

Algorithm for Optimizing Energy Use and Path Resilience in Sensor Networks

Lawrence A. Bush, Christopher D. Carothers and Boleslaw K. Szymanski

Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, U.S.A.

{BushL2,chrisc,szymansk}@cs.rpi.edu

Abstract

Sensor networks will change the way computers interface with our world and with each other. This transformation will be shaped by the network centric paradigm demonstrated in sensor networks. Sensor networks also require a data-centric communication paradigm to efficiently and effectively share data. Directed Diffusion is a data-centric communication paradigm that forms a foundation of this paper. Energy efficient routing algorithms have been developed for Directed Diffusion; however, we have found improvements to them, which are described in this paper. We also present computer simulation results, which verify the effectiveness of previously established routing algorithms and compare them to our new and improved routing algorithms. The results show significant increase in energy efficiency and resilience. Finally, the paper incorporates effective techniques for modeling of sensor networks to demonstrate the usefulness of the new algorithms.

1 Introduction

A sensor network is a distributed sensing technology that can be used to monitor physical phenomena. A sensor network is easily deployed and is therefore useful for many applications. A sensor network is made up of many distributed sensor nodes, necessitating an Ad-hoc routing paradigm. Sensor nodes are battery operated. Many sensor network applications require thousands of sensor nodes which will be deployed in remote locations. This makes battery replacement impractical. Therefore, energy conservation is very important for sensor networks. Traditional Ad-hoc routing algorithms are not optimized for energy conservation. For this reason, energy efficient Ad-hoc routing paradigms are an area of active research.

Directed Diffusion (DD) is an example of an energy efficient routing paradigm which is central to this paper. Directed Diffusion algorithms consist of a flooding phase followed by a path reinforcement phase and a routing phase. The flooding phase of the DD paradigm is very costly. Therefore, avoiding this phase is desirable. Algorithms have been developed with the objective of extending the duration of the routing phase between floodings. This effectively avoids some flooding. The complicating factor is that sensor nodes are prone to failure. Therefore, in order to extend the routing phase, the employed paths must be resilient to node fail-

ure. The means to accomplish this is by setting up multiple routing paths between the source node (which senses the physical phenomena) and the sink node. These paths can be set up in many different ways. However, setting up and maintaining extra paths requires extra energy. Therefore, it is important for this act to be energy efficient. An example of such an algorithm is presented in [1]. While adaptations of DD have been studied, more studies are needed. An algorithm which reduces the energy used to maintain multiple paths or improves the resilience to path failure would extend the life of the network.

In our paper, we present a better alternative to the Braided Multi-path algorithm presented in [1]. Our new algorithm simultaneously reduces the maintenance overhead associated with these multiple paths and increases their resilience to node failure. Computer simulation results which verify the claims made in [1] are presented here. These results are used as a baseline for the comparison of our new algorithm. The new algorithm was shown to improve resilience to path failure by over a third while using as little as 25% of the energy for a given epoch. These significant improvements in energy efficiency and resilience encourage us to further develop the sensor network simulation capabilities of Rensselaer's Optimistic Simulation System (ROSS).

2 Related Work

2.1 Directed Diffusion

Current sensor network research is focused on algorithms and protocols for managing energy consumption using protocols such as DD. DD addresses scalability, energy efficiency and robustness to topology changes. Unlike Internet protocol, it accomplishes this using named data rather than global node identifiers. DD uses only local routing information, avoiding much of the routing table updates required for path based connectivity. Specifically, DD only stores the next hop routing information rather than using the connectivity abstraction. This also reduces the amount of routing information stored at each node and improves routing efficiency and resiliency by allowing topology changes to be handled locally. These attributes enable data delivery without total network connectivity which would inhibit scalability. The on-demand path construction also enables robustness while saving energy. While adaptations of DD have been studied, more studies are needed. DD is a sound

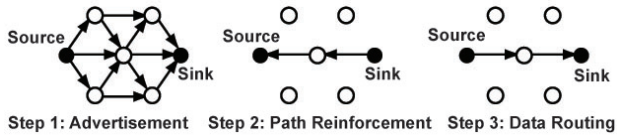


Figure 1: Source Initiated DD [1].

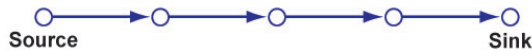


Figure 2: Single-path routing. [1].

paradigm, which continues to evoke new generic and application specific ideas.

