On Selecting a Reliable Topology in Wireless Sensor Networks

Israat Tanzeena Haque
Computing Science
University of Alberta, Canada
israat@cs.ualberta.ca

Mohammad Saiful Islam
National Oilwell Varco
Alberta, Canada
msislam3@ualberta.ca

Janelle Harms
Computing Science
University of Alberta, Canada
janelleh@ualberta.ca

***Abstract*—Wireless Sensor Networks (WSNs) have a wide range of applications including smart home and environmental monitoring systems. Sensors in such systems gather and propagate critical information to sinks (data collectors). Reliability is a crucial issue in these networks but the wireless medium is subject to interference and noise that can disrupt transmissions. Thus we design a reliable routing topology construction framework for WSNs and evaluate the proposed reliable topologies' performance. We observe that there are tradeoffs of reliability, energy consumption and path properties that applications can take advantage of.**

I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of a set of resource-constrained (limited processing and battery power) sensor devices that are connected through a wireless medium. Usually a special node called the *sink* coordinates the sensors that collect and deliver data to the sink. Neighbors of a sensor node fall within its radio range. Since each sensor has a limited radio range, a multi-hop network topology is formed. Applications of WSNs include environmental and industrial monitoring, localization and tracking, smart home monitoring etc. All these applications deal with critical data that must reach the sink to take appropriate action. However, wireless networks are subject to interference/noise, thus reliability (completing a specific task without failure) is a vital issue here. One way to improve the reliability is to have alternative routes to deliver the information. This motivates us to address the problem of constructing reliable routing topologies for critical information delivery and evaluating their tradeoffs.

There is work that generates topologies to optimize node degree and path length [1], [2], end-to-end delay and packet loss or energy consumption [3], [4]. However, they do not provide alternative paths to deliver data in the presence of link failures. A $(\frac{2\pi}{\theta})-$edge connected topology, with $\theta \geq \pi$, is proposed as the physical topology for WSNs in [5], where sensors are deployed in prerequisite locations in order to hold the above property. In [6], a $k-$edge connected topology, called LTRT, based on minimum spanning trees provides low node degree and energy consumption with low computation cost. In both of these constructions, the routing performance is not evaluated. A survey of multipath routing for WSNs is given in [7].

We first show that a $k-$edge connected routing topology construction approach can be applied to any spanning structure and define a general algorithm for this purpose. In particular, we consider building k-edge connected routing topologies from the basic spanning topologies of Minimum Spanning Tree (MST) [8], Shortest Path Trees (SPT), Degree-Constrained Shortest Path Trees (DCSPT) [9] and Gabriel Graphs (GS) [10]. Forwarding algorithms and a scheduling algorithm are defined to ensure conflict-free transmission for converge cast traffic. Simulation results explore the tradeoffs among the different topologies with respect to path properties, energy use and reliability. Finally, queue based packet forwarding schemes are proposed and evaluated on the distributed Gabriel based topology.

We organize the remaining work as follows. We define the routing topology construction problem in Section II and propose the generalized routing topology construction algorithm in Section III. The next section describes the scheduling and forwarding. The simulation environment is described in Section V. The next section presents the performance analysis in a static environment. Section VII contains the behavior of the topologies in more dynamic network scenarios. In Section VIII, we evaluate new forwarding algorithms for the distributed Gabriel-based routing topology followed by the concluding remarks.

II. THE PROBLEM DEFINITION

A sensor network can be described as a set V consisting of N nodes placed in a 2-dimensional Euclidean space. We assume that the transmission range of all nodes is R. Two nodes can communicate with each other if and only if their Euclidean distance is at most R. The ability to communicate is represented by an edge, in E, between the corresponding nodes. The resulting graph, *G(V,E)*, is the *physical topology* of the network. G varies over time due to the presence or absence of the links among nodes. The topology construction problem is to define a subgraph of a given physical topology, where packet routing will be performed. Thus we call such a subgraph of the physical topology, the *routing topology*.

The spanning topologies that we use as a basis of the construction of routing topologies are the Minimum Spanning Tree, the Shortest Path Tree and the Gabriel Graph. A *spanning tree* is a subgraph of G that contains

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

all the vertices of G and forms a tree. Each edge of G has a positive weight corresponding to the Euclidean distance between endpoints. A spanning tree with sum of edge weights less than or equal to any other spanning tree of same graph G is called a *Minimum Spanning Tree (MST)* [8]. MSTs have shorter links and therefore can save energy but have tradeoffs in that paths have more hops. In [8] it has been shown that the degree of MST is at most 6 in G.

