Bridge protection algorithms - a technique for fault-tolerance in sensor networks

Saad Ahmad Khan, Ladislau Bölöni[1], Damla Turgut

Department of Electrical Engineering & Computer Sciences,
University of Central Florida,
4000 University Blvd, Orlando FL
USA 32816

Abstract

Sensor networks operating in the field might be subject to *catastrophic events* which destroy a large number of nodes in the geographic area. Often, the aftermath of such an event is the creation of a *network of bridged fragments* where connectivity is maintained by one or several bridge nodes. These networks are vulnerable, because the bridge nodes will soon exhaust their energy resources leading to the fragmentation of the network. This paper describes a *bridge protection algorithm* (BPA), a combination of techniques which, in response to a catastrophic event, change the behavior of a set of topologically important nodes in the network. These techniques protect the bridge node by letting some nodes take over some of the responsibilities of the sink. At the same time, they relieve some other overwhelmed nodes and prevent the apparition of additional bridge nodes. To achieve this, the algorithm sacrifices the length of some routes in order to distribute routes away from critical areas. In a variation on the BPA algorithm, we show that if geographic information about the nodes is available, replacing shortest path routing with a routing model which follows the edges of the relational neighborhood graph will lead to further improvements in the expected connected lifetime of the network.

Keywords: Sensor network, Fault tolerance, Bridge protection

1. Introduction

The nodes of a sensor network must be deployed in such a way that both the sensing and the communication requirements of the overall network are met. Sensor nodes can go off-line for a variety of reasons: running out of energy, environmental events (*e.g.*

Email addresses: skhan@eecs.ucf.edu (Saad Ahmad Khan), lboloni@eecs.ucf.edu (Ladislau Bölöni), turgut@eecs.ucf.edu (Damla Turgut)

[1]Corresponding author, tel:1-407-823-2320, fax:1-407-823-5835

forest fire, landslides) as well as the activity of opposing forces (*e.g.* intruders disabling or compromising the sensor nodes which they detected through visual observation or radio-location). In such scenarios, naturally, the sensing quality suffers, as the off-line nodes do not contribute their sensing to the overall picture. The sensing quality loss is proportional with the number of off-line nodes.

The worst case scenario, however, happens when the loss of a single node can lead to the fragmentation of the network into disjoint subsets of nodes. This way, the loss of a single node can lead to a catastrophic loss of functionality, because even from areas where the sensors are intact, data cannot reach the sink. A well-engineered network will never fragment due to the energy consumption during the normal course of operation as the differences in the energy consumption can be taken into account during design time.

If, however, a natural or man-made catastrophic event destroys a large subset of the nodes, the remaining network can emerge with a heavily unbalanced topology which could not have been predicted at deployment time. Let us consider a situation where the connectivity still exists, but the network graph is split into several fragments, linked by *bridge nodes*. We define the bridge node as a node whose removal disconnects the network[2]. In contrast to nodes which have been engineered to handle a high load, bridge nodes are general purpose nodes which ended up in the bridge position due to unpredictable external circumstances. They do not have higher energy resources or longer transmission range, and yet they need to transport the complete traffic of the fragment on the opposite side from the sink.

In this paper we describe a series of techniques called Bridge Protection Algorithms (BPAs). An early version of this technique has been presented in [1]. BPAs form a coherent response of the network to a catastrophic event which created a network topology of bridged fragments. The BPA changes the behavior of the bridge nodes and their neighbors in such a way as to lower the energy consumption of the bridge and to prevent future failures in the area which could create new bridge nodes.

The reminder of the paper is organized as follows. Section 2 describes the scenario we are considering, the performance metrics and the ways in which we can model catastrophic events. Section 3 describes the basic principles behind bridge protection algorithms, including the classification of specific nodes in the local area of a bridge node. Section 4 proposes a technique where geographical information about the nodes can be used to improve the efficiency of the BPA algorithm, by replacing shortest path routing with a routing in the network defined by the relational neighborhood graph. Section 5 describes the results of a simulation study comparing the performance of the BPA variants with the baseline response of a sensor network to a catastrophic event. Related work is discussed in Section 6. We conclude in Section 7.

[2]Our usage of the term "bridge" differs slightly from the standard usage in graph theory. In graph theory a "bridge" is defined as an *edge* whose removal fragments the graph, while a node whose removal disconnects the graph is called a "cut-node".

2. Scenario: catastrophic events in an intruder tracking sensor network

2.1. The sensing task and the physical network

The scenario we are considering is that of an intruder detection system protecting an *area of interest* such as the surroundings of a high value military installation. In such a scenario the "smartdust" model where disposable nodes are deployed randomly (e.g. thrown out from airplanes) is not appropriate. Instead, the area is protected by a *permanently deployed wireless sensor network*, where sensor nodes costing hundreds or thousands of dollars each are deployed in carefully chosen locations. The nodes need to be in position for many years, requiring regular battery changes, with nodes expected to have larger energy consumption (for instance by being closer to the sink) being given batteries of larger capacity. In such a permanent sensor network, under normal conditions, we don't have the extreme energy limitations of the "smartdust" type of nodes. This, however, changes in the case of a catastrophic event when the bridge node needs to take on traffic many times larger than what it was designed for.

The ideal arrangement of nodes would be a rectangular or hexagonal regular grid. The density of the grid depends on the sensing and transmission range of the nodes. The sensing range determines how well the interest area will be covered by the sensors. We would prefer that every location to be covered, even by multiple nodes: but this is a *soft* preference: an intruder detection system can operate with partial coverage. The transmission range dependency, however, is hard: if a node can not communicate with its neighbors, the system will not be operational. One reasonable compromise is to determine the grid size such that the node is within transmission range of all neighboring nodes, including along the diagonal, but it is not in the transmission range of nodes two hops away. In an ideal connection, this would imply that each node would have eight neighbors. Common sense engineering considerations dictate that arrangement to be chosen such that the transmission range of the nodes to be somewhat higher than the required minimum to maintain connectivity.

In practice, however, environmental conditions (*e.g.* the obstacles and camouflage opportunities in the environment) make the achievement of a perfect grid unfeasible. The customer would prefer to position the node to a location at some distance from the exact grid position, if this location offers advantages. In the resulting "grid with noise" arrangement of the nodes, some nodes might not reach all the near neighbors, but they might possibly reach one hop away neighbors. The main flow of information on the network is directed from the sensor nodes to the sink. The nodes detect intruders in their sensor range and send their observations with a hop-by-hop approach to the *sink node*, which has the ability to directly transmit the data to the customer.

Let us now discuss the nature of the routing algorithm used in a permanently deployed sensor network. In an intruder tracking system the overriding design requirement is that the information about intruders is transmitted as quickly as possible to the sink. This requires that the routing algorithm must converge to the shortest path in the

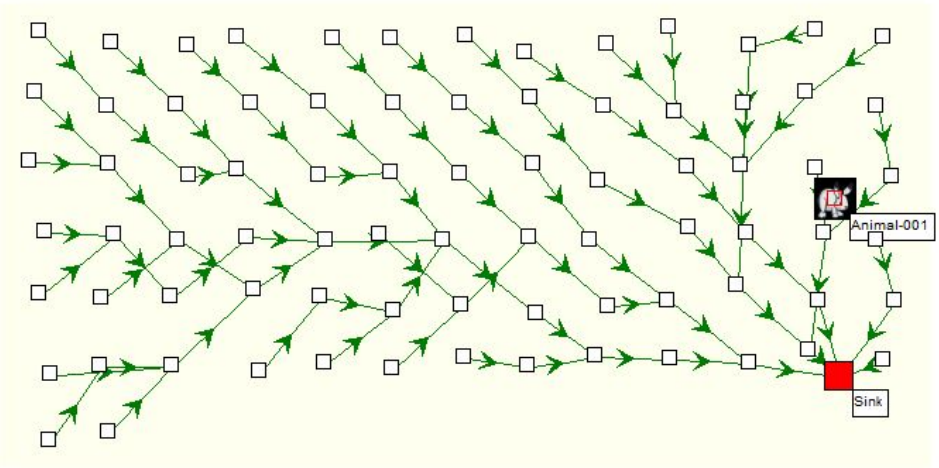


Figure 1: Forwarding paths in the SP routing before the catastrophic event (YAES screenshot).

