, s, sink)
18                                                  if  paths is empty    // no path exists
19                                                          dis ← dis ∪ s // the node s is disconnected
20                                                  end
21                                          end
                                            // number of disconnected nodes are above certain threshold ns_th
22                                          if  length (dis) >  ns_th
23                                                  regions ←  regions ∪ {twoReg_id, dis}
24                                  end
25                          end
26                  end
27          end
            // return the set of dual-failed regions with their associated disconnected nodes
28          return regions
end
```

Figure 9. Algorithm for Finding the number of disconnected nodes under dual region-failures

find all sensors located within the region by the algorithm shown in Fig. 9. Sensors located within the failure region are then removed from the routing tables of all nodes in the network topology by executing the algorithm shown in Fig. 6. The above mentioned steps are repeated for the second region-failure. Now, we have come up with a network topology without the failed nodes due to the dual region-failure. Then, we examine the path availability from each node in the network topology to the sink node by executing the algorithm depicted in Fig. 7. If no path is available from a node to the sink node the node is marked as a disconnected node. On the other hand, if a path is found the node is marked as connected node. then we

calculate the number of disconnected nodes due to the failure region. The above mentioned scenario is repeated by iterating all the possible combinations of the regions centers until the whole topology is visited by region-failures. As the number of disconnected nodes due to dual region-failures is usually large, we use a threshold such that we get the dual region-failures that lead to a number of disconnected nodes greater than the predefined threshold.

Finally, Equation 1 is applied to estimate the worst case region-failure.

4. Results

In this section, we present our simulation results. In our simulations we consider the network topology shown in Fig. 10 in which node 0 is the sink node. The failure information of different region failures generated during our simulations results are shown in Table 1 and Table 2, respectively.
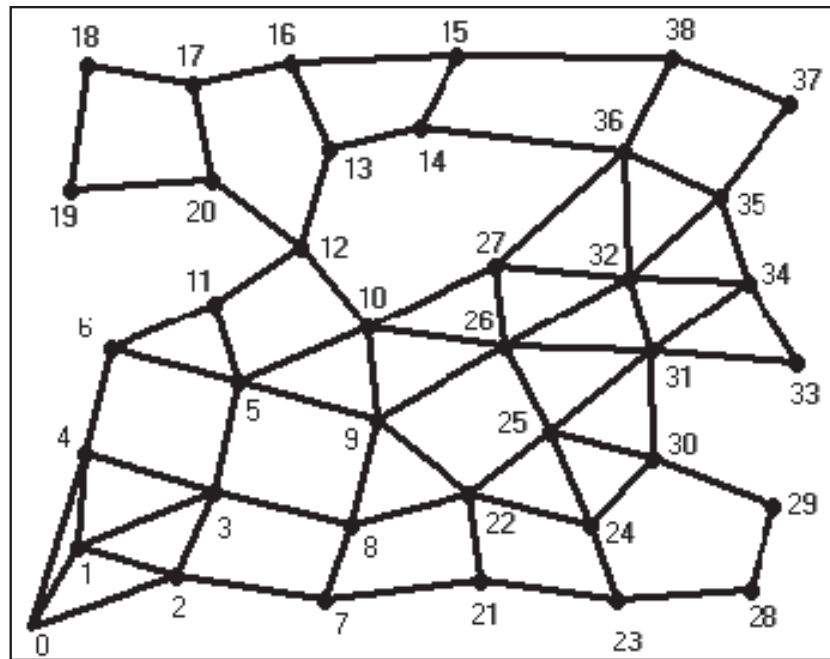
Figure 10. Network Topology

Note that, due to the large number of region-failures generated, we present only region failures that lead to disconnecting more than two and eight nodes for the single and dual failure scenarios, respectively.

Hereafter, we investigate the location of the worst-case region-failure under Link-based, Node-based (without a mission critical node) and Node-based with a mission critical node in Fig. 12, Fig. 14 and Fig. 16, respectively. In the Node-based with a mission critical node, the node 28 is chosen as a Mission-Critical(MC) node.

