Protecting bridges: reorganizing sensor networks after catastrophic events

Ladislau Bölöni and Damla Turgut
Dept. of Electrical Engineering and Computer Science
University of Central Florida
Orlando, Florida 32816-2362
Email: {lboloni,turgut}@eecs.ucf.edu

Abstract— **The worst case scenario for the life cycle of a sensor network is the fragmentation of a network which still has many functional and well-powered nodes. The loss of connectivity renders even the functional nodes useless as the nodes are not able to transmit their observations to the sink. A well-engineered sensor network will *not* fragment due to the energy consumption occurring during normal functioning. However, catastrophic events, which unpredictably destroy a large subset of the nodes, can transform a well engineered network into a heavily unbalanced one. Very often, even if the network is not yet fragmented, the connectivity is relying on one or more *bridge* nodes, which survived the catastrophic event accidentally. If the network operates as before, the bridge nodes will soon exhaust their power resources by having to route an unexpectedly large number of packets.**

This paper describes the 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, BPA sacrifices the length of some routes in order to distribute routes away from critical areas.

Through a simulation study we show that the application of these techniques can significantly decrease the load of the nodes in the critical areas, while only minimally affecting the performance of the network.

Keywords–bridge node; intruder tracking; federated; sensor networks

I. 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.* 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. It is true that the energy consumption in a hop-by-hop network will be heterogeneous: nodes closer to the sink, as well as nodes along the main forwarding trunks will have a higher energy consumption. However, this higher energy consumption can be predicted and the operator of the network can, for instance, provide such nodes with a larger battery.

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 domains, linked by *bridge nodes*. We define the bridge node as a node whose removal disconnects the network[1].

In contrast to nodes which have been engineered to handle a high load, bridge nodes are simple 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.

The bridge node faces the same threats as like other nodes: energy exhaustion, accidental environmental damage, and opponent activity. We cannot do anything about accidental damage. For the other two categories of danger, we can offer the following considerations:

Energy exhaustion: in general, the energy consumption of the nodes increases with the sensing, computation and networking capabilities. The bridge nodes have significantly higher networking responsibilities than any other node: they need to transport the complete traffic of the fragment: sensing reports and status reports from the nodes to the sink and commands from the sink to the nodes. Thus, not only the failure of a bridge node is more damaging, but bridge nodes will also exhaust their energy faster.

Opponent action: the goal of the opponent is to destroy the sensing capability of the operator of the sensor network. The more transmission activity a node performs the more likely that the opponent can detect it (for an analysis of the impact of transmission on the lost of stealth see [1]). Furthermore, an active opponent who invests resources into locating and

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

disabling sensor nodes, will likely reason that nodes with a higher traffic play a more important role.

We can conclude that the bridge nodes are both more important for the quality of sensing and they are also under additional threat. In this paper, we describe the Bridge Protection Algorithm (BPA), a series of techniques which form a coherent but localized response of the network to a catastrophic event which created a network topology with one or more bridges. 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 II presents related work. Section III describes the bridge protection algorithm. Section IV describes the results of a simulation study comparing the performance of BPA with the baseline response of a sensor network to a catastrophic event. We conclude and propose future work in Section V.

II. 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 [2]. 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. [3] where randomly moving mobile nodes (mules) transport data among the nodes of a sparsely connected network.

Almasaeid and Kamal [4], [5] use mobile agents to act as data relays between fragments of a sensor network which became fragmented.

Zhao et al [6] 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 [7]) 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.

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. [8] 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 *Steiner Minimum Tree with Minimum number of Steiner Points and bounded edge length (SMT-MSP)*. The authors proceed to show that polynomial approximation algorithms are possible, albeit with relatively unfavorable performance ratios of 2.5.

Hou et al. [9] 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 [10] 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.

Finally, in the work by Abbasi et al. [11] and Akkaya et al. [12] 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 reader can refer to Younis and Akkaya [13] for a complete survey of node placement techniques in sensor networks.