number of hops. Note that this does not determine a unique routing algorithm - many routing algorithms used in wireless networks converge to this or closely related metrics. These include distributed techniques using variations of Bellman Ford (DSDV, OLSR and Babel), gradient based techniques such as directed diffusion, as well as centralized techniques where the neighborhood information is flooded until it reaches the sink, which calculates all the routes and transmits them back to the nodes. On-demand routing protocols such as AODV also choose shortest path, although on-demand routing does not offer advantages in a permanent sensor network. Different routing algorithms differentiate themselves in the way they handle changes in the network topology triggered by node failures or mobile nodes, as well as by the routing overhead, the cost of the signaling necessary to establish and modify the routes. However, for a sensor network with nodes deployed for years in the same position there is little to differentiate between proactive routing algorithms: the calculated routes will be the same and the cost of overhead will be amortized by the long timespan over which the routing tables remain unchanged. In the reminder of this paper, we will consider as our baseline algorithm a generic shortest path algorithm SP, which can stand-in for any algorithm which generates a shortest path route. With this assumption, the forwarding paths in the sensor network will look like in Figure 1.

The sink node is interested in (a) intruder tracking and (b) monitoring the health of the sensor network.

Intruder tracking: The sink node is interested in tracking intruders: in particular, for

each intruder it wants to know whether it is inside the interest area or not and, for intruders inside the interest area, their most recent location. The sink is also interested in independent confirmations of the location of a certain intruder. We will make the assumption that the nodes will transmit their own observations at fixed time intervals, but immediately forward other node's transmissions.

Although simple, this policy has several important practical consequences. If an intruder leaves the interest area, the sensor node will send exactly one transmission reporting it. It will continue to track the intruder, as long as it is in the sensor range, but it will not report its location, unless it re-enters the interest area. If the sensor makes several successive observations before the next scheduled transmission, it will transmit only the most recent observation (as the sink is not interested in historical information).

Note that in any well-engineered deployment, this default algorithm will be augmented with a number of additional strategies to lower the energy consumption of the network. These might include temporal and/or spatial aggregation of reports, selective transmission based on value of information or estimates of the sink's knowledge and so on. The algorithms we propose in this paper are orthogonal to and can be combined freely with aggregation and selective transmission. Nevertheless, with many simultaneous techniques applied diminishing returns will inevitably set in. Other techniques, such as clustering, require more extensive changes to be combined with BPA.

Sensor network health monitoring: In order to correctly interpret the received data, the sink node also needs to monitor the health and integrity of the sensor network, *i.e.* the sink needs to know which nodes are functional. If a node is not sending data, it can mean either that the node is not seeing any intruders or that the node is down.

To maintain the state of the network we will require the nodes to send *heartbeat* messages at specific intervals when they do not have anything else to send. Any reported observation automatically replaces the heartbeat signal.

2.2. Modeling catastrophic events

In the introduction of this paper we discussed the idea of a catastrophic event damaging a network and creating bridge nodes. In the following, we will discuss more precisely what type of catastrophic events we consider.

We define a catastrophic event (CE) as a sudden loss of a significant number of sensor nodes. We will concentrate on geographically limited catastrophic events, in which all the nodes in a given, well specified geographic area are lost. Such events can be the result of forest fires, floods, chemical contamination or the action of opponent forces.

Depending on the relative position of the geographic area of the CE and the interest area, we consider several cases based on the ability of a network to collect useful data. For each case, we will also consider the ways in which the network and its customer can respond to a CE. In general, a non-physical intervention (such as recomputing the routing tables) is a comparatively cheap way to handle a CE. On the other hand, physical

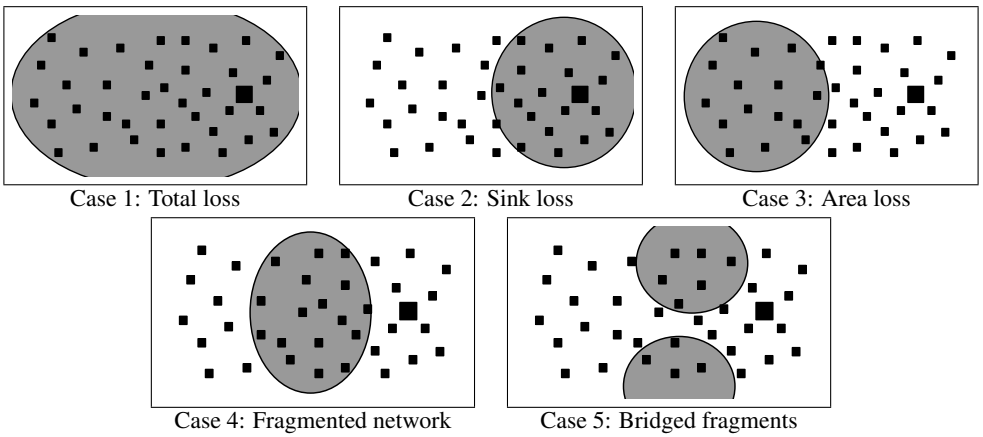


Figure 2: Possible impacts on the network by a catastrophic event.

intervention, such as the deployment of new nodes, is expensive and intrusive. The five cases listed below are illustrated in Figure 2.

Case 1: Total loss. In this case, all the nodes in the network as well as the sink are lost. There is no remaining ability to collect data, and the only way to recover data collection is to deploy a new sensor network and sink.

Case 2: Sink loss. In this case, not all the nodes are lost, but the sink is part of the geographic area of the CE. Thus, although some of the nodes are functional and communicating, no data is reaching the customer. The immediate solution for this situation is to install a new sink node in the functioning area of the network. This might require multiple sinks for disconnected fragments. Unless regular sensor nodes can be converted to act as sink nodes, this requires physical intervention in the network.

Case 3: Area loss. The network had lost significant areas, but the sink is functional and the remaining part of the network is strongly connected. The network can recover its remaining functionality through an update of the routing tables.

Case 4: Fragmented network. The CE divided the network into two or more fragments, only one being connected to the sink. The network can recover data collection from the fragment connected to the sink, but to collect data from the disconnected fragments, it needs physical intervention. This can take either the form of deploying new sensor nodes to bridge the fragments or by installing a new sink in the disconnected fragment.

Case 5: Bridged fragments. In this case, the network is close to be fragmented, however, the fragments are still connected by a narrow bridge. This situation also arises when a fragmented network is repaired by adding a minimal number of bridge nodes. A network of bridged fragments is still functional, provided that the routing tables had been appropriately recalculated. However, a bridged network is vulnerable, because the transmission of a complete fragment depends on the bridge node (or a small number of bridge nodes).

The techniques we described in the remainder of this paper are mitigating the vulnerability of a network of bridged fragments - a scenario which can appear either as the direct result of a CE or as a result of the repairs following a CE which created a fragmented network.

3. Basic principles of bridge protection

3.1. The topology of a network of bridged fragments

Our techniques described in the remainder of this paper will be based on specialized routing and processing techniques in sensor nodes function of their topological position in the network. In this section we will identify the type of nodes which will receive differentiated responsibilities.