The results shown in Fig. 11 show that, based on the traditional definition of the worst-case region cut, the failure region with id 5 is considered the worst-region cut as its failure leads to having the maximum number of failed links. The location of this failure-region is depicted

Region id	Failed Nodes IDs	No. of Failed links	No. of Disconnected Nodes
1	12	4	8
2	11, 12	6	8
3	12, 13	6	7
4	2, 3	8	5
5	31, 32	11	3
6	13	3	3
7	7, 8	6	3
8	7	3	3
9	21	3	2

Table 1. Region-failure information

Region id	Failed Nodes' IDs at Region 1	Failed Nodes IDs at Region 2
1	6	2, 3
2	4	2, 3
3	1, 4	2
4	1, 4	2, 3
5	2	12
6	2	11, 12
7	2	12, 13
8	2, 3	5, 11
9	2, 3	12
10	2, 3	11, 12
11	2, 3	12, 13
12	5, 11	7, 8
13	12	7
14	12	7, 8
15	12	31, 32
16	11, 12	7
17	11, 12	7, 8
18	11, 12	31, 32
19	12, 13	31,32

Table 2. Failed Nodes due to dual region-failure

in Fig 12. It is also notable that, according to Table 1 the failure of this region results in disconnecting 3 sensor nodes namely, node 33, 34, and 35 as depicted in Fig. 12.

The results shown in Fig. 13 clearly indicate that, in absence of mission-critical nodes within the network topology the proposed model is very clever to find out the worst-case region-failure. The worst-case region-failure is the region-failure that has the maximum
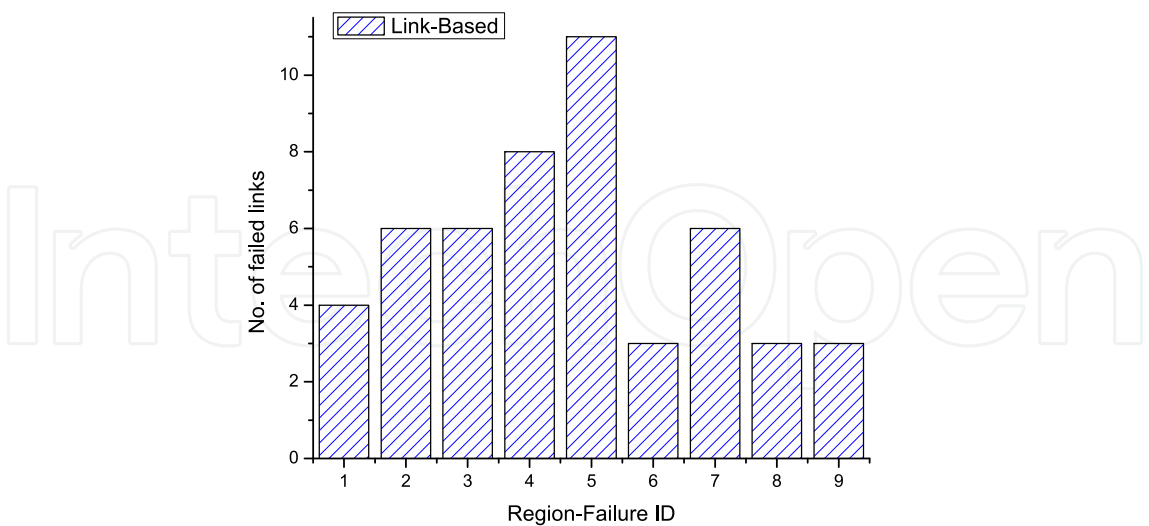
Figure 11. Single Failure-region weights under link-based approach.