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.

III. Bridge protection algorithm

The bridge protection algorithm changes the behavior of a set of topologically important nodes in response to a catastrophic event. These changes are done with respect to a *baseline algorithm*, which describes the forwarding behavior of the network in the absence of the catastrophic event. Any algorithm which establishes a deterministic forwarding architecture can serve as baseline. In this paper we assume that the baseline algorithm creates a shortest part forwarding, but we will not make assumptions on the specific technique through which this is achieved. We will also assume that the baseline algorithm is capable to repair the forwarding paths after the catastrophic event.

In the following we first describe the considered sensing task and the physical network. We follow with a discussion of the baseline algorithm and its response to the catastrophic event. Finally we describe the changes introduced by bridge protection algorithm.

A. The sensing task and the physical network

The sensing task considered in this paper is one of intruder detection and tracking inside an *interest area* which is usually assumed to be rectangular. Sensor nodes are deployed in and around the interest area.

In contrast with some of the early assumptions about sensor networks, which predicted that sensor nodes would become so cheap that they can be simply dispersed from airplanes, most of the modern intruder detection systems assume an *engineered deployment:* the nodes have been deployed individually at carefully chosen locations, with the explicit goal to protect the area. The ideal arrangement of nodes would be on a regular grid (rectangular or hexagonal). 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.

In practice, however, environmental conditions (e.g. the obstacles and camouflage opportunities in the environment) make the achievement of a perfect grid unfeasible. The deployer 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 nodes detect intruders in their sensor range and send their observations with a hop-by-hop approach to the *sink node*. In the scenario we are considering, the sink node is situated outside the interest area.

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 to know whether an intruder is inside the interest area or not, and if it is inside the interest area, about its most recent location. The sink is also interested in independent confirmations of the locations of a certain intruder. The intruders all start from the outside of the network and follow random waypoint mobility model.

Although simple, this policy has several important practical consequences. If an intruder moves outside the interest area, the sensor node will send exactly one transmission reporting that the node left the interest area. It will continue to track the node, as long as it is in the sensor range, but it will not report its location, unless the node enters the interest area (as the sink is not interested in intruders outside the interest area). If the node makes several successive observations, but they are scheduled to transmit only at certain time intervals, the node will transmit only the most recent observation (as the sink is not interested in historical information). The node will, however, not perform occlusion reasoning between observations of different sensors in the style of TAB[1], *i.e.* it will not discard previous observations made by different sensors if they have a newer observation.

We will make the assumption that the nodes will transmit their own observations at fixed time intervals, but immediately forward other nodes transmissions.

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 certain intervals when they do not have anything else to send. Any reported observation automatically replaces the heartbeat signal. The interval between heartbeat signals is an order of magnitude larger than the interval between successive intruder reports.

B. The baseline algorithm for sensing, dissemination and recovery

The BPA algorithm intruduces changes with respect to a baseline algorithm. We assume that the baseline algorithm creates a deterministic routing tree, with the root of the tree in the sink. In this case deterministic means that in the absence of external events changing the network, all the packets will be forwarded along the same route to the sink. We will assume that the routing tree is based on the shortest path (with the length of the paths measured in hops). Figure 1 shows an example of such a routing tree in the environment considered.

We will assume that the behavior of the nodes follows the following rules:

- if there is no sensed intruder in the sensing range, the nodes send a heartbeat message event t_{hb} interval. (Default value $t_{hb} = 10sec$).
- if there are one or more intruders in the sending range, the nodes transmit the sensed data at every $t_s < t_{hb}$ interval. (Default value $t_s = 1sec$).
- nodes forward received messages towards the sink.