There are essentially two types of DD: Sink Initiated and Source Initiated. This paper deals with Source Initiated DD as described in [1] and [2]. In Source Initiated DD, the source will advertise the availability of its data. This is essentially a flooding mechanism, during which the network sets up routing gradients back to the source. This is followed by a path reinforcement phase which identifies the best path(s) and initiates the sending of data packets. A diagram of this process is shown in Figure 1. In [1], different path construction techniques are tested to increase the time period between the source-initiated flooding phases. Full network flooding is then only needed when all of the paths connecting the source-sink pair fail. Specifically, the paper [1] presents a Braided Multi-path reinforcement strategy for DD routing in sensor networks.

2.2 Braided Multi-path Routing

Our new algorithm is a better alternative to the Braided Multi-path routing algorithm presented in [1], in terms of resilience and energy use. Our algorithm efficiently finds and maintains routes within the Source Initiated DD paradigm. In doing so, it accomplishes the objective of extending the life of the network over what has been achieved by the Braided Multi-path routing algorithm. Our algorithm has the added benefit of improving continuity of path connections. To that end, the baseline Braided Multi-path routing algorithm is explained here.

Braided Multi-path, presented in [1], conserves energy by prudently avoiding the costly flooding phase of DD by extending the routing phase. The routing phase ends when a node on the routing path fails and, consequently, the source can no longer send data to the sink. To recover from this situation, the flooding phase must be initiated, assuming that the routing algorithm reinforces only a Single Path as shown in Figure 2. However, a multi-path routing strategy such as Disjoint Multi-path routing (Figure 3) can be employed to increase resilience to node failure. The Disjoint Multi-path algorithm works like the Single Path algorithm except that an alternate path is reinforced in addition to the primary path. This path must enforce the disjoint property which means that it does not intersect the primary path. The paper [1] shows that this algorithm improved path resilience, but at a significant cost.

It is important to note that the maintenance of alternate

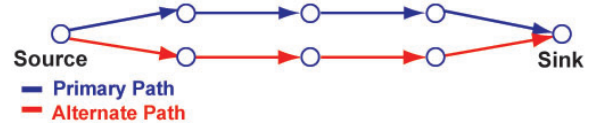


Figure 3: Disjoint Multi-path routing [1].

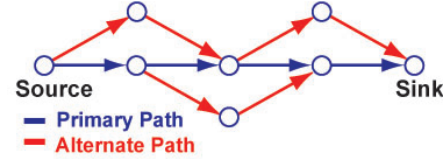


Figure 4: Braided Multi-path routing [1].

paths requires some additional energy. This is primarily in the form of keep-alive packets. Essentially, the keep-alive packets are periodically sent from the source down the alternate paths to keep them from going into sleep mode. This enables fast recovery from failure on the primary path.

In the paper [1], Braided Multi-path (Figure 4) is presented as an alternative to Disjoint Multi-path. In the Braided Multi-path algorithm, the sink reinforces the primary path as was done in the Single Path routing algorithm. Additionally, at each node on the primary path, an alternate reinforcement packet is also sent down the penultimate path of that node. That reinforcement marks that node as the alternate path. The reinforcement continues from there following the shortest path to the source, reinforcing all the way. If the reinforcement happens upon the primary path, it rejoins it and terminates. This process continues until the reinforcement packet reaches the source. The sum of these creates a braid-like path set consisting of a primary path and a series of alternate paths which each serve to circumnavigate a particular node of the primary path. Compared to Disjoint Multi-path, Braided Multi-path increases the resilience of the path, but at a lower path maintenance cost. This algorithm is the primary result presented in [1].

3 Solution Description

This section describes our improved algorithm. We began developing our algorithm by addressing keep-alive packets. These keep-alive packets, which are needed to maintain (keep open) the paths that have been discovered through the flooding mechanism, can be disseminated in a different way which we call “Dealer Routing.” This saves a great deal of energy. After developing an effective method to mitigate this energy use, we proceeded to address the issue of path resilience to node failures. We found a method to address this, then improved on it using a more effective path reinforcement termination strategy. This strategy made path reinforcement more efficient and well behaved. The result is a significant improvement in resilience to node failures, that leads to a better quality of service, and an order of magnitude improvement in energy use.

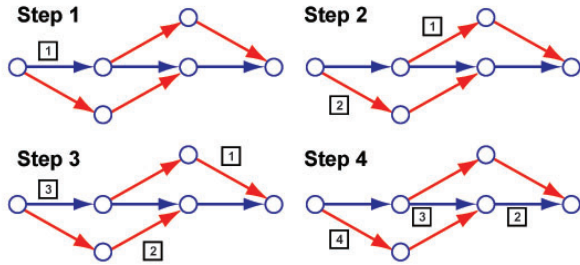


Figure 5: This diagram depicts four steps of Braided Dealer Routing. The squares represent data packets.