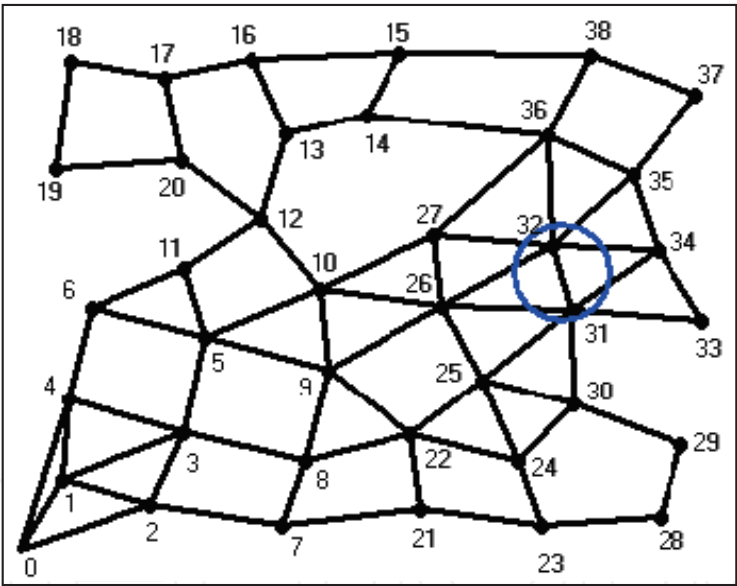

Figure 12. Worst-case region failure under Link-based approach.

impact on the network performance which is region 2 in this case as its failure results in disconnecting 8 sensor nodes, namely, 13, 14, 15, 16, 17, 18, 19 and 20. However, according to Fig. 11, the failure of this region leads to failure of 6 links only.

The results shown in Fig. 15 indicate clearly that, with the proposed model of a region-cut, introducing a mission-critical node into the network topology leads to a change in selecting the worst-case region cut which is region 4 as its failure results in disconnecting 5 sensor nodes, namely, 7, 8, 21, 23, and 28 as shown in Table 1. However, according to Fig. 11, the failure of this region leads to failure of 8 links.
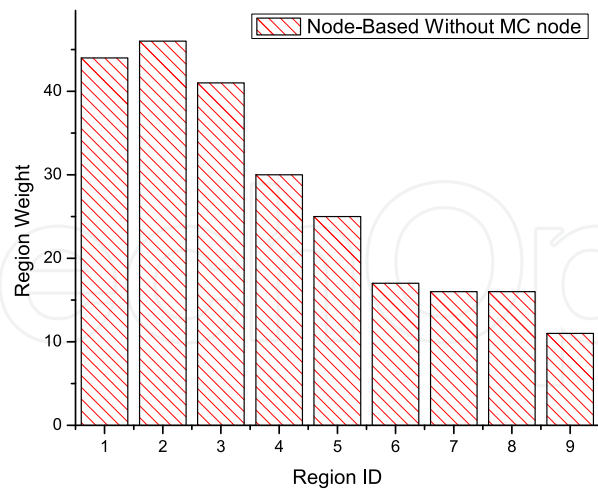
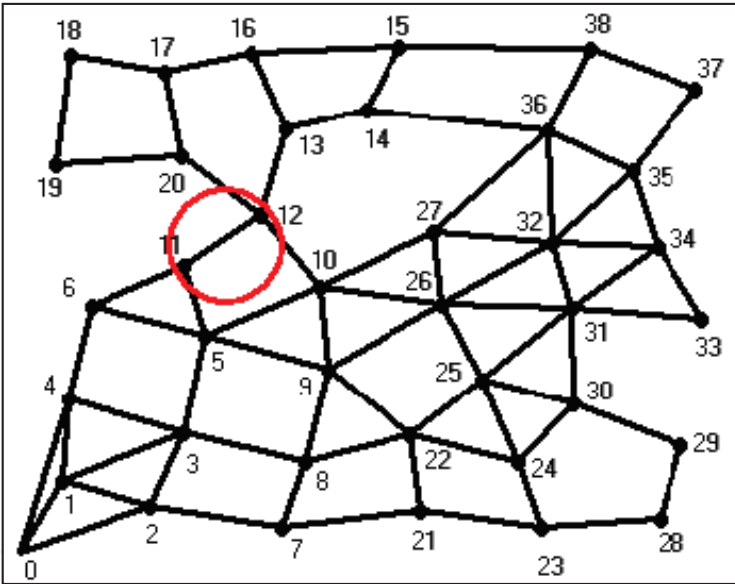**Figure 13.** Single Failure-region weights under node-based approach without MC-node.

Figure 14. Worst-case region cut using Node-based approach.