It must be obvious that this algorithm leads to a non-homogeneous consumption of the energy resources of the

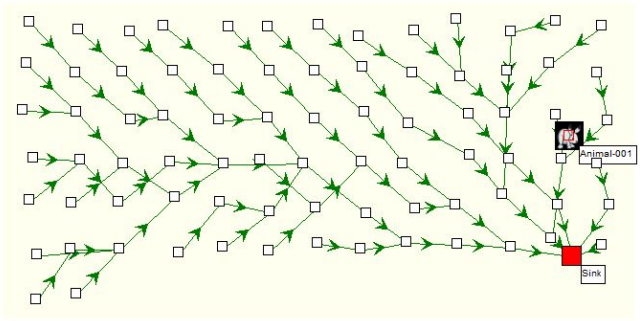

Fig. 1. Forwarding paths in the baseline algorithm before the catastrophic event (YAES screenshot).

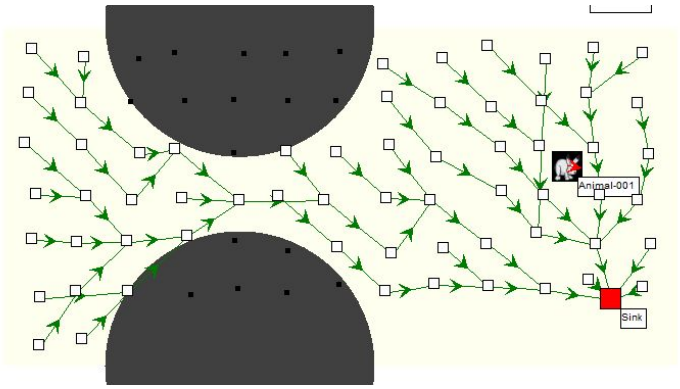

Fig. 2. Forwarding paths reconfigured by the baseline algorithm after the catastrophic event (YAES screenshot).

nodes. Nodes closer to the sink, and more likely to be on the shortest path will carry more traffic and thus consume more energy. Such differences, however, are predictable and compensating for them is part of the correct engineering of the network.

There are also unpredictable differences: for instance, nodes which see many intruders will consume more energy than nodes which only need to transmit occasional heartbeat messages. Yet the engineering of the network can account even for such differences, by estimating the maximum number of intruders and allocating the energy resources with a safety margin.

Let us now assume that a catastrophic event disables a large number of nodes. In Figure 2 the gray areas show the area of the nodes which will not be functional. This event will cut most of the normal paths of the sensor nodes to the sink, yet the network is not, as of yet, fragmented. The two parts of the network are still joined together by a *bridge node*.

The catastrophic event has two effects on the network. The sensing impact is the loss of the sensing data of the disabled nodes. The communication impact is due to the fact that even nodes which survived the catastrophic event, might have lost their paths to the sink. The baseline algorithm's response to this is to reinitiate the creation of the shortest path routing, yielding the situation in Figure 2.

We notice the paths of all nodes from the left side pass through the bridge node. For easier reference, we will call the subgraph which is on the side of the sink from the bridge node the *near-side* while the one opposite of the bridge node the *far-side* of the network.

The bridge node needs to forward the complete set of observations made on the far side, significantly increasing its load. To be sure, the original network also had nodes which handle high loads: the nodes which are close to the sink, as the traffic of the complete network converges there. But there is an important difference: the nodes near the original sink node were known at deployment time to expect a high load, and have can be engineered in an appropriate way (with a higher energy resource, for example). The bridge node, on the other hand, is just a regular node which became a bridge node due to an accidental situation.