We assumed that prior to the CE the network uses shortest path routing from each node to the sink (Figure 1). The first step after the CE is the recomputing of the routing tables. An example of this, for the case of a CE creating a network of bridged fragments is shown in Figure 3. For the remainder of this discussion, we shall assume that there are exactly two bridged fragments (the generalization to multiple fragments is immediate). We will call the fragment which contains the sink the *near side* while the fragment which does not contain the sink the *far side* fragment. We call a node a *bridge node* if removing it from the network will disconnect the far side fragment from the sink. There may be more than one bridge node. We call *gate nodes* the non-bridge neighbors of a bridge node on the far side. Intuitively, the bridge nodes are heavily loaded, because they need to handle the complete information flow from the far side. Similarly, we call *fan-out nodes* the non-bridge neighbors of a bridge node on the near side. There are always at least two gate nodes and two fan-out nodes because if there would be only one, that node would be a bridge node. See Figure 4 for an example of the location of the bridge node, gate nodes and fan-out nodes.

3.2. The objectives of the bridge protection algorithm

We designed the bridge protection algorithm with the following objectives in mind:

- Bridge node protection: the BPA should be able to protect the bridge nodes from prematurely exhausting their energy resources.
- Prevent the creation of new bridge nodes: we have seen that there are at least two of the gate and fan-out nodes, thus the failure of a gate or fan-out node will not disconnect the network. However, such a node might be transformed into a bridge node if all the other gate or fan-out nodes fail.
- Minimal intrusion and sustaining the main functionality of the network: The BPA algorithm should be able to sustain the normal operation of the network with minimal intrusion. The algorithm should not intervene with the functionality of the rest of the network.

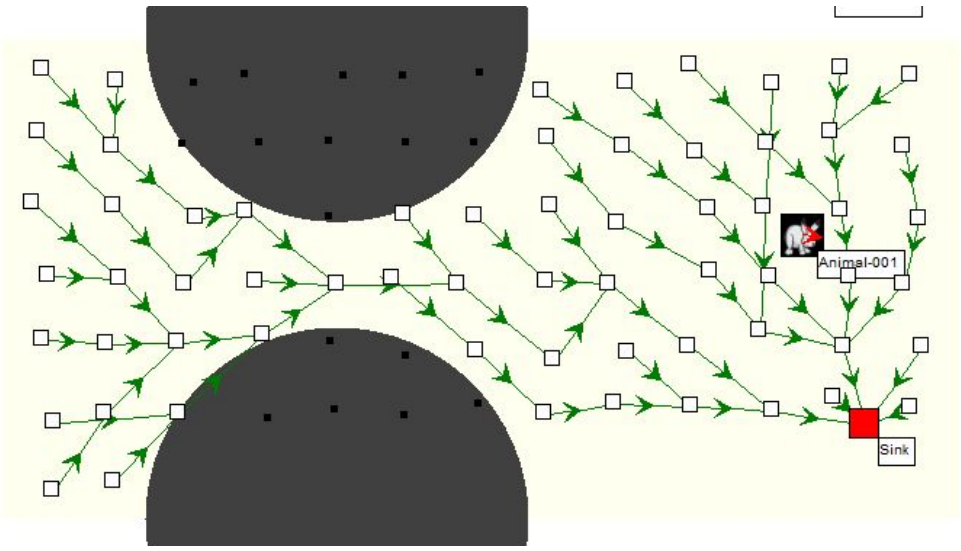


Figure 3: Forwarding paths reconfigured by the baseline algorithm after the catastrophic event (YAES screenshot).

In the following we discuss the behavior of a routing algorithm we label BPA-SP (bridge protection running on top of shortest path routing).

3.3. BPA functionality at the gate nodes

The BPA algorithm changes the functionality of the gate nodes (see Algorithm 2), in order to reduce the amount of data reaching the bridge nodes. Intuitively, this must be done one step before the bridge node, because once the data reached the bridge, any processing will inevitably consume from the bridge node's energy. The overall technique we will deploy involves shifting some of the reasoning about the messages which happen at the sink node to the gate nodes. Thus the gate nodes will treat differently the following message types:

- Heartbeat messages: normally, heartbeat messages from all the nodes reach the sink, which will infer that a quiet node which sends neither observations nor heartbeat messages over a period of time is likely a failed node. While heartbeat messages are transmitted at relatively long intervals, they can present a significant load on the bridge which must forward all the heartbeat messages of the far side nodes. To avoid this, the gate nodes will perform the reasoning about the live status of the far side nodes and only send *node failure messages* (if necessary) through the bridge node. On the other hand, the gate nodes will continue to send their own heartbeat messages through the bridge as before.

- Occlusion reasoning (see Algorithm 2 lines 3-13): in our scenario, the sink is not interested in historical observations about the intruders. Due to the different paths of the messages in the network, the sink often receives duplicate or obsolete messages about a given intruder. These messages are discarded, but by that time, they had used up resources in the bridge node. The *occlusion reasoning* (similar to that deployed in [2]) brings this process to the gate nodes. The gate node will delay the forwarding of messages about every intruder for a time Δt after the last report about the same intruder. During this time, the node collects reports about the intruder received from various sensing nodes and sorts them by *observation time* (a timestamp added by the observing node). After the delay, the gate node will only forward the most recent observation about the given intruder.

Although occlusion reasoning decreases the number of messages passing through the node, it introduces a delay in the transmission of the messages, and thus it should not be deployed at every regular node if the goal is to have a fast reporting of the intruders. Its application, however, is appropriate at the gate nodes due to two reasons. First, the gate nodes sit just before the bottleneck of the bridge, with a large majority of the far side messages passing through them. It is thus more likely that a gate node will be able to find opportunities for occlusion reasoning by eliminating messages coming from different nodes. Second, at the gate node the cost-benefit analysis is different, as the benefit of protecting the bridge and thus the ability of collecting data from the far side outweigh the cost of the delay in reporting.

3.4. BPA functionality at the bridge node

The bridge node is a bottleneck between the far and near side of the network: all the observations transmitted from the far side will pass through the bridge. All these messages have a common destination, the sink node, thus if the bridge node uses a deterministic routing model, such as shortest path, the next hop, which is necessarily be one of the fan-out nodes will also receive this complete load. In addition, this node must also forward messages it might receive from other near-side nodes and its own observations. Thus, this fan-out node is under even higher danger of exhausting its resources than the bridge node.

To avoid loading a single fan-out node with the complete traffic of the far side, the bridge node will split the traffic between the fan-out nodes (see Algorithm 3), performing a round robin technique when deciding which fan-out node should receive any given observation.

Figure 4 illustrates the gate, bridge and fan-out nodes after a catastrophic event and the fact that, in contrast to other nodes which forward to unique next hops, the bridge node splits its traffic over its three fan-out nodes.

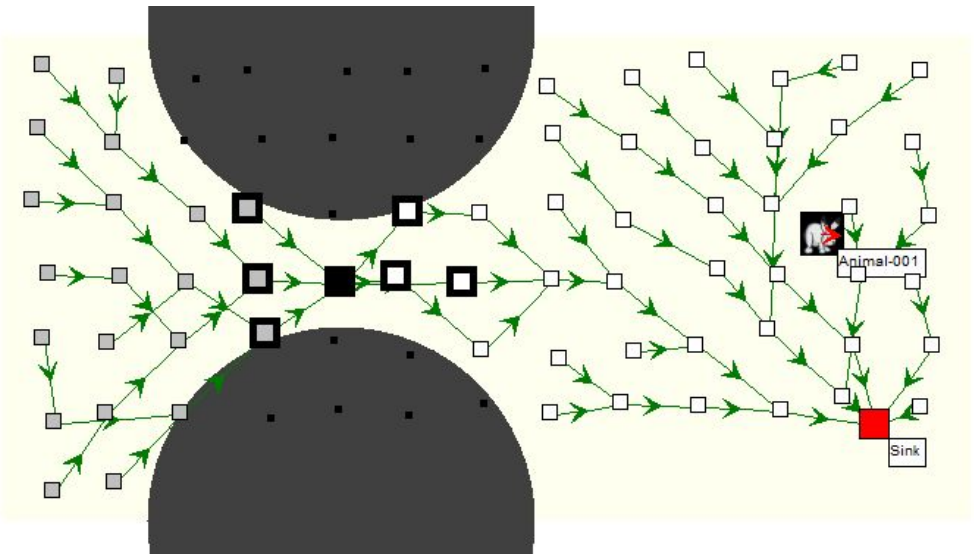


Figure 4: Forwarding paths using the Bridge Protection Algorithm after the catastrophic event. The special nodes are marked as follows: bridge solid black square, gate: black border surrounding gray square, fan-out: black border surrounding white square (YAES screenshot).