A *Shortest Path Tree* is a subgraph of G which minimizes the distance between any node and the sink. A *Degree-Constrained Shortest Path Tree (DCSPT)* is an SPT that constrains the degree of each node to at most d, a constant. DCSPTs have shortest paths that improve delay and also control the node degree which can be beneficial for scheduling.

For the *Gabriel Graph (GG)*, assume that $disk(u,v)$, is denoted by a circle containing u and v with the diameter equal to the Euclidean distance between them. Then edge (u,v) of G belongs to GG if and only if no other nodes $w \in V$ are located within or on the periphery of $disk(u,v)$ [10]. Gabriel graphs have relatively short links and therefore good energy-use properties.

However, these basic graph structures do not ensure alternative paths exist and therefore are not suitable for reliable routing topologies. A graph is said to be *k-edge connected* if it does not contain $k-1$ edges whose removal results in a disconnected graph. For any pair of nodes in G, there are k unique paths between them. Given a $k-$edge connected physical graph G, the problem of constructing a $k-$edge connected routing topology is to find a connected subgraph $D_k(G)$ of G such that there are k edge-disjoint alternative paths between *every pair* of nodes in $D_k(G)$. We consider the problem of constructing a k-edge connected topology, for $k=2$, i.e., a 2-edge connected routing topology out of the given physical topology.

III. Algorithm to Construct Reliable Topology

We present a generalized algorithm in Algorithm 1 to construct a $k-$edge connected topology for different spanning structures following the MST-based topology (LTRT) of [6]. Let us give an intuitive idea of constructing a $k-$edge connected subgraph of a given G, for $k=2$. A 1-edge connected topology, say $D_1(G)$, is first constructed as a subgraph of G. Then all the edges that belong to $D_1(G)$ are removed from G. This may make G disconnected and composed of two or more connected components. In the second phase of the topology construction, we consider these connected components and for each one build a corresponding 1-edge connected topology. These newly found edges are then appended to $D_1(G)$ to form the 2-edge connected topology $D_2(G)$.

An *edge cut* of a graph is a subset of its edges whose removal results in a graph with more connected components than the original graph. An edge cut *separates vertex set A from vertex set B* if the only edges in the graph connecting A and B are edges of the cut. A *component-connected spanning subgraph* of graph $G=(V,E)$ consists of one connected spanning subgraph for each connected component of G.

The $k-$connectivity property of $D_k(G)$ is proved below in Theorem 1, which is a straightforward generalization of the theorem and proof presented in [6] for the MST-base LTRT spanning subgraph to any component-connected spanning subgraph.

Theorem 1: Let $G=(V,E)$ be a graph. Let E_1 be the edges of a component-connected spanning subgraph of G and, for all $i \geq 2$, let E_i be the edges of a component-connected spanning subgraph of $(V, E - \cup_{j=1}^{i-1} E_j)$. For all positive integers i, let $D_i(G)$ be the graph $(V, \cup_{j=1}^{i} E_j)$. Then for $k \geq 1$, if G is k-edge connected then $D_k(G)$ is k-edge connected.

Proof: We prove by induction that $D_\ell(G)$ is ℓ-edge connected for all $1 \leq \ell \leq k$. Note that $D_1(G)$ is a connected spanning subgraph of G and is therefore 1-edge connected. Now consider ℓ in the range $2 \leq \ell \leq k$ and assume that $D_{\ell-1}(G)$ is $(\ell-1)$-edge connected. Note that $D_{\ell-1}(G) = (V, \cup_{j=1}^{\ell-1} E_j)$ and $D_\ell(G) = (V, \cup_{j=1}^{\ell} E_j)$, that is, $D_\ell(G)$ is simply $D_{\ell-1}(G)$ plus the edges E_ℓ.