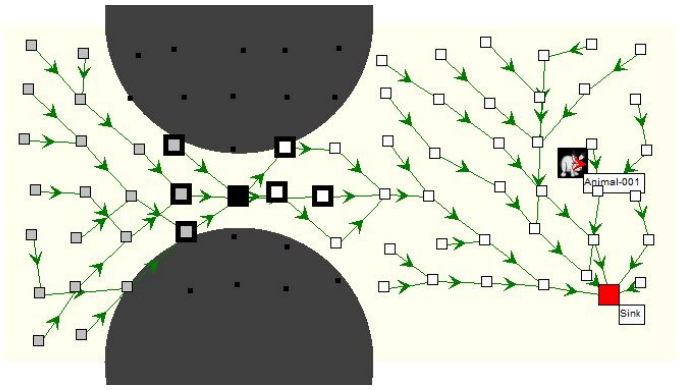

Fig. 3. Forwarding paths using the Bridge Protection Algorithm after the catastrophic event. The special nodes are marked as follows: bridge solid black square, gate: black border surrounding gray square, fan-out: black border surrounding white square (YAES screenshot).

C. The bridge protection algorithm

The bridge protection algorithm has been designed with the following design principles in mind:

- The protection of the bridge: the algorithm needs to protect the bridge node from failure through the exhaustion of its energy resources.
- The protection of the potential bridge nodes: the sensor network affected by the catastrophic event will also have nodes which, although currently not bridge nodes, can become one if one or more node fails.
- Minimal intrusion: our goal is to change only the behavior of a small number of nodes, letting the remainder of the nodes operate as before.
- Maintaining functionality: as much as possible, the network should maintain the existing functionality.

The BPA algorithm identifies and changes the functionality of three classes of nodes: the bridge nodes, the *gate nodes* defined as the far-side immediate neighbors of the bridge nodes and the *fan-out nodes* defined as the near-side immediate neighbors of the bridge nodes (see Figure 3).

Gate nodes: The objective of the gate is to reduce the traffic reaching the bridge node. The gates use two techniques to protect the bridge nodes:

- **Converting heartbeat:** collect but do not send the heartbeat messages. The gate nodes will maintain the

state of the nodes whose paths are traversing them. If the nodes suddenly fail to transmit, the timeout will be noticed at the gate node, and a node-down message generated. The gate nodes themselves (and the bridge) will continue to generate and transmit heartbeat messages (when necessary).

- **Occlusion reasoning:** the gate nodes will perform occlusion reasoning over the received messages. They will delay the forwarding of the received messages for a time Δt and for the received messages, they will send to the bridge only the most recent received message for each intruder.

Note that both techniques can be essentially perceived as a way to push the reasoning process of the sink into the far-side. Such a reasoning is possible in the gate nodes because of the funnel effect of the bridge: all the information collected in the far side converge through the gate nodes to the bridge node. In a normal sensor network, in the absence of a catastrophic event, such a convergence would occur only at the sink.

Bridge node: The bridge node receives the (filtered) transmissions from the gate nodes. The bridge node will forward all the transmissions it receives, but doing so it must protect the fan-out nodes. A deterministic routing mechanism would choose a single forwarding node from the bridge node. This forwarding node would carry the complete load of the bridge (the complete output of the far side, although filtered by the gates) in addition to the observations made by the bridge itself, its own observations and the traffic of other near-side nodes routing through it. This applies, recursively, to all the nodes along the shortest path from the bridge to the sink. Although topologically less critical, these nodes will be even more overloaded than the bridge and even more likely to exhaust their energy resources.

To prevent this effect, BPA requires the bridge to distribute its transmissions among the fan-out nodes. If the bridge does not have information about the available energy of the fan-out nodes, it forwards the packets in a round-robin fashion among the nodes. If such information is available, the forwarding schedule is adjusted such that each fan-out node receives a share of the traffic proportional with its energy resources.

Fan-out nodes: the fan-out nodes take the traffic from the far-side, distributed to them by the bridge node and forward them to the sink. In a grid-with-noise architecture with a diagonal density, there will be typically 1-4 fan-out nodes.

The fan-out nodes must prevent the *premature re-collapsing of the fan-out*. The bridge has taken an effort to distribute the traffic over the fan-out nodes. Yet it can happen that on their turn, the fan-out nodes will forward those packets to a common node, a node which, again, will need to handle the full load of the far side.