3.5. BPA functionality at the fan-out nodes

With the technique described above, the fan-out nodes receive only a fraction of the traffic of the far side. However, in large networks, where the fan-out nodes are far from the sink, there is still a danger that the far side traffic will re-coalesce at an intermediary node which was not engineered to handle large traffic. This can happen, for instance, if all the fan-out nodes forward to the same common successor. In these cases, we can replicate the load splitting behavior of the bridge at successive layers of fan-out nodes (see Algorithm 4), until we reach a desirable spread of the far side traffic. Note that a later coalescing of the far side traffic is not a problem, as in an engineered network the nodes close to the sink have the resources to handle large amounts of traffic.

4. Improving BPA with routing in the relational neighborhood graph

4.1. The impact of the transmission range

In this section we aim to improve the BPA techniques by reconsidering the underlying routing algorithm. We start by showing that a routing algorithm which is optimal for a well-deployed sensor network might not necessarily be the best choice for protecting the bridge of a network of bridged fragments.

Let us now consider the impact of the transmission range of the nodes on the shortest path routing. We will restrict our consideration to sensor nodes which have adjustable

Algorithm 1 Default behavior of a sensor node

```
1: When received intruder report m do
2:     forwardMessage(m)
3:     reset heartbeat timer
4: End When
5: When received heartbeat message m do
6:     forwardMessage(m)
7:     reset heartbeat timer
8: End When
9: When intruder i sighted do
10:    m' = new message
11:    m' = intruder identifier, location, time of sighting
12:    forwardMessage(m')
13:    reset heartbeat timer
14: End When
15: When heartbeat timer expired do
16:    m' = new heartbeat message
17:    forwardMessage(m')
18: End When
```

power levels (for instance, TelosB node allows the setting of 32 levels from -24dBm to 0 dBm) and assume that the power level is adjusted function of the transmission distance. For a given network deployment, we will consider the cases where the sensor nodes have transmission ranges of 110, 120 and 130 meters respectively. Our intuition tells us that the larger the transmission range, the better the network will perform. Indeed in Figure 5 which shows the number of messages generated as a function of the number of intruders operating in the area, the number of messages is consistently smaller for longer values of the transmission range.

Unfortunately, if we measure the energy consumption of the bridge node, we reach a different conclusion. Figure 6 shows that the energy consumption of the bridge node *increases* consistently with the increase of the transmission range. This result, which is contrary to our first intuition is due to the interaction of several facts. First, the increasing transmission distance allows nodes to access more neighbors and they would select the one that would be the farthest from themselves on the way towards the sink node. The longer distance over which the transmission is made requires more transmission energy, furthermore, due to the nature of signal attenuation, the energy increases super-linearly with the distance [3]. This energy increase might be cancelled out for the overall network by the smaller number of messages which need to be transmitted. However, the decrease in the number of messages does not apply to the bridge node, which must transmit *all* the messages from the far side of the network.

Algorithm 2 Behavior of a gate node in BPA

```
1: When received intruder report m about intruder i from node x originating in y do
2:    reset heartbeat timer for x and y
3:    p = most recent report about intruder i
4:    if p == null then
5:       set a delayed transmission timer for i
6:    end if
7:    if m.sightingTime > p.sightingTime then
8:       set m as the most recent report about intruder i
9:       discard p
10:   else
11:      discard m
12:   end if
13: End When
14: When timer for intruder i expired do
15:   m = most recent report about intruder i
16:   forwardMessage(m)
17: End When
18: When received heartbeat message m from node x originating in y do
19:   reset heartbeat timer for x and y
20: End When
21: When heartbeat timer expired for x do
22:   m = heartbeat failed for x
23:   forwardMessage(m)
24: End When
25: // other events handled as default
```

Algorithm 3 Behavior of a bridge node in BPA

```
1: When received intruder report m about intruder i do
2:    choose next hop f from the fan-out nodes using round-robin
3:    forward m to f
4:    reset heartbeat timer
5: End When
6: // other events handled as default
```

Algorithm 4 Behavior of a fan-out node in BPA

1: **When** received intruder report m about intruder i **do**
2: choose next hop f using the load splitting routes (eg. none or BPA-RNG)
3: forward m to f
4: reset heartbeat timer
5: **End When**
6: // other events handled as default

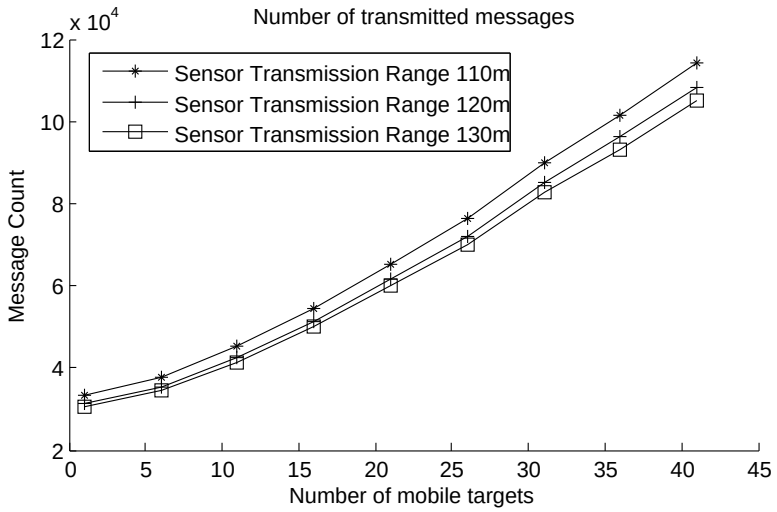


Figure 5: The number of messages function of the number of intruders for transmission range values of 110, 120 and 130 meters respectively for shortest path routing.

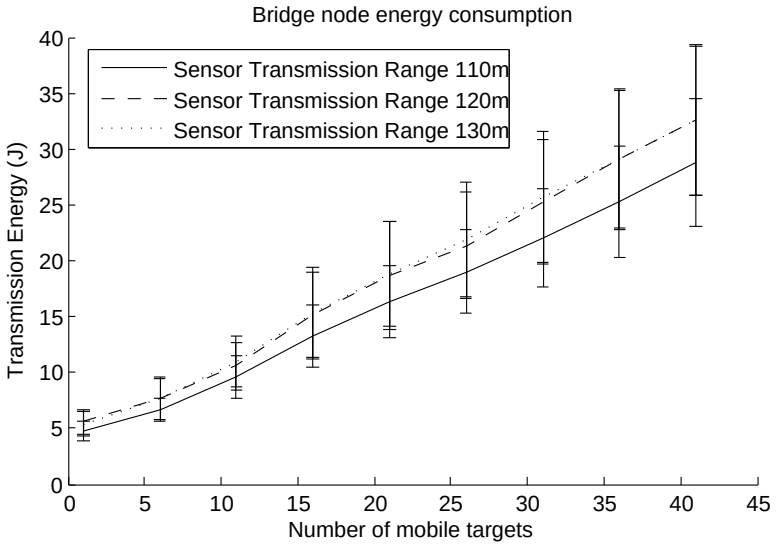


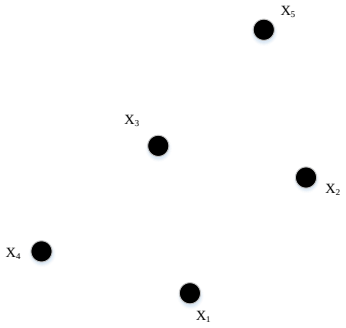
Figure 6: The energy consumption of the bridge node function of the number of intruders for transmission range values of 110, 120 and 130 meters respectively for shortest path routing.