Suppose that $D_\ell(G)$ is not ℓ-edge connected and let $\{e_1, \ldots, e_{\ell-1}\}$ be an edge cut of $D_\ell(G)$ that separates vertex set A from vertex set B where $V = A \cup B$. Each edge of $\{e_1, \ldots, e_{\ell-1}\}$ must be an edge of $D_{\ell-1}(G)$ since $D_{\ell-1}(G)$ is $(\ell-1)$-edge connected. Since G is ℓ-edge connected, there is at least one edge of $E - \cup_{j=1}^{\ell-1} E_j$ that connects A and B. But now there are two vertices, one in A and one in B, that are connected in $(V, E - \cup_{j=1}^{\ell-1} E_j)$ but not in (V, E_ℓ), contradicting that (V, E_ℓ) is a component-connected spanning subgraph of $(V, E - \cup_{j=1}^{\ell-1} E_j)$. Therefore we conclude that $D_\ell(G)$ is ℓ-edge connected and the theorem follows. ■

Algorithm 1 $k-$edge connected topology $D_k(G)$

```
1: Extract E_1, the edges of a component-connected
   spanning subgraph of G.
2: D_1(G) ← E_1.
3: for i ← 1 to k − 1 do
4:    Remove edges of D_i(G) and get the connected
      components D_i1(G), D_i2(G), ...., D_il(G)
5:    for j ← 1 to l do
6:       Generate component-connected spanning sub-
         graph D_ij(G).
7:    end for
8:    Merge D_i(G), D_i1(G), D_i2(G), ..., D_il(G) to obtain
      D_k−1(G)
9: end for
```

Following Algorithm 1 we define the routing topologies as follows: MST2 is created by first constructing the subgraph $D_1(G)$ using MST, removing its edges and using MST on the subgraphs again. SPT2, DCSPT2 and Gabriel2 are constructed in the same way. In addition, Gabriel is combined with the other topologies as follows: DCSPT-Gabriel (DCGB) topology is a subgraph of G which first constructs the subgraph following definition of DCSPT, then using the Gabriel graph construction on the remaining components. MST-Gabriel (MSGB) combines the MST construction (first) followed by the Gabriel construction.

Complexity analysis: There are N nodes. Finding the DCSPT, SPT, and MST is $O(N^2)$ if no special data structures are used. Whereas, for Gabriel, if a node's degree is d, then the algorithm requires d^2 operations to find its edges. Therefore, in the worst case each node takes $O(N^2)$ to obtain the edges of the routing topology. However, each node can find the edges locally, in parallel, if implemented in a distributed way which makes the algorithm take $O(N^2)$ time. In the subsequent steps we repeat the topology construction on the edges that are not chosen in step 1. However, the number of nodes remain unchanged. Thus the complexity depends on k, i.e., $O(kN^2)$.

IV. Scheduling and Forwarding

A collision-free TDMA [11] scheduling algorithm is used to arbitrate the shared bandwidth by assigning links to slots to control the packet delivery. In particular, the sink computes a *conflict graph* [11] that reflects which groups of links mutually interfere and hence cannot be active simultaneously. This graph is then colored using a heuristic (highest degree node is colored first) to ensure that interfering links of the original routing topology will not be assigned to same slot. Note that for comparison purposes we consider the same centralized scheduling for all the topologies.

A sensor node detects a down link when its transmission is not acknowledged. It is assumed that the slot size is long enough to include an ACK from the receiver. Therefore, the failed links are locally detected and the forwarding of packets is locally decided. Because we have a 2-edge connected topology, there are at least two different parents (links) that can be taken to route to the sink. In the presence of a down link, we alternate between these two links: one the best parent and the other the possibly suboptimal parent. A sensor tries the optimal parent as per its scheduling slot. If the attempt fails to reach that optimal parent, the sensor tries the second parent in the following scheduling period. The process goes on until either the packet is successfully delivered to a parent or the retry limit is reached (we set it to 7 in our simulation). In the latter case the packet is dropped.

V. Performance Evaluation

In this section we define the simulation environment and the necessary models to evaluate the performance of the proposed reliable routing topologies. In particular, we are interested in the quality of the paths and the ability to provide alternative paths in the presence of broken links. N nodes are generated at random and uniformly spread over a 100m$\times$100m area with $N \in 50, 75, 100, 125, 150$. The transmission range of each node is 25m.

We assume that the sink has global topology information. It builds the $k-$edge connected topology and finds the conflict-free schedule and forwards this information to the sensor nodes at set-up time. First, the structure of the routing topologies is evaluated (static evaluation) based on the average node degree, the average path length (in hops) from nodes to the sink, and the average communication energy consumption. The path length and energy use is then re-evaluated in the presence of link failure. Second, the topologies are evaluated (dynamic evaluation) with traffic routed to the sink to measure packet delivery ratio, average hops, and energy use. All the performance metrics are averaged over 100 runs and results are given with 95% confidence intervals.