3.1 Dealer Routing

Dealer Routing is an alternative to using keep-alive packets. Keep-alive packets are used to keep the alternate paths in a ready state, rather than in the low energy use sleep state. Therefore, when a failure is detected in the primary path, an alternate path can be reinforced quickly, instead of using flooding based path discovery.

Dealer Routing, however, takes an alternate approach which eliminates the need for keep-alive packets. The routing sequence used in this method is analogous to dealing cards. Dealer Routing alternates which path it sends a packet down. Since the paths are braided, each node may have multiple paths. If there are two or more reinforced paths from the given node, it will send the first data packet down one of them, the second data packet down another and so on, in an established order, until each reinforced path has received a data packet as shown in Figure 5. Each of these nodes behaves by the same rules. By doing so, each path is kept alive by the data packets because each path is periodically used. The data packets perform double duty as keep-alive packets. As a data packet passes through a node, the routing information contained at the node is updated. This eliminates the need for sending the low-rate keep-alive packets.

The amount of time that a node will stay awake without receiving a packet is dictated by a timer. All of the simulations discussed here use the same time period. The time-to-sleep period should be as short as possible so that unused nodes will turn off. However, it should be long enough so that the nodes in the multi-path receive a packet before they go to sleep.

Dealer Routing assumes that the requested data rate is high enough to keep all of the nodes in the path set from going to sleep. This assumption is increasingly important for the routing algorithms presented in sections 3.2 and 3.3. It enables the data packets to update the node routing information at least as frequently as the low-rate data packets would have. The low-rate data frequency is matched to the internal timing mechanism of the sensor nodes. However, in the event that the magnitude of the data packet frequency is insufficient to entirely satisfy this requirement, the algorithms presented in 3.2 and 3.3 will selectively remove any stale paths from the routing path set via a path reinforcement mechanism. This ensures that the routing path set is maintainable. In our simulations, we chose a data rate that

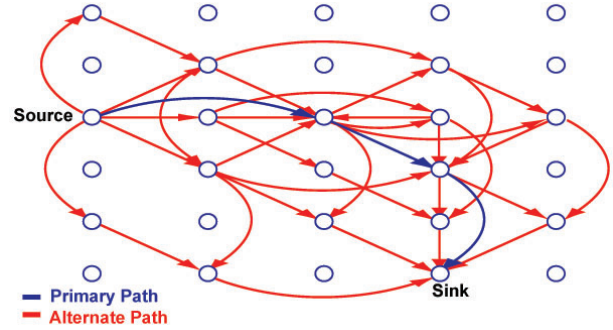


Figure 6: This diagram shows an example of reinforced paths in Log N-Braided Dealer Routing.

will cause this to happen when path lengths become long and consequently the path set becomes too bushy. This situation was chosen in order to test the algorithms under a more difficult scenario. This led to the development of the Log N-BDR which terminates the path discovery in a way that creates a fairly constant path set bushyness from source to sink.

In order to test the effectiveness of the Dealer Routing strategy, we integrated it into Braided Multi-path routing, resulting in Braided Dealer Routing (BDR). The flooding and path reinforcement phases of BDR operate the same as they did in the Braided Multi-path algorithm presented in [1]. However, the routing phase uses the Dealer Routing strategy instead of primary path routing to avoid the use of keep-alive packets. The objective is to extend the lifetime of the network by reducing the energy use in the routing phase while maintaining a similar level of resilience to node failures.

3.2 N-Braided Dealer Routing

N-BDR improves resilience (over BDR) by allowing a greater than binary alternate path splitting during the path construction phase. In this algorithm, any number of N-ary path splitting rates can be used. As in the Braided Dealer Routing algorithm, an alternate reinforcement packet is sent down the penultimate path of each primary path node and marks that node as the alternate path. However, it does not stop there, rather an alternate reinforcement packet is sent down N-1 alternate (next best) paths of each primary path node, marking them as alternate paths. Each of these reinforcement packets subsequently reinforces the shortest path to the source. The sum of these creates a braid-like path set consisting of a primary path and a series of alternate paths. However, in this case, it has a wider swath. The paths in the swath intermingle in interesting ways which increases resilience.

3.3 Log N-Braided Dealer Routing

This algorithm further improves resilience (over N-BDR) with minimal increase in energy consumption by increasing

the interconnectedness between pathes and consequently the number of ways for the packets to be routed around failed nodes.