In conclusion, as most sensor networks actually use nodes with transmission ranges longer than the minimum necessary to ensure connectivity, they will behave sub-optimally with respect to the protection of the bridge.

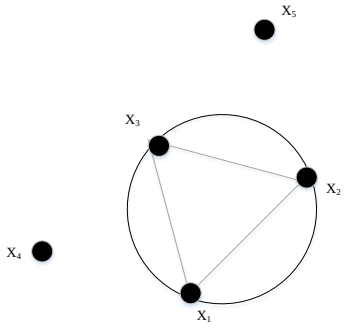
4.2. A computational geometry-based solution: routing in the relative neighborhood graph

In the following we will describe an approach to reduce the graph over which the routing takes place by using techniques of computational geometry. We start from the concept of the Delaunay triangulation [4] of a set of points X, which is a subdivision of the plane into triangles in such a way that no point will be in the circumcircle of any of the triangles. Figure 7 shows several examples which illustrate what qualifies and what doesn't qualify as a Delaunay triangle. Intuitively, Delaunay triangulation tries to avoid "skinny" triangles, in general preferring triangles with comparatively short edges for their given size.

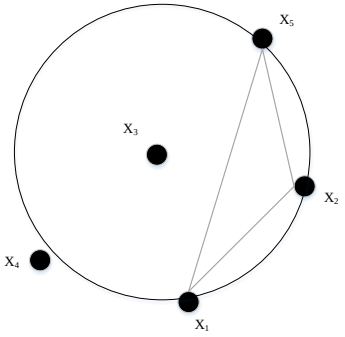
A Delauney triangulation of the nodes in a sensor network will still contain many routing alternatives. We choose to further reduce the graph by considering the relative neighborhood graph (RNG) [5], which is always a subset of the Delaunay triangulation. By definition, two nodes x_i and x_j are relative neighbors if they are at least as close neighbors to each other as they are to any other neighboring node. Hence, we obtain the relative neighbours x_i and x_j $\forall\ i, j = 1, 2, \cdots n, i \neq j \iff x_i$ and x_j. This means that for two-dimensional space where $R = 2$, the RNG $G = (V, E)$ having relative neighbors



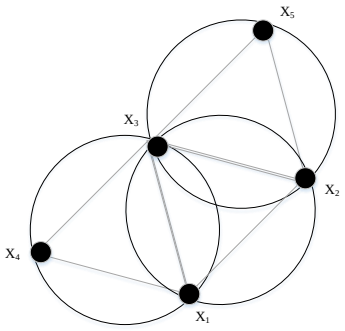
(a) The sensor network nodes labeled from $X_1 \cdots X_5$



(b) A valid delaunay triangulation between the nodes

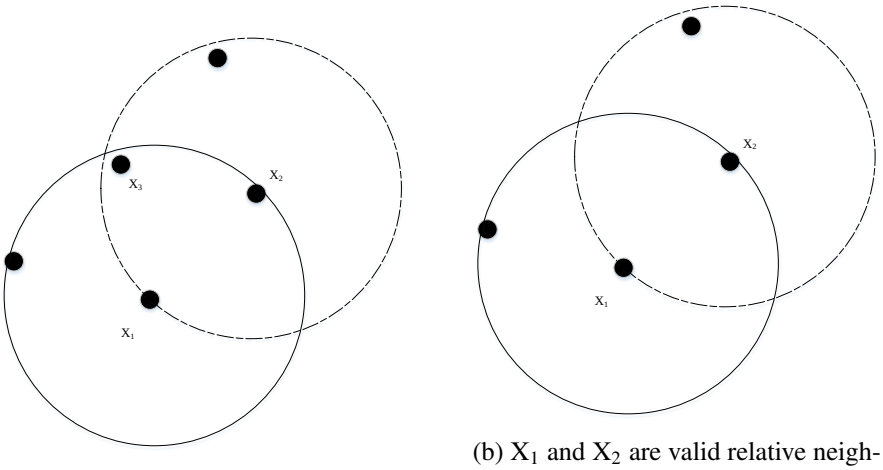


(c) A non-valid triangulation between the nodes



(d) Complete formation for delaunay triangulation between the nodes

Figure 7: Examples of configurations which qualify and which does not qualify as Delauney triangles.



(a) X_1 and X_2 are not relative neighbors as X_3 is inside their lune.

(b) X_1 and X_2 are valid relative neighbors as no other node exists inside their lune.

Figure 8: The lune shape geometry for relative neighborhood graph nodes.

x_i, x_j is given as:

$$(x_i, x_j) \in E \iff \forall x_z \in V \setminus \{x_i, x_j\} \Rightarrow d(x_i, x_j) > d(x_i, x_z) \wedge d(x_i, x_j) > d(x_j, x_z) \quad (1)$$

Using Equation 1 the nodes x_i and x_j will only be relative neighbors if the *lune* of node x_i and x_j does not contain any other node x_z. The geometrical lune shape for the neighbors in the RNG graph can be seen in Figure 8.

Algorithm 5 shows the pseudo-code for the creation of the relative neighbor graph for a specific node N_u of the sensor network. The algorithm for Relative Neighborhood Graph (RNG) consists of two steps. First, we create the Delaunay triangulation of the network. For this, initially we need to find all the neighbors at 1-hop distance from the node N_u and initialize the 1-hop neighbor list (Q) for the node N_u (Line 2-6). The next step is to calculate the lune shapes between the relative neighbors according to Equation 1. Afterwards we need to delete those edges for $v \in V$ which are within the lune of relative neighbors. Hence, the sensor node N_u calculates its distance to neighbor N_s and checks whether another neighbor node N_x exists within the lune of N_u and N_s (Line 9). If N_x does exist within the lune, then N_u discards N_s as N_s does not satisfies the RNG graph property (see Equation 1 and Line 10). This process of neighbor selection continues for all of 1-hop neighbors of N_u and in the end we get a list of neighbors that satisfy the relative neighbor graphy property, *i.e.*, no third node can exist within the lune of two connected nodes.

The RNG is a subset of the original connectivity graph of the sensor network which,

Algorithm 5 Relative Neighbor Selection For Disjoint Paths For Node N_u

```
1: Q ← ∅
2: for all N_s ∈ sensorNetwork do
3:    if connected [N_u, N_s] then
4:       Q = Q ∪ N_s
5:    end if
6: end for
7: for all N_s ∈ Q do
8:    for all N_x ∈ Q & N_x ≠ N_s do
9:       if dist[N_u, N_s] > max(dist[N_u, N_x], dist[N_s, Nx]) then
10:         Q = Q − N_s
11:      end if
12:   end for
13: end for
14: return Q
```

in general, favors the shorter edges while still maintaining the connectivity of the graph. We implement a routing algorithm called BPA-RNG which differs from BPA-SP by the fact that the underlying routing will be performed exclusively in the RNG graph. In general, we expect that BPA-RNG will have longer paths than BPA-SP, but that it will have a lower energy consumption at the bridge and fan-out nodes, and thus extend the life of the network.

5. Simulation study

5.1. Simulation scenario

In the following we describe a series of experiments which compare the performance of BPA-SP and BPA-RNG with a baseline shortest path routing. The algorithms and the experimental scenario have been implemented in the YAES simulator [6]. Table 1 summarizes the simulation parameters. The sensor network consists of 80 nodes distributed in an interest area of 1000 x 500 meters. The nodes are deployed using a "grid with noise" arrangement (see discussion in Section 2.1), with a Gaussian noise with the standard deviation of 0.1 times the length of the grid edge being added to the coordinates. We consider the sensor network deployment to track the presence of 5-30 intruders. We assume that the intruders perform a random waypoint movement with a speed of 5-15 m/s. As we discussed in Section 2, the sensor nodes report the movement of the intruders to the sink and, if no reports are sent for a time of 10s, they send a heartbeat message. We use the energy dissipation model from Rappaport [3].