Therefore, we claim that, using the failed links as the only criteria for defining the worst-case region cut is impractical as it ignores the case that the failure of some nodes may lead to failure of few links however its impact on the network is more severe due to disconnecting larger number of nodes. Moreover, it disregards the fact that some network nodes have higher priority than others.

Hereafter, we investigate the location of the worst-case dual region-failures without including a MC-node and with a MC-node in Fig. 17 and Fig. 18, respectively.

The locations of the worst-case dual region-failures shown in Fig. 17 indicate that, using the link-based approach (blue regions including nodes 11,12 and 31, 32) lead to cutting of 17 links and disconnecting 12 nodes. On the other hand, under the dual region failure scenario,
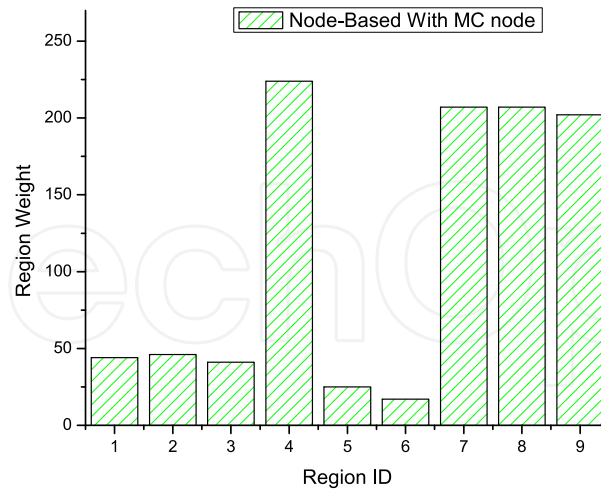
Figure 15. Worst-case single region failure under Node-based approach with MC node
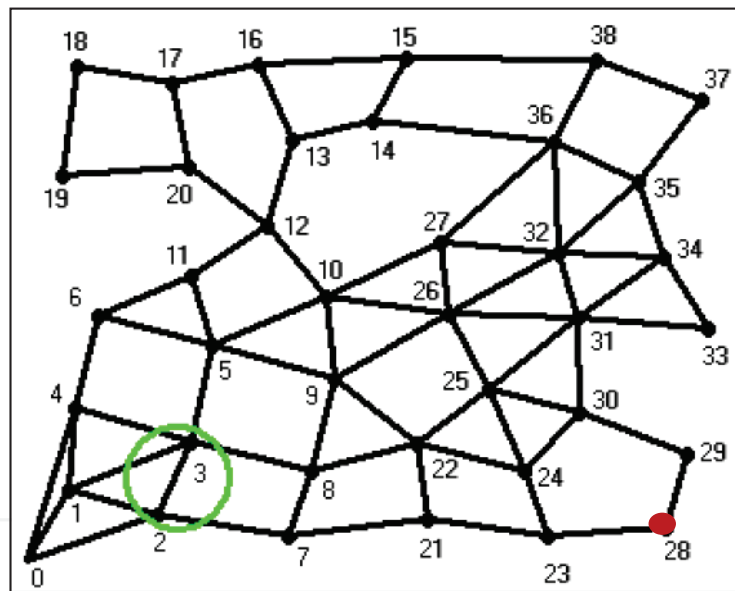

Figure 16. Worst-case region failure underNode-based approach with MC node

using the proposed Node-based approach, the worst-case region-failure depicted in Fig. 17 (red regions including nodes 1,4 and 2) have only 8 link-cuts however, it results in a full disconnection of all the network nodes. This is due to the fact that, based on our approach, the worst-case region cuts are located near by the sink node and the failure of these nodes lead to isolating the sink node from the whole network. The results shown in Fig. 18 show that, by introducing the MC-node number 28, the worst-case dual region failure remains unchanged as the earlier case without MC-node. This is due to fact that , by introducing the MC-node, no further influence can affect the network as the maximum impact has been already happened by having a complete disconnected sensor network.