This algorithm is similar to N-BDR in that it will send an alternate reinforcement packet down the $N - 1$ next best paths from each alternate node. In this case, however, N is not the original N , but is equal to $(N - 1)/2$. In other words, if $N = 5$, the primary path will reinforce $N - 1$ (4) alternate paths. Each of these alternate paths will reinforce $(N - 1)/2$ (2) paths. Subsequently, each of these alternate paths will reinforce $(N - 1)/2/2$ (1) paths. Each alternate path emanating from the primary path splits for $\log_2(N - 1)$ levels. Each of the new paths, resulting from the split, reinforce the shortest path to the source, from that point.

This termination strategy is an improvement in resilience and energy use over N-Braided Dealer Routing because it creates a more consistent path set width between the source and the sink. This reduces the data rate required to keep the path nodes alive. This strategy also creates a braid-like path set consisting of a primary path and a wider, very intertwined series of alternate paths as shown in Figure 6. The objective is to further increase resilience by creating more ways to route around failures. This method uses additional nodes and therefore more energy. However, the termination strategy behaves in a constrained way. The reinforcements generally converge towards the primary path, reducing the number of extra nodes that are used while greatly increasing the number of interconnections.

4 Solution Analysis

4.1 Model Validation

The purpose of the initial simulation runs was to verify the results of the paper [1] and establish a baseline. These simulations compared Single, Multi-path and Braided routing algorithms. The tests simulate all three algorithms for a prescribed duration. The simulated time period was sufficiently long to measure the energy used during the flooding, reinforcement and data routing phases. The relative length of these phases will affect the results because the flooding phase uses a large amount of energy yet the primary energy savings takes place during the data routing phase. Therefore, if the data routing phase is extended, then the energy savings is greater. With that said, the simulation results support the results from the paper [1] because the Dis-joint Multi-path and Braided Multi-path algorithms result in greater resilience than the Single Path routing algorithm. Also, the Braided Multi-path uses less energy than the Dis-joint Multi-path method.

4.2 Performance Results

Our simulation results were obtained using ROSS: Rensse-laers Optimistic Simulation System described in [3]. The network is modeled as a 32 by 32 regular mesh. This idealized topology was chosen in order to isolate the behavior of the algorithms from the impact of topology variations. Our model assumes a fixed transmission radius as described

in [1]. The simulations model isolated failures in the manner described in [1]. An isolated failure is the failure of an individual node (i.e. due to malfunction or battery drain). We also model pattern failures as described in [1]. A pattern failure is the failure of a group of nodes. The simulation results discussed in this paper were run in 56 different configurations (i.e. pattern failures, path length, N-Braidedness). Each configuration was run for 30 trials. The results from the simulations of the new algorithms are very encouraging.

4.3 Braided Dealer Routing

BDR showed some positive and interesting performance characteristics. For example, it demonstrated significant universally lower energy use than the baseline algorithm. We expected the resilience of the baseline and new algorithms to be equal. The simulation results for a path length of 6 concur with our assumption. However, for a path length of 20, the simulation results measured the resilience of the new BDR algorithm, to be lower than the resilience of the baseline Braided Multipath routing algorithm. The results are due to the modest data rate and short time-to-sleep used, which tests our algorithms under unfavorable conditions.

To address this problem, we increase the arity of the multi-path while using a termination strategy that has a pruning effect. In this way, we sought to improve the algorithm independent of the snooze parameter and data rate. The explanation follows in the next two subsections.

4.4 N-Braided Dealer Routing

The N-BDR algorithm spends some of the energy saved by the Dealer routing strategy, in an attempt to buy additional resilience. While the N-BDR algorithm was not the best performing routing algorithm that we found, it does demonstrate that increasing the arity of braided routing is a viable variation of Dealer Routing. The simulations showed that the energy use of N-BDR was still lower than the baseline Braided Multi-path despite the higher arity. However, the effect on resilience was mixed.

Specifically, the simulation results measured the energy use of N-BDR to be 11% and 75% lower than the baseline algorithm for path lengths of 6 and 20 respectively. The results also showed the resilience of the N-BDR to be better than the resilience of the baseline algorithm for a path length of 6, but worse for a path length of 20. The probable cause of the poor results for the longer path length is that some of the nodes in the routing path set went to sleep because they did not receive a data packet often enough. This is a consequence of a bushy path set. We could have tuned the time-to-sleep to prevent this, however, we instead devised a more general solution where the required data rate is independent of the path length. This method is described in the next subsection.