General settings	
Number of nodes	80
Distribution area	1000m x 500m
Interest area	1000m x 500m
Sensing range	150m
Transmission range	140m
Sink node location	(1100, 600)
Heartbeat message interval	10s
Simulation time	100 sec
Transmission power model	
Path loss index n	4
α_{11}	45 nJ/bit
α_2	0.001 pJ/bit/m^4
Intruder Specifications	
Intruder speed	5m/s - 15m/s
Number of intruders	5 to 30
Catastrophic events	
Event 1	t=5, circular area, range 400, center (375, 195)
Event 2	t=5, circular area, range 400, center (375, 450)

Table 1: The parameters of the simulation experiments

5.2. Catastrophic events, recovery and bridge protection

As we are interested in the behavior of the network in response to a catastrophic event which leaves it as a network of bridged fragments, we have considered a scenario where the network starts up with shortest path routing, but very early in the scenario (at t_k = 5s) the simultaneous occurrence of two catastrophic events (as seen in Figure 3) transform it into a network of bridged fragments[3]. As an answer to this, the network responds by transitioning to one of the following routing algorithms:

SP: shortest path routing over the recalculated graph. In this baseline scenario the only response of the network is to recompute the shortest path routes, over which it would route messages as before without any specific bridge protection methods.

BPA-SP: bridge protection algorithm over shortest path. In this scenario, the routing tables are recalculated using a shortest path model and the bridge, gate and fan-out nodes are identified. The nodes use the bridge protection techniques described in Section 3.

BPA-RNG: bridge protection algorithm over the relative neighborhood graph. In this scenario, we are using all the bridge protection algorithms as described in Section 3 but the routing tables are calculated over the relative neighborhood graph as described in Section 4.

The identification of bridge, gate and fan-out nodes of the network is done at the sink, using graph theoretic techniques. If the routing algorithm is a centralized one, the sink already has the necessary information. If the routing algorithm is a decentralized one, the nodes will transmit their neighborhood information to the sink in their first heartbeat message after the catastrophic event. The sink will notify the nodes of their status as bridge, gate or fan-out node through the return path a trivially low overhead of only one message per special node.

5.3. Behavior of BPA-RNG for different transmission ranges

Our main motivation for developing alternative techniques to BPA-SP was the observation that the shortest path routing behaves in a suboptimal way when the transmission range exceeds the required minimum to keep the network connected. In particular, we found that as the transmission range was increasing, the number of messages were decreasing, but the energy expenditure of the bridge node was increasing.

In this first experiment, we verify that the BPA-RNG algorithm provides a more consistent behavior when the transmission range is increased. Figure 9 shows the number of messages transmitted network-wide function of the number of intruders. The three graphs correspond to the BPA-RNG routing technique, for transmission ranges of 110, 120 and 130 meters respectively. We find that the number of transmitted messages vary

[3]Note that this is not the only possibility of the creation of a scenario with bridged fragments. It is possible that the two events occur at different points in time or that a bridge is created by a single circular event occurring near the boundary of the sensor deployment.

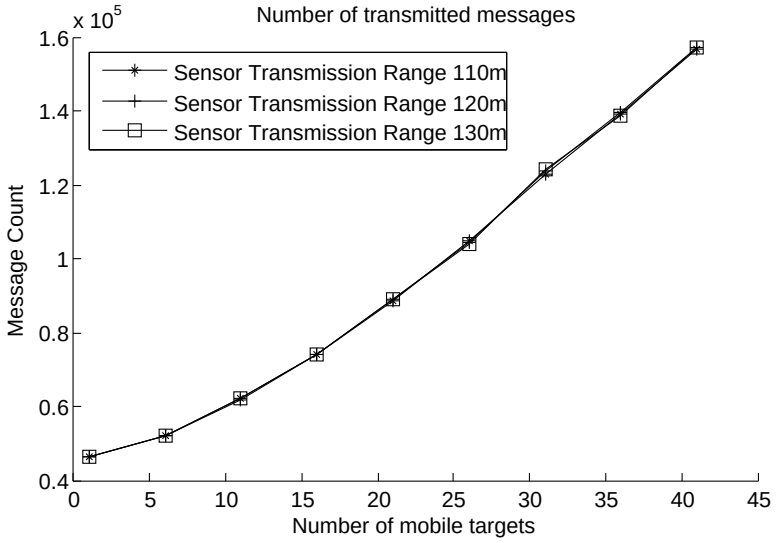


Figure 9: The total number of transmitted messages function of the number of intruders for transmission range values of 110, 120 and 130 meters respectively for a routing model using BPA-RNG algorithm.

very little as a function of the transmission range (in contrast to the BP algorithm in Figure 5) where the number of messages varies significantly with the transmission range.

5.4. Total transmitted messages

In this experiment we compared the total number of transmitted messages for the three algorithms as a function of the number of intruders. The results of the experiment are shown in Figure 10.

We find that the BPA-RNG algorithm sends the largest number of messages, followed by SP and BPA-SP. Albeit SP and BPA-SP are using the same routing paths, the BPA-SP sends fewer messages due to the processing performed by the gate nodes. While this processing is also performed for BPA-RNG, the overall number of messages is still higher as the different routing tables prefer paths which might be longer in terms of the number of hops but consist of shorter individual hops.

5.5. Bridge node energy consumption

In this experiment, we compared the energy consumption of the bridge node function of the number of intruders for the three considered algorithms (see Figure 11). In a network of bridged fragments with a bridge node with a limited energy resource, this value directly determine the moment when the data collection from nodes on the far side is lost.

Overall, as expected, the energy consumption of the bridge node is increasing with the number of intruders for all protocols. In general, the BPA algorithm significantly

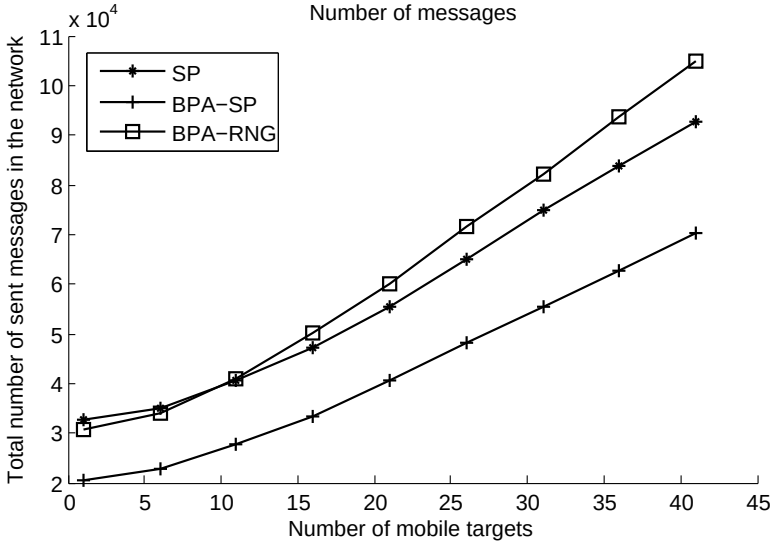


Figure 10: The total number of transmitted messages function of the number of intruders for the SP, BPA-SP and BPA-RNG algorithms.