The prevent this, BPA enforces the rule that each fan-out node will have a different forwarding node (if possible). This can not, of course, prevent the future collapse of the forwarding into a single path. However, the closer we get to the sink, the more likely that nodes have been engineered such that they can handle a large traffic. If the bridge node is far away from the sink, BPA can request the fan-out nodes to perform another step of fan-out, just like a bridge (in effect, creating 2nd, 3rd etc order fan-out nodes).

TABLE I

THE PARAMETERS OF THE SIMULATION EXPERIMENTS

General settings	
Interest area	1000×500 m
Sensor distribution area	1000×500 m
Sensor deployment	Grid-with-noise , static
Sensor range	150
Transmission range	130
Sink node location	(1100, 600)
Simulation time	100 sec
Intruders	
Number	2 . . .40
Velocity	10±5 m/sec
Movement pattern	random-waypoint
Catastrophic event	
Event 1	t=5, circular area, range 400, center (375, 195)
Event 2	t=5, circular area, range 400, center (375, 450)

IV. SIMULATION STUDY

To study the impact of the BPA algorithm on the sensor network we have performed a simulation study using the YAES [14] simulation framework.

We have implemented the baseline algorithm with the shortest path based recovery as described in Section III-B, and the BPA algorithm as described in Section III-C. Table I describes the simulation parameters.

As we are mostly interested in the behavior of the network in response to the catastrophic event, we made the catastrophic event happen very early in the process. The tracking accuracy of the two networks will be identical, with the very minor difference that the BPA algorithm might force some packets to take a slightly longer path in the near side.

The values we shall measure are the consumed energy at various points in the network. We use the energy dissipation model from Rappaport [15]. We could transform the consumed energy into remaining energy, by simply making an assumption about the initial power resources of the node (e.g. battery size). However, as we have shown, the deployer of the network has a considerable freedom of choice in the energy resources of the individual nodes. The battery power can be easily varied accross different deployed nodes, and an engineered deployment would increase the size or number of batteries at locations closer to the sink. The expected lifetime of a specific node is thus not an issue of the network but of the engineering of the network.

The consumed energy however is entirely dependent on the protocol. We will investigate (a) the energy consumed in the bridge node and (b) the highest energy consumption among the fan-out nodes. Both in the baseline and the BPA algorithms, observations are transmitted only when an intruder is in the sensor range. In contrast, the much rarer heartbeat transmissions happen in the absence of the intruder.

As the energy consumption depends on the number of intruders, to investigate the behavior of the systems under various scenarios, we have added to the scenario a number of intruder nodes. For the purpose of the current scenario, we assumed the intruders to be wild animals, with a movement described by a random waypoint model, over an area significantly larger than the interest area. Thus the intruders will enter and leave the interest area at random locations. Naturally, an

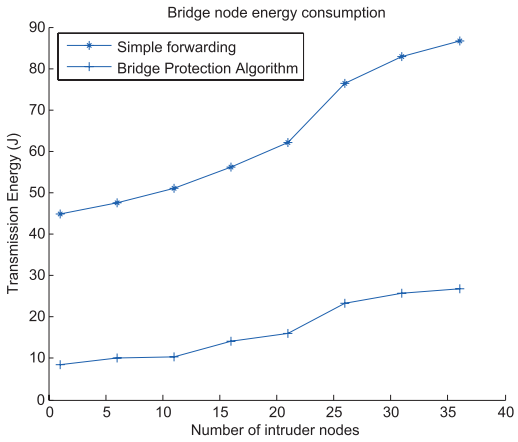

Fig. 4. The energy consumption of the bridge node

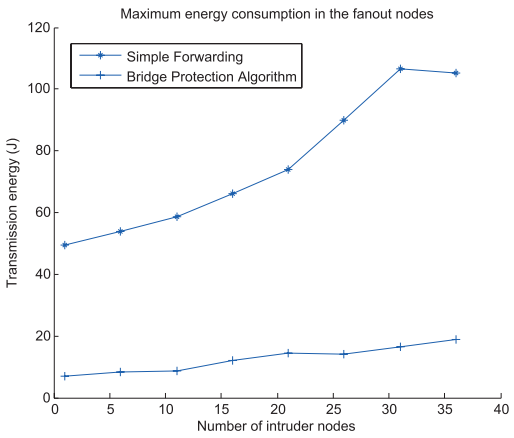

Fig. 5. The energy consumption of the highest consuming fan-out node.

opponent intruder would have a purposeful movement which can not be described by random waypoint.