VI. Performance Analysis in Static Environment

The structure of the topologies is evaluated in this section both with no down links and in the presence of down links.

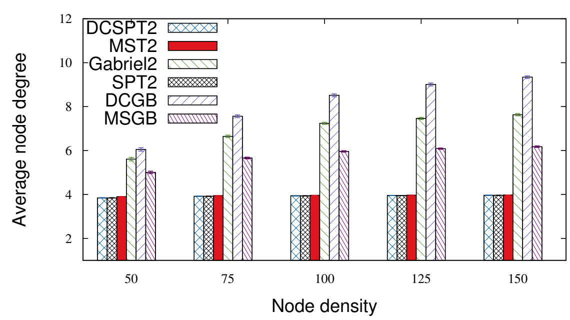

Fig. 1. The average node degree of a node.

In this section, the sink is chosen based on the DC-SPT structure which has the most conservative topology with respect to degree and path length restrictions (for dynamic evaluation this restriction is relaxed).

The average node degree (including the sink) of different topologies is shown in Figure 1 for different density of networks. The guaranteed maximum degree that exists with MST and DCSPT subgraphs no longer hold for the 2-edge connected topologies of MST2 and DCSPT2, however, the average node degrees of DCSPT2, SPT2, and MST2 are the smallest and do not vary much with the node density. This limited node degree can be useful for improving scheduling effectiveness and energy use. The node degree of the three Gabriel-based topologies: Gabriel2, MST-Gabriel, and DCSPT-Gabriel are higher and increase slightly with the node density. Gabriel graphs are a superset of MST graph and thus have more edges. In addition, the construction of Gabriel graphs do not consider reducing node degree.

The average number of hops from the sensors to the sink is given in Figure 2. As expected, DCSPT2,

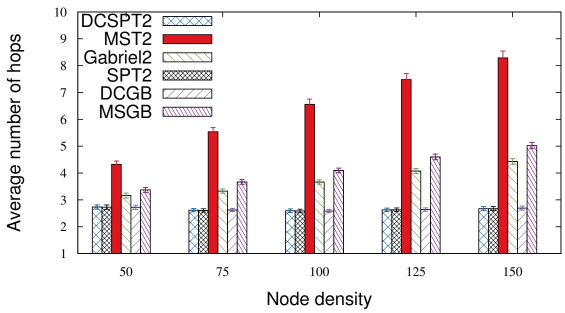

Fig. 2. The average number of hops from sensors to the sink.

SPT2, and DCSPT-Gabriel topologies give the best performance in terms of minimizing the number of hops from the sensors to the sink. These SPT-based topologies maintain almost a constant path length despite the density changes because the diameter of the network does not change significantly with the node density. The longest number of hops is taken by MST2 topology because MST2 chooses the shortest weight edges leading to longer (hop) routes. As node density increases, more and more small-weight edges are considered and the corresponding path length of MST2 increases. The Gabriel2 graph contains more edges than MST2, so a smaller number of hops is observed for the Gabriel2 topologies.

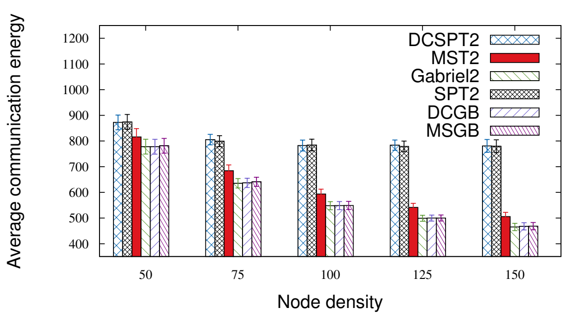

Fig. 3. The average communication energy from sensors to the sink.

The communication energy use of different topologies are shown in Figure 3. This is purely based on the structure of the topologies when no links fail. To send a message, a node typically consumes energy equal to the squared distance between itself and the receiving neighbor and therefore the squared path length is used as the energy metric. The three Gabriel-based topologies give the best energy use since Gabriel avoids longer links. The MST2 topologies also have short links but they have longer hop paths than the Gabriel-based topologies, thus have more energy usage. The DCSPT2 and SPT2 topologies have the worst use of energy.