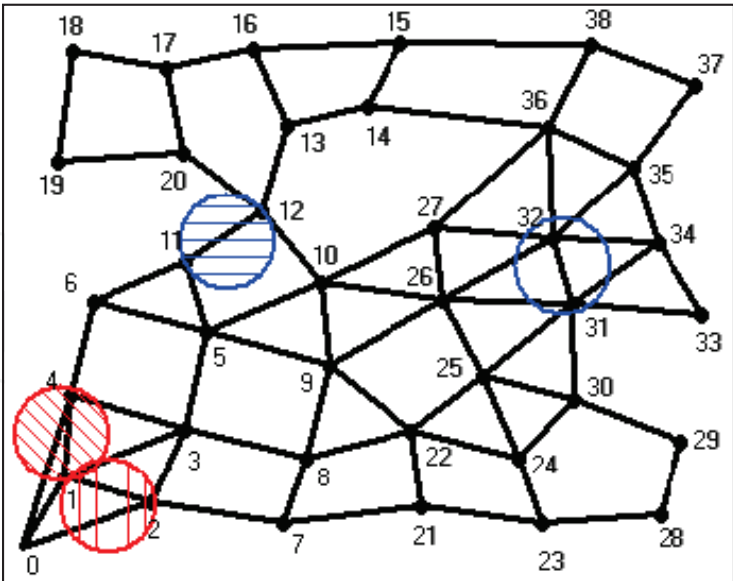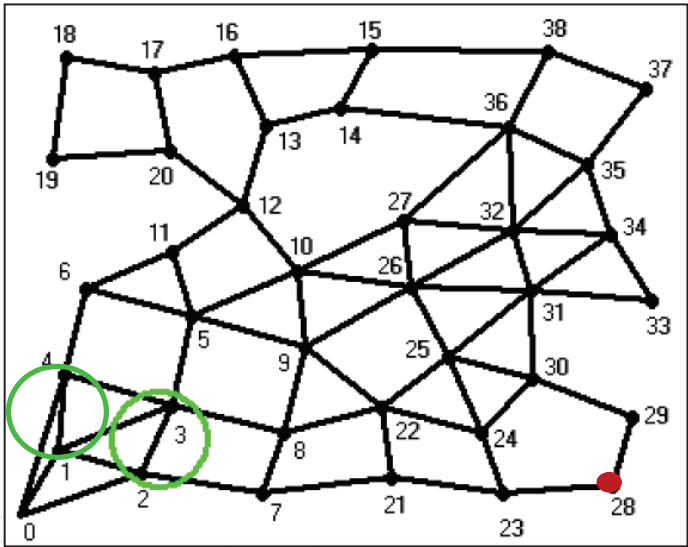
Figure 17. Worst-case Dual-Region Failures

Figure 18. Worst-case Dual-Region Failures with a mission node

5. Conclusion

In this chapter, we introduced a new model for for a worst-case cut (partition) due to failure regions. The proposed model takes into consideration the physical correlation among the locations of the network nodes and the possible priority of some nodes over the others. Based on the proposed model, We identified the location of a disaster that would have the maximum impact on a sample test network under single and dual region-failure scenario. Extensive Simulation results indicate that, using the number of failed links as the main criteria for defining the worst-case region cut, underestimates the impact of a region failure on the overall network performance.

Author details

Mohamed M. A. Azim and Aly M. Al-Semary
Taibah University, Saudi Arabia

6. References

[1] Liotine, M. (2003). *Mission-Critical Network Planning*, Artech House Publishers, ISBN: 1-58053-516-X, USA.

[2] Markopoulou, A.; Iannaccone, G.; Bhattacharyya, S.; Chuah, C., & Diot, C. (2004). Characterization of failures in an IP backbone, *IEEE INFOCOM 2004* , pp. 2307-2317.

[3] A. Azim, Mohamed Mostafa and Kabir, M. N.,(2011), Availability Study of M:N Automatic Protection Switching Scheme in WDM Networks, International Journal of High Speed Networks, Vol. 18, No. 1, pp. 1-13, IOS Press, Dec. 2011.

[4] Lee, K.; Lee, H.-W.; Modiano, E., (2011), "Reliability in Layered Networks With Random Link Failures," IEEE/ACM Transactions on Networking, vol.19, no.6, pp.1835-1848, Dec. 2011.