4.5 Log N-Braided Dealer Routing

The Log N-BDR algorithm is the best performing routing algorithm in our simulations. It demonstrated the high-

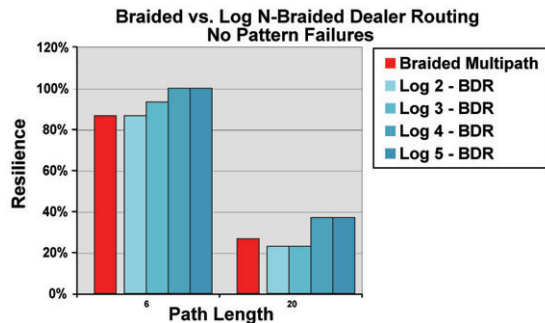


Figure 7: The resilience of the Log N-Braided Dealer Routing is better than the resilience of the baseline Braided Multipath routing algorithm.

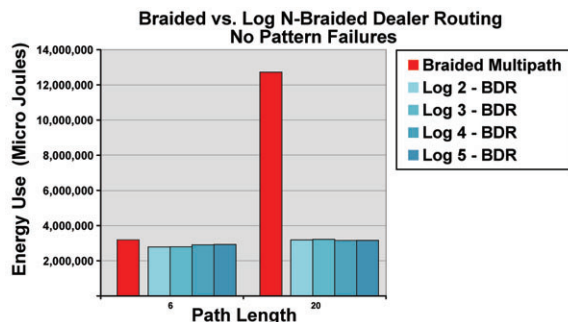


Figure 8: Energy use of Log N-Braided Dealer Routing is approximately 75% lower than the baseline Braided Multipath for a path length of 20.

est resilience (Figure 7) and lowest energy use (Figure 8) in most cases. The few algorithm configurations showing equivalent resilience used far more energy making them less desirable. Likewise, the Log N-BDR algorithm used significantly less energy than the other algorithms, with the exception of the less resilient Single Path algorithm. Specifically, the Log N-BDR, using an arity of 4 or 5, was more resilient and used less energy than the baseline algorithm presented in [1]. Most importantly, the energy use of the Log N-BDR algorithm is more scalable than the baseline algorithm. This is clear from the results using a path length of 20 with no pattern failures. The Log-BDR algorithm was significantly more resilient while using only 30% as much energy as the baseline algorithm. Overall, the Log N-BDR algorithm demonstrated the best cost-benefit tradeoff between energy use and resilience to node failure.

4.6 Pattern Failure Impacts

One interesting effect of pattern failures is that they level the playing field. When pattern failures are simulated, the resilience improvement is less pronounced. Pattern failures are very difficult to recover from because they take out a large group of nodes which is likely to sever all of the maintained paths. Our conclusion is that it is better to let the flooding mechanism route around these large groups of failed nodes because it is too expensive to maintain a path

set that is resilient to them. Isolated failures, on the other hand, can be dealt with effectively.

4.7 Reinforcing Disjoint Paths

While conducting simulation runs and adjusting various algorithm parameters we observed that it often took a long time for the network to construct alternate disjoint paths. This is because the alternate path often converges back to the primary path causing a “negative reinforcement” which enforces the disjoint property in the algorithm. The consequence of this behavior is that it will take much longer to construct the alternate path than it takes to construct the primary path. This affects recovery latency and causes keep-alive complications. Therefore, braided paths are more desirable for this reason alone. Since braided paths do not enforce the disjoint property, they take less time to construct.

5 Conclusion

In this paper we have reported on our studies of DD routing algorithms. Our studies include constructing computer simulations, validating our simulation model by confirming previous research results and measuring the behavior of our new algorithms given certain network assumptions. While our ultimate objective was to extend network life, the supporting objectives are to conserve energy, extend the duration of the routing phase and improve resilience to node failures.

We have constructed new algorithms to help achieve these goals. The results are very encouraging and interesting. Some of our findings are as follows: Dealer Routing is useful because it avoids using keep-alive packets which saves a significant amount of energy. Multi-path routing is well suited to handle isolated failures but does not solve the pattern failure problem. Flooding remains the better solution for handling pattern failures. Increasing the number of braided paths combined with a strategy to increase the interconnectedness of these paths improves resilience to node failure with a manageable increase in overhead.

References

- [1] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. “Highly-Resilient, Energy-Efficient Multi-path Routing in Wireless Sensor Networks.” *Mobile Computing and Communications Review*, Vol. 4, No. 5, October 2001.
- [2] C. Intanagonwiwat, R. Govindan, and D. Estrin. “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks.” *In Proceedings of 6th ACM/IEEE Mobicom Conference*, 2000.
- [3] C. D. Carothers, D. Bauer and S. Pearce. “ROSS: A High-Performance, Low Memory, Modular Time Warp System.” *In Proceedings of the 14th Workshop of Parallel on Distributed Simulation (PADS 2000)*, pages 5360, May 2000.
- [4] R. Shah and J. Rabaey. “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks.” *In Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.