Observation: Certain tradeoffs can be identified here. If we want to deploy a network with a limited node degree, the choice would be MST2, SPT2, and DCPST2. If the next goal is to optimize the path length, then MST2 needs to be removed from the above list. Thus the shortest path based topologies are the ones that ensure low node degree and path length at the price of highest communication energy consumption. On the other hand, the Gabriel-based topology improves the energy use and has moderate path length and node degree.

A. *Reliability measures*

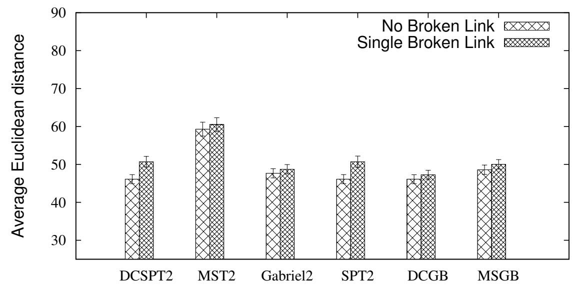

Fig. 4. The performance measures with broken links.

To measure the reliability in the static environment, we first measure the average alternative path length shown in Figure 4. The DCSPT2 and SPT2 topologies are affected more (relatively) when links from the optimal routes fail. The Gabriel-based topologies show very little impact with broken links indicating that they have several (nearly optimal) alternative routes to the sink. Thus there is a trade-off between the node degree (which reflects routing topology density) and the reliability of the routing topology.

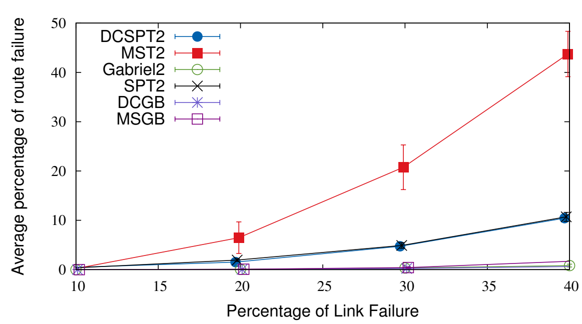

Fig. 5. The reliability measures.

To measure the *degree of reliability* (the resilience of a routing topology to broken links) in the static environment we randomly choose a subset of links to remove from each topology and measure the corresponding path length. This is shown in Figure 5 with different percentages of randomly chosen failed/down links. MST2 has the worst performance due to its longer hop paths. The Gabriel-based topologies have the best performance due to more available alternative paths.

Observation: In terms of the reliability, the structures of the Gabriel-based routing topologies are more robust. These topologies also have the best communication energy consumption with moderate path length and node degree, which make Gabriel based topology a good choice for a resource-constrained environment where reliability is an issue.

VII. Performance Analysis in Dynamic Environment

In this section, the performance of the routing topologies is investigated with routed traffic. The results of the previous section show that the Gabriel-based topologies demonstrate better robustness based on idealized conditions. Routing and scheduling play a role in the actual performance of these topologies.

There are 100 nodes, 1 of which is the sink and 20 are sources that generate traffic at the beginning of each scheduling period. The simulation time consists of 10000 scheduling periods. Since our topology is 2-edge connected, any sensors from the network can be chosen as the sink. Thus, in this dynamic environment, we choose the sensor with minimum scheduling length as the sink to offer each topology their best routing configuration. We run the simulation for the same time period and generate the same number of packets for each topology to make the comparison consistent.

A. Effect of packet generation probability

In the first set of experiments, packets are generated at the sources with probabilities 0.1, 0.2, 0.3, 0.4, and 0.5 and the link failure probability is set to 0.01.

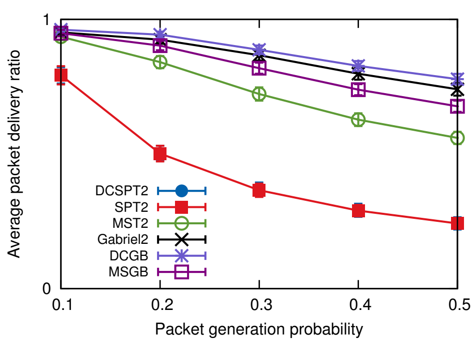

Fig. 6. The average packet delivery ratio with different packet generation probability.