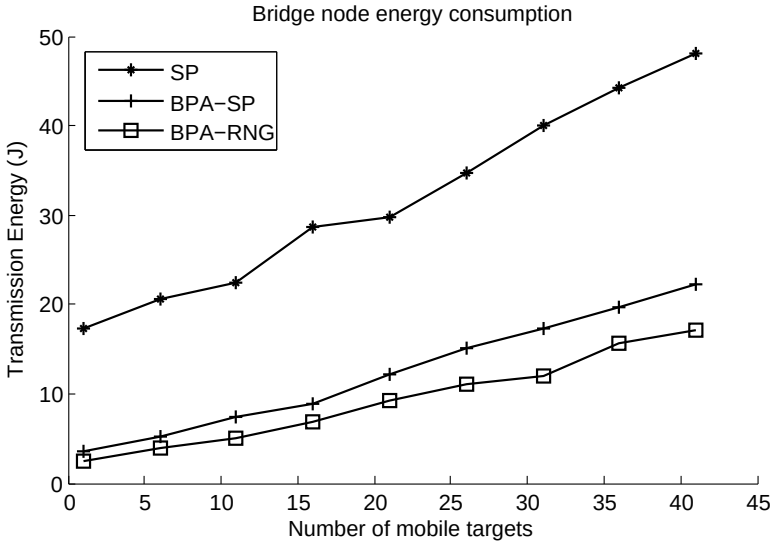


Figure 11: The energy consumption of the bridge node function of the number of intruders for the SP, BPA-SP and BPA-RNG algorithms.

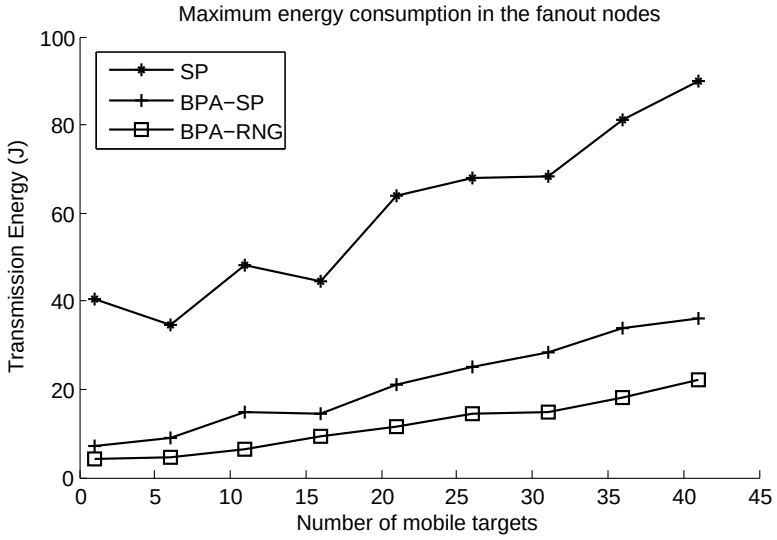


Figure 12: The highest fan-out node energy consumption function of the number of intruders for the SP, BPA-SP and BPA-RNG algorithms.

lowers the energy consumption of the bridge node for both the BPA-SP and BPA-RNG algorithms. In addition, the BPA-RNG algorithm further reduces the energy consumption by about 25% in the specific experiments.

5.6. Maximum energy consumption of a fan-out node

As we noted before, it is not only the bridge node which is in danger to exhaust its energy resources too soon due to overload, but also the fan-out nodes, especially in the case of SP where the traffic of the far side is concentrated to one fan-out node. It is thus of interest to investigate the energy consumption of the fan-out node with the highest energy consumption. The results of the experiment are shown in Figure 12.

In general, we find that the fan-out node energy consumption increases with the number of intruders for all techniques. There are, however, accidental variations - for instance, if most intruders happen to be in the near-side of the bridged fragments, then those intruders will not contribute to the energy consumption of the fan-out nodes.

We find that the bridge protection techniques significantly reduce the fan-out energy consumption - both BPA-SP and BPA-RNG have significantly lower values. Between these two values, BPA-RNG has a significant advantage, especially in scenarios of more than 15 intruders, where BPA-RNG consistently consumes only about half of the energy of BPA-SP at the fan-out nodes.

5.7. Network life extension ratio

The objective of the BPA algorithms is to extend the life of the network. Considering that energy exhaustion of the bridge node is the most likely cause of the network to be-

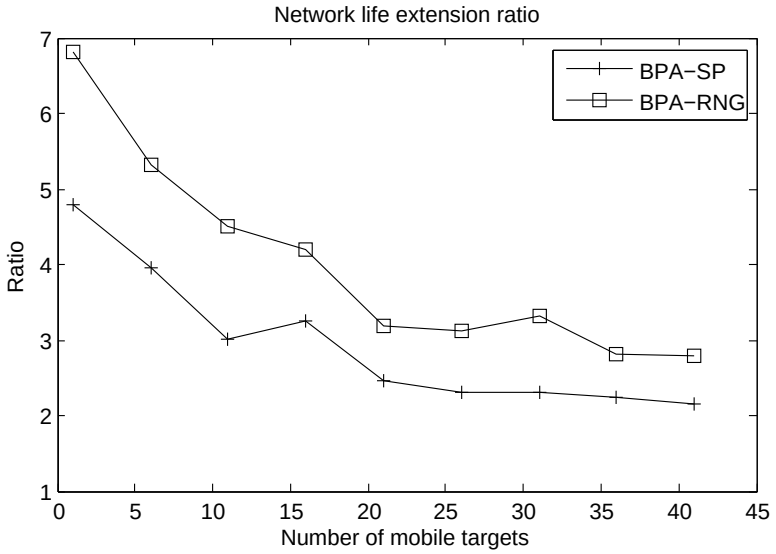


Figure 13: The relative increase of the network lifetime for BPA-SP and BPA-RNG compared to the SP algorithm.

come disconnected, the best way to illustrate the benefits of the BPA-SP and BPA-RNC algorithms is to study how much longer a network can maintain connectivity compared to the basic algorithm. (Note that this is a relative value - the absolute number depends on the remaining energy in the bridge node at the moment of the catastrophic event, a value which is not under the control of the networking algorithms).

Figure 13 shows the relative increase of the network lifetime for BPA-SP and BPA-RNG compared to the SP algorithm. Overall, we find that ratio of the life extension decreases with the number of intruders. However, even for 40 intruders tracked, BPA-SP extends the lifetime about 2 times, while BPA-RNG about 3 times. If the number of intruders is low, these values are significantly higher: up to about 5 times for BPA-SP and about 7 times for BPA-RNG.

6. Related work

Early sensor network literature considered that the fragmentation of the sensor network (due to the exhaustion of the energy resources of a group of nodes) represents the end of the life-cycle of the network [7]. Although system wide algorithms have been designed to postpone fragmentation, for instance, by energy aware routing, there was little consideration given to what can be done if the fragmentation already happened (or is about to happen).

In recent years, however, a series of papers have been investigating the problem of *federated sensor networks* - systems whose topology is either separated in disconnected

graphs or it is connected with weak, narrow and/or intermittent connection. Existing work in the area can be grouped into two distinct approaches.

The first approach proposes the linking of the federated networks using *mobile nodes*. One of the earliest approaches is the data mule architecture of Shah et al. [8] where randomly moving mobile nodes (mules) transport data among the nodes of a sparsely connected network. Recent work developed by Basagni et al. [9, 10] use linear optimization techniques and heuristics to maximize the lifetime of the sensor network.

Almasaeid and Kamal [11, 12] use mobile relay nodes (MRs) to act as data relays between fragments of a sensor network which became fragmented. The authors use steady state probabilities to model the delays between the fragments and the delays between fragments-to-sink. Factors such as the sojourn time: time for the MRs to stay in between fragments to relay data is also taken into account for the optimized movement policy of the MRs. Mathematical modeling of the movement policy was proposed using closed queueing network. Evaluating its performance using TOSSIM [13] the authors emphasize the impact of sojourn time and other factors such as count and speed of MRs for the fragmented network data delivery.

In the work by Abbasi et al. [14, 15], the recovery of a fragmented network is performed by moving some of the existing nodes to positions where they can reconnect the fragments and provide connectivity at a specific level (one or two-connectivity). These techniques can be seen as hybrids between the mobile node-based and the relay node-based approaches. The same problem is investigated by Akkaya et al. [16], and Imran et al. [17, 18, 19].