The energy consumption of the bridge node is shown in Figure 4. As we have expected, the BPA algorithm significantly reduces the energy consumption. The magnitude of the improvement depends on the number of intruders. For a very small number of intruders, the improvement can be as high as 5 times, for a larger number (30 intruders), the ratio decreases to about 3 times. The reason for this decrease is that for a large number of intruders, the relative properties of the heartbeat messages in the traffic is lower.

Let us now consider the energy consumption of the highest fan-out node, in Figure 5. Looking at the graph without the BPA, the first observation is that the energy consumption of the highest consuming fan-out node is even higher than the bridge node. This is, obviously, the consequence of the fact that in a deterministic forwarding algorithm all the traffic flowing through the bridge node will be forwarded to a unique forwarding destination, to which we will also add the observations of the bridge node itself, and possibly, other nodes on the near side which have chosen the specific node as a forwarding next hop.

Thus, the fan-out node chosen for forwarding, although it is not in a topologically critical position, has a high chance to exhaust its energy even before the bridge, possibly creating a new bridge in the alternate node.

The application of the BPA algorithm radically reduces the energy consumption of the fan-out node: first, due to the reduction in the overall traffic on the bridge which is passed on to the node, and second, due to the fact that even this reduced traffic is split among the fan-out nodes.

V. Conclusion and Future Work

Exceptional situations require exceptional measures. The algorithm described in this paper is special in the sense that instead of proposing a general purpose approach to routing, it proposes the series of extraordinary measures which a network might take in response to the catastrophic events to postpone the most grave consequences (the fragmentation of the network). Thus the algorithm can find applications in networks affected by catastrophic events which are close to fragmentation. At the same time, the algorithm can be applied to protect relay nodes added to recover from fragmentation - if the minimum number of relay nodes is added, these nodes will be, by definition, bridge nodes.

References

[1] D. Turgut, B. Turgut, and L. Bölöni, "Stealthy dissemination in intruder tracking sensor networks," in *Proceedings of IEEE Local Computer Networks (LCN 2009)*, October 2009, pp. 22–29.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[4] H. Almasaeid and A. Kamal, "Data delivery in fragmented wireless sensor networks using mobile agents," in *Proceedings of the ACM MSWiM*, 2007, pp. 86–94.

[5] ——, "Modeling mobility-assisted data collection in wireless sensor networks," in *Proceedings of the IEEE GLOBECOM*, 2008, pp. 1–5.

[6] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the ACM MobiHoc*, 2004, pp. 187–198.

[7] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[8] X. Cheng, D. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[9] Y. Hou, Y. Shi, H. Sherali, and S. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2579–2590, 2005.

[10] S. Lee and M. Younis, "QoS-aware relay node placement in a segmented wireless sensor network," in *Proceedings of the IEEE ICC*, 2009, pp. 1–5.

[11] A. A. Abbasi, M. Younis, and K. Akkaya, "Movement-assisted connectivity restoration in wireless sensor and actor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, pp. 1366–1379, September 2009.

[12] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 258–271.

[13] M. Younis and K. Akkaya, "Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey," *Elsevier Ad Hoc Network Journal*, vol. 6, no. 4, pp. 621–655.

[14] L. Bölöni and D. Turgut, "YAES - a modular simulator for mobile networks," in *Proceedings of the ACM MSWiM*, 2005, pp. 169–173.

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