The packet delivery (see Figure 6) of DCSPT2 and SPT2 is lower than the other routing topologies. This can be explained by looking at the average path length in hops in Figure 7. Here these algorithms, despite finding shortest paths have higher average path lengths. This occurs because the sink that provides the best schedule is towards the edge of the graph because that provides the most opportunity for parallel transmission and the least conflict. However, longer paths are more likely to break with link failure and packets are less likely to be delivered during the simulation time period. In contrast, Gabriel and MST have more diversity in paths and their sink placement results in shorter paths. The Gabriel-based topologies have more centrally-placed sinks and the shortest path lengths are therefore able to deliver packets more reliably.

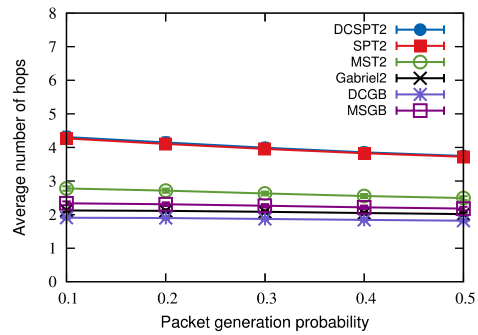

Fig. 7. The average number of hops with different packet generation probability.

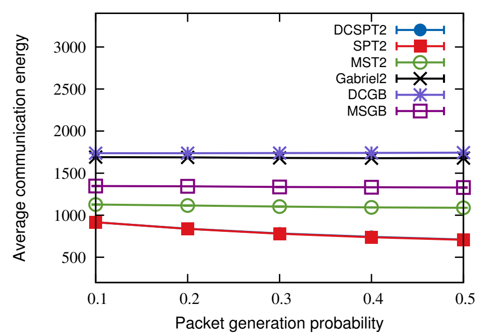

Fig. 8. The average communication energy with different packet generation probability.

To measure the communication energy we consider the transmission power for both successful transmissions and retransmissions in case of unsuccessful attempts. Thus if a packet reaches the sink with a small number of successful transmissions and minimal retransmissions, it will consume less communication energy. We may then expect the energy consumption trend of the topologies (shown in Figure 8) to be similar to their delivery ratio. The downward trend of the graphs in Figure 7 and Figure 8 is explained by the lower packet delivery ratio at the higher load particularly for longer paths.

B. Effect of link failure probability

We have also measured the reaction of the topologies to different link failure probabilities given a packet generation probability of 0.10. As expected the packet delivery ratio degrades and DCSPT2 and SPT2 are affected the most by the link failure due to the placement of the sink. The other topologies offer a steady performance, where the Gabriel-based combined approaches are the best. The average number of hops also increases due to exploring the suboptimal routes. The results are shown in Figure 9, 10 and 11.

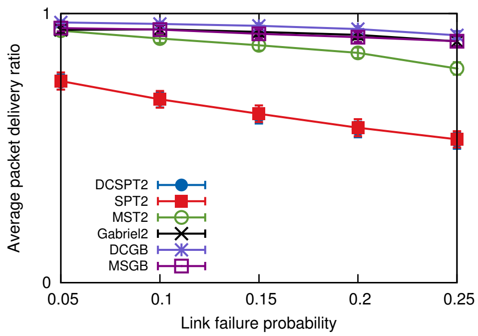

Fig. 9. The average packet delivery ratio with different link failure probability.

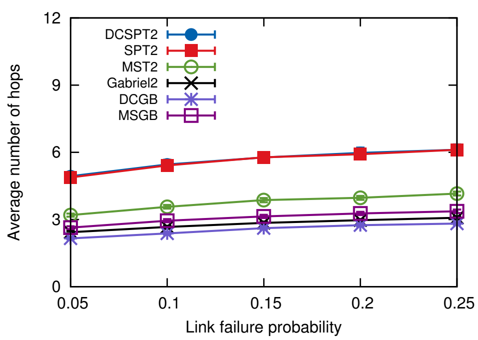

Fig. 10. The average number of hops with different link failure probability.

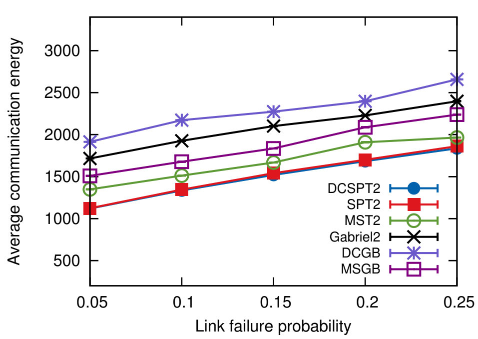

Fig. 11. The average communication energy with different link failure probability.