[5] Kini, S.; Ramasubramanian, S.; Kvalbein, A.; Hansen, A.F.;(2010) , "Fast Recovery From Dual-Link or Single- Node Failures in IP Networks Using Tunneling," IEEE/ACM Transactions on Networking, vol.18, no.6, pp. 1988-1999, Dec 2010.

[6] A. Azim, Mohamed Mostafa and El-semary, M. Aly,(2012) Vulnerability Assessment for Mission Critical Networks Against Region Failures: A Case Study, International Conference on Computing and Information Technology (ICCIT2012), pp. 838-843, March 2012.

[7] Sen A., Shen B., Zhou L., & Hao B. (2006). Fault-tolerance in sensor networks: A new evaluation metric *INFOCOM, 2006*, pp. 1-12.

[8] Sen A., Murthy S., & Banerjee S. (2009). Region-based connectivity: A new paradigm for design of fault-tolerant networks *High Performance Switching and Routing 2009*, pp. 94-100, Paris, France.

[9] Sen A., Banerjee S., Ghosh P. & Shirazipourazad S. (2009). Impact of region-based faults on the connectivity of wireless networks . *Allerton conference on communication control and computing 2009*, Monticello, IL, 2009, pp. 1430-1437.

[10] Jiajia Liu; Xiaohong Jiang; Nishiyama, H.; Kato, N.; , "Reliability Assessment for Wireless Mesh Networks Under Probabilistic Region Failure Model," Vehicular Technology, IEEE Transactions on , vol.60, no.5, pp.2253-2264, Jun 2011

[11] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, Assessing the vulnerability of geographically correlated network failures, in Proc. MILCOM, 2008, pp. 1-6.

[12] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, Assessing the vulnerability of the fiber infrastructure to disasters, in Proc. INFOCOM, 2009, pp. 1566-1574.

[13] S. Neumayer and E. Modiano, Network reliability with geographically correlated failures, in Proc. INFOCOM, 2010, pp. 1658-1666.

[14] M. M. Hayat, J. E. Pezoa, D. Dietz, & S. Dhakal, Dynamic load balancing for robust distributed computing in the presence of topological impairments, in Handbook of Science and Technology for Homeland Security. Hoboken, NJ: Wiley, 2009.

[15] W. Wu, B. Moran, J. Manton & M. Zukerman, Topology design of undersea cables considering survivability under major disasters, in Proc. WAINA, 2009, pp. 1154-1159.

[16] K. Kim and N. Venkatasabramanian, Assessing the impact of geograph- ically correlated failures on overlay-based data dissemination, in Proc. GLOBECOM, 2010, pp. 1-5.

[17] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, Network vulnerability to single, multiple, and probabilistic physical attacks, in Proc. MILCOM, 2010, pp. 1824-1829.

[18] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, The resilience of WDM networks to probabilistic geographical failures, *INFOCOM, 2011*, to be published.

[19] Myounggyu Won & Stoleru, R.; , "Destination-Based Cut Detection in Wireless Sensor Networks," Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on , vol., no., pp.55-62, 24-26 Oct. 2011.

[20] Barooah, P., Chenji, H., Stoleru, R. & Kalmar-Nagy, T., (2012) Cut Detection in Wireless Sensor Networks, *IEEE Transactions on Parallel and Distributed Systems*, vol.23, no.3, pp.483-490.

[21] Turner, D., Levchenko, K., Snoeren, A. C., & Savage, S., *SIGCOMM Computer Communcations*, California fault lines: understanding the causes and impact of network failures, Rev., vol. 40, pp. 315-326, August 2010.

[22] S. Kini, S. Ramasubramanian, A. Kvalbein, and A. F. Hansen, Fast recovery from dual link failures in ip networks, in Proceedings INFOCOM 2009, pp. 1368 - 1376, 2009.

[23] H. Choi, S. Subramaniam, and H. ah Choi, On double-link failure recovery in WDM optical networks,- in in Proceedings of IEEE INFOCOM, 2002, pp. 808-816.