Zhao et al. [20] describe an approach where a set of special nodes called message ferries are providing communication services to networks of nodes (which can be themselves mobile). The paper describes two different approaches depending on whether the movement is initiated by the nodes (nodes move close to ferries in order to communicate) or whether the ferries pro-actively move to meet the nodes.

In contrast to these approaches which consider that the mobility of the specific nodes is explicitly designed to address the connection of the network fragments, opportunistic networking (Pelusi et al.[21]) designs routing protocols to take advantage of opportunities created by moving nodes to bring the transmitted data closer to the destination. In these systems, it is possible for messages to reach their destination even if there is no moment in time when a fully connected route exists between the source and destination.

The idea of using cascaded node movement has also been proposed to avoid possible fragmentation by Wang et al. [22]. The idea of cascaded network movement works in two phases. The first phase is to locate a nearby sensor node (near the failure node) that has low activity and is suitable for replacement. The second phase is to use cascaded movements between sensor nodes to rearrange the network. Instead of having the direct movement of redundant sensor node to the point of fragmentation, the nearby nodes relocate themselves to cover the fragmented area.

Another approach to federated sensor networks investigates how the federations can be connected using a number of nodes called *relays*. Relay nodes might have special properties, such as longer range or higher energy resources. The challenge is to choose the location of the relay nodes such that connectivity, and possibly, certain quality of service criteria are achieved with a minimum number of nodes.

Cheng et al. [23] show that even the simplest possible formulation of the relay node problem (asking only for the minimum number of relay nodes) is equivalent to the NP-hard problem.

Hou et al. [24] consider the response of the network operator to the fragmentation of the network, which can be a combination of deploying new relay nodes and adding additional energy resources to existing nodes. The resulting joint problem of energy provisioning and relay node placement can be formulated as a mixed-integer nonlinear programming problem. These class of problems being NP-hard, the authors propose a heuristic approach which transforms the problem into a linear programming problem, without loosing important points of the search space.

Lee and Younis [25] solve the relay node placement problem in a network where the requirement is not only the maintenance of connectivity but also a series of quality of service requirements. As the problem is NP-hard, the proposed approach OQAP (Optimized QoS Aware Placement of relay-nodes) pursues a greedy heuristics while modeling the network as a grid.

The bridge protection algorithm described in this paper considers a scenario where the federations are connected using a very narrow and vulnerable link. Instead of considering the situation after the fragmentation of the network into federations, BPA considers a network close to fragmentation, and changes the behavior of the nodes in such a way that they protect the bridge nodes, postponing, as long as possible, the fragmentation of the network.

The BPA algorithm complements, rather than replaces, existing federated sensor network technologies. In our running scenario, we have defined the bridge nodes as the remaining nodes which maintain connectivity after a catastrophic event. However, bridge nodes can appear in a different way as well: from the relay nodes introduced by the relay node placement algorithms. In fact, if a minimal number of relay nodes are chosen, these nodes will, by definition, be bridges. The BPA algorithm, applied in tandem to a relay node placement algorithm, can maximize the benefit of the repair, and postpones the necessity of additional repairs in the future.

7. Conclusions

In this paper we considered the case of an intruder tracking sensor network faced with a catastrophic event which result in a network topology of bridged fragments. Such networks are vulnerable as the exhaustion of the energy resources of a bridge node can disconnect large fragments of the network. We have introduced two routing protocols

BPA-SP and BPA-RNG which protect the bridge by imposing a differentiated behavior on nodes with special roles in the topology. Through a simulation study, we have shown that the proposed protocols can extend the connected lifetime of the network 2-7 times over the default shortest path routing.

References

[1] L. Bölöni, D. Turgut, Protecting bridges: reorganizing sensor networks after catastrophic events, in: Proc. of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC-2011), 2011, pp. 2028–2033.

[2] D. Turgut, B. Turgut, L. Bölöni, Stealthy dissemination in intruder tracking sensor networks, in: Proc. of IEEE Local Computer Networks (LCN 2009), 2009, pp. 22–29.

[3] T. Rappaport, Wireless Communications: Principles & Practice, Prentice-Hall, 1996.

[4] D.-T. Lee, A. K. Lin, Generalized Delaunay triangulation for planar graphs, Discrete & Computational Geometry 1 (1) (1986) 201–217.

[5] K. J. Supowit, The relative neighborhood graph, with an application to minimum spanning trees, Journal of the ACM (JACM) 30 (3) (1983) 428–448.

[6] L. Bölöni, D. Turgut, YAES - a modular simulator for mobile networks, in: Proc. of the 8-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2005), 2005, pp. 169–173.

[7] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer Networks 52 (12) (2008) 2292–2330.

[8] R. Shah, S. Roy, S. Jain, W. Brunette, Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks, Ad Hoc Networks 1 (2-3) (2003) 215–233.

[9] S. Basagni, A. Carosi, C. Petrioli, Heuristics for lifetime maximization in wireless sensor networks with multiple mobile sinks, in: Proc. of the IEEE ICC'09, 2009, pp. 1–6.

[10] S. Basagni, A. Carosi, C. Petrioli, C. A. Phillips, Coordinated and controlled mobility of multiple sinks for maximizing the lifetime of wireless sensor networks, Wireless Networks 17 (3) (2011) 759–778.

[11] H. Almasaeid, A. Kamal, Data delivery in fragmented wireless sensor networks using mobile agents, in: Proc. of the ACM Int'l Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2007), 2007, pp. 86–94.

[12] H. Almasaeid, A. Kamal, Modeling mobility-assisted data collection in wireless sensor networks, in: Proc. of the IEEE Global Communications Conference (GLOBECOM-2008), 2008, pp. 1–5.

[13] P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: Accurate and scalable simulation of entire TinyOS applications, in: Proc. of the 1st International Conference on Embedded Networked Sensor Systems, 2003, pp. 126–137.

[14] A. A. Abbasi, K. Akkaya, M. Younis, A distributed connectivity restoration algorithm in wireless sensor and actor networks, in: Proc. of 32nd IEEE Conference on Local Computer Networks (LCN 2007), 2007, pp. 496–503.

[15] A. A. Abbasi, M. Younis, K. Akkaya, Movement-assisted connectivity restoration in wireless sensor and actor networks, IEEE Transactions on Parallel and Distributed Systems 20 (9) (2009) 1366–1379.

[16] K. Akkaya, F. Senel, A. Thimmapuram, S. Uludag, Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility, IEEE Transactions on Computers 59 (2) (2010) 258–271.

[17] M. Imran, M. Younis, A. Md Said, H. Hasbullah, Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks, in: Proc. of IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS 2010), 2010, pp. 679–683.

[18] M. Imran, M. Younis, A. M. Said, H. Hasbullah, Partitioning detection and connectivity restoration

algorithm for wireless sensor and actor networks, in: Proc. of the 8th IEEE/IFIP international conference on embedded and ubiquitous computing (EUC 2010), 2010, pp. 200–207.

[19] M. Imran, M. Younis, A. Md Said, H. Hasbullah, Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks, Journal of Network and Computer Applications 35 (2) (2012) 844–856.

[20] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2004), 2004, pp. 187–198.

[21] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Communications Magazine 44 (11) (2006) 134–141.

[22] G. Wang, G. Cao, T. La Porta, W. Zhang, Sensor relocation in mobile sensor networks, in: Proc. of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM-2005), Vol. 4, 2005, pp. 2302–2312.

[23] X. Cheng, D. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, Wireless Networks 14 (3) (2008) 347–355.

[24] Y. Hou, Y. Shi, H. Sherali, S. Midkiff, On energy provisioning and relay node placement for wireless sensor networks, IEEE Transactions on Wireless Communications 4 (5) (2005) 2579–2590.

[25] S. Lee, M. Younis, QoS-aware relay node placement in a segmented wireless sensor network, in: Proc. of the IEEE Int'l Conference on Communications (ICC 2009), 2009, pp. 1–5.