Observation: The performance evaluation in the dynamic environment reveals that the Gabriel-based topologies have the best performance in terms of both the reliability and other performance metrics. However, it is interesting that the packet generation probability has more impact on the performance compared to the link failure probability. The topologies show better resilience to loss than to congestion.

VIII. Performance Enhancement of the Gabriel-based Topology

The performance analysis in the previous two sections suggest that the Gabriel-based 2-edge connected reliable topology offers the best communication energy consumption (a critical issue in WSNs) and good path length with high throughput. In addition, the algorithm to find the routing topology can be implemented in a completely distributed way. There are distributed TDMA scheduling algorithms that could be applied as well. In the following, we now focus only on the Gabriel-based topology and present its performance in the dynamic environment with new local forwarding strategies.

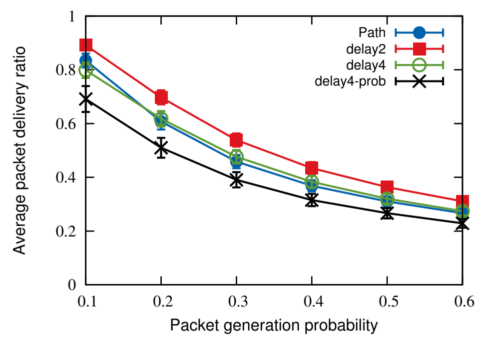

Fig. 12. The average packet delivery ratio for Gabriel based 2-edge connected topology.

Note that, in the previous experiments, DCSPT was used as the basis for finding routing topologies because it is most conservative in terms of sensor density requirement in the physical network. The same underlying physical topology was used for the rest of the topologies. Now, concentrating on only Gabriel2 networks, we are able to reduce the radio range of the topology and still find connected routing topologies. In particular, the radio range is reduced from 25m to 18m. This immediately ensures lower node degree and less energy consumption.

We focus on developing a new forwarding protocol to alleviate the packet dropping issue. We observe that if packets reach the sink quickly, less congestion occurs at the sensors and this leads to a high throughput. As Gabriel2 ensures a good route length from the sensors to the sink, packets have the chance to reach the sink quickly at low packet generation rates. However, at the high rates, sensors may become congested and start dropping packets resulting in lower throughput. One way of handling this situation is to consider the queue size of the sensors while routing packets. In Algorithm 2 we present such an algorithm.

Each node has at least 2 choices of parent. $W_1(v_i)$, $W_2(v_i)$, and $N(v_i)$ represent the best-route and the second best-route parents, and the neighbors of a sensor node v_i, respectively. $Queue(W_1(v_i))$ and $MaxSize$ are the current queue size of $W_j(v_i)$ and the maximum queue size of a sensor, respectively. We consider the same max queue size of 2000 packets for each sensor.

Algorithm 2 Queue based routing

Initialization:
for $i \leftarrow 1$ to $|V|-1$ **do**
  set two best parents $W_1(v_i) \in N(v_i)$ and $W_2(v_i) \in N(v_i)$ for each sensor node v_i.
end for
Forwarding at node i***:***
if $Queue(W_1(v_i)) \geq 0.9 \times MaxSize$ **then**
  $NHop(v_i) \leftarrow W_2(v_i)$
else
  $NHop(v_i) \leftarrow W_1(v_i)$
end if

$NHop(v_i)$ sets the next hop parent of a sensor v_i.

The Algorithm 2 states that initially each sensor chooses their two best parents, $W_1(v_i)$ and $W_2(v_i)$, along the optimal and the next optimal routes. During initial packet forwarding a sensor sends packet to $W_1(v_i)$ and waits for the ACK that includes the current queue status. In the subsequent forwarding, it checks whether the queue utilization of $W_1(v_i)$ reaches 90%. In that case, the packet is forwarded to $W_2(v_i)$. Note that the retry limit for a packet forwarding is 7, after which the packet is dropped.

We compare 4 algorithms in Figure 12. *path* is our previous way of forwarding that alternates between the 2 best (shortest hop) parents. We have 3 variations of the queue-based routing: *delay2* that chooses the best parent as long as it is not overloaded considering only the two best paths, *delay4* chooses from up to 4 of the best parents and the 4th algorithm, *delay4-prob*, chooses from up to 4 parents based on a probability derived from the relative queue sizes. The simulation results show that considering queue size helps improve the throughput of the basic 2 parents scenario. The other two variants that consider more parents and potentially longer paths, may take a longer time to reach the sink. Thus following a shorter route along the less congested nodes seems useful. We have also checked the average delay in terms of the number of hops from a sensor to the sink while using the above 4 packet forwarding strategies on Gabriel2. The results in Figure 13 show that *delay4-prob* experiences short delay in case of a successful packet delivery. The route of *path* is slightly longer with a higher throughput compared to the probabilistic approach. Depending solely on the queue size to forward the next-hop packet may lead the packet to follow a slightly longer route.

IX. Conclusions

We have designed a set of reliable 2-edge connected routing topologies to provide alternative paths for WSNs. Using simulation, we analyze the performance with respect to reliability and path quality in the presence of link failure. It is apparent that there is no particular topology that optimizes all the required metrics, but some can offer a nice balance between them. In particular, the Gabriel graph structures provide a good balance between reliability and other desirable

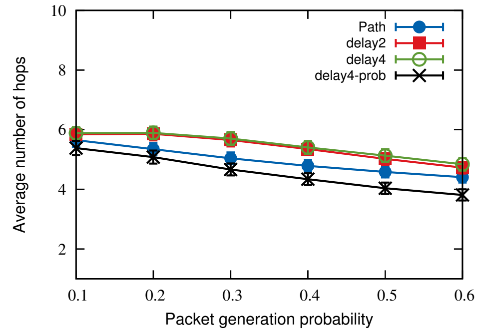

Fig. 13. The average number of hops for Gabriel based 2-edge connected topology.

properties. We evaluate different forwarding algorithms and find that basing choice of path on a congestion indication such as queue length improves throughput.

Acknowledgements

This research is funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] A. Ghosh, O. Incel, V. Kumar, and B. Krishnamachari, "Multichannel scheduling and spanning trees: Throughput delay tradeoff for fast data collection in sensor networks," *IEEE Trans. on Networking*, May 2011.

[2] E. Gelal, G. Jakllari, S. V. Krishnamurthy, and N. E. Young, "Topology control to simultaneously achieve nearoptimal node degree and low path stretch in ad hoc networks," in *In the proc. of IEEE SECON*, Sept. 2006, pp. 431–439.

[3] H. K. Qureshia, S. Rizvia, M. Saleemc, S. A. Khayama, V. Rakocevicb, and M. Rajarajanb, "Poly: A reliable and energy efficient topology control protocol for wireless sensor networks," *Computer Comm.*, vol. 34, no. 10, pp. 1235–1242, July 2011.

[4] K. Staniec and G. Debita, "Evaluation of topological planarity and reliability for interference reduction in radio sensor networks," *EURASIP Jour on Wireless Comm. and Net. 2012*, vol. 2012, no. 56, Feb. 2012.

[5] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme, "Using local geometry for tunable topology control in sensor networks," *IEEE Trans. on Mobile Computing*, vol. 8, no. 2, pp. 218–230, 2009.

[6] K. Miyao, H. Nakayama, N. Ansari, and N. Kato, "LTRT: An efficient and reliable topology control algorithm for ad-hoc networks," *IEEE Trans. on Wireless Comm.*, vol. 8, no. 12, pp. 6050–6058, Dec. 2009.

[7] M. Masdari and M. Tanabi, "Multipath routing protocols in wireless sensor networks: A survey and analysis," *Future Generation Comm. and Net.*, vol. 6, no. 6, pp. 181–192, 2013.

[8] N. Li, J. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," in *Proc. of INFOCOM*, March 2003, pp. 1702–1712.

[9] J. Bang-Jensen, S. Thomasse, and A. Yeo, "Small degree out-branchings," *Jour of Graph Theory*, vol. 42, no. 4, pp. 297–307, 2003.

[10] K. Gabriel and R. Sokal, "A new statistical approach to geographic variation analysis," in *Systematic Zoology*, 1969, pp. 259–278.

[11] S. Ergen and V. Pravin, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Net.*, vol. 16, no. 4, pp. 985–997, May 2010.