International Journal of Computing and Digital Systems

PRM Mitigation Scheme for the Nodes Failure Effect in the WSNs CGD Protocol

Sarah Yasseen Alshareeda[1]

[1]*Dept. of Computer Engineering, College of Engineering, Basrah University, Iraq*
e-mail: sarahdiginets@yahoo.com

Received 28 May 2012, Revised 31 Oct. 2012, Accepted 10 Dec. 2012

Abstract: This work proposes a proactive reactive scheme to mitigate the effect of random permanent node(s) failure on the Ideal Shortest Path Tree (ISPT) Spreading rules in the Wireless Sensor Networks' (WSNs) Contour Guided Dissemination (CGD) protocol. To prove the efficiency of the scheme, simulations have been conducted using the OMNET^{++} simulator; the simulation results of this work demonstrate how the random node failures affect/degrade the ISPT performance and how the new Proactive Reactive Mitigation (PRM) scheme reinstates the performance by delivering the messages between contour rows with least message loss rate and without introducing an excessive overhead to the used quality of service metrics: average delay, jitter, and throughput compared with the ISPT; its performance is approximately the same of the ISPT.

Keywords: WSNs, Node failure, CGD Dissemination Protocol.

I. INTRODUCTION

The promising technology of Wireless Sensor Networks (WSNs) is altering our world because of their ubiquity and suitability in a wide range of applications. Normally, a WSN contains a huge number of resource-constrained nodes, which are deployed either randomly or at prefixed locations, to sense certain physical phenomena. Upon detection, these nodes coordinate among themselves to produce high quality information to be delivered to special nodes called sinks or base stations [1, 2].

Because of the radio range limitations and energy consumption constraints, sensor nodes, which can't communicate directly with the sink(s), can route or disseminate their information to nearby sensors in a multihop fashion until data reach the base station(s) [3]. Thus, many routing and dissemination protocols have emerged. In all of them, the ability to reliably deliver information among nodes is important especially when the network suffers disturbances such as link failure, RF interference, and node failure.

Hence, it is necessary to design dissemination methods that can cope with such disruptions. This work focuses on the design of such method to handle the random node(s) failure in the Ideal Shortest Path Tree (ISPT) spreading scheme of the WSN Contour Guided Dissemination (CGD) protocol.

II. BACKGROUND

Roughly speaking, a WSN can suffer from two types of node failures. The first type is the random individual node failure caused by some internal problem in node, battery drain. The second one is an area failure where the nodes covering an area may fail altogether because of a natural disaster (like an earthquake), bomb explosion, fire, or successful Denial of Service attacks [4]. In both cases, once detected, misbehaving nodes must be isolated from the rest of the sensor network, no longer used by running applications. Hence, other nodes must compensate.

A. *Recovering from Node Failure*

The permanent effects of node failure should not affect the overall task of the WSN. Defective sensors must be excluded from the communication. One of the ways that is used to overcome such faults is rerouting. Rerouting means the formation of alternative paths to reroute data around failed nodes. Rerouting protocols are classified into two groups [5, 6]:

1) *Multi-path Routing Protocols [5, 7]*

Multi-path protocols enhance network performance and increase the likelihood that (important) data eventually reaches its destination in the face of failed nodes by maintaining multiple paths between the source and sink at the expense of increased energy consumption and traffic generation. Two distinct mechanisms exist for sending redundant packets through multiple paths, disjoint and braided [8].

2) *Single-path with Repair Routing Protocols*

Single-path with repair schemes [8] forward data along a single path, repair the path only when a break is detected, and apply local flooding for finding alternative paths. Path repair approaches are different depending on how the original paths are established and what the reason of path break is. Generally speaking, there are two kinds of path repairing: source-initiated path repair and local path repair.

- Source-initiated path repair works such that, when a link failure occurs, the break upstream node creates a route error message, and sends it all the way back to the source. The source node is responsible for finding an alternate path and resending the packet. Source-initiated path repair works fine for short routes. In this sense, source- initiated path repair does not scale to large networks [8].
- Local path repair only informs the immediate upstream node, i.e., the broken path is repaired locally without disturbing the source and other upstream nodes [8].

Comparing multipath with single-path, robustness is achieved through redundancy, which means more energy cost. In a network with low node failure ratio, most of this extra energy consumption is wasted. Compared to multipath routing, single-path routing with local repair is more energy efficient. In any case, energy/robustness trade off should always be kept in mind when designing a WSN routing protocol [8].

III. CONTOUR GUIDED DISSEMINATION PROTOCOL

The traditional dissemination methods commonly disseminate data in a multihop fashion, but they do not fully utilize all the available paths between a pair of nodes [1, 9, and 10]. In contrast, CGD is a multihop multipath forwarding sender-appointed dissemination method, which exploits the location of each node and all of the available shortest paths between a pair of nodes (called source and sink). The CGD protocol relies on four main definitions [11]:

Definition1: CGD uses a fixed 2D grid topology of ***IxJ*** embedded nodes, each ***n(i,j)*** node has ***eight*** neighbors, i.e., CGD uses ***Ξ8 E***mbedding.

Definition2: In ***Ξ8*** Embedding, there exist multiple shortest paths between a pair of nodes ***n(i,j)*** and ***n(q,r)*** that are located at ***(i,j)*** and ***(q,r)*** on the 2D grid, respectively. The union of all the shortest paths between source and sink is called ***Contour***. Each path in the contour is different from every other path by at least one edge. The nodes that are at the same distance from the source are collectively referred to as a row in the contour: each row has width ***w***, where w_i refers to the number of nodes in row ***i***. CGD uses ***Orientation1*** contours because the source node in ***O1*** octant of the 2D plane and the sink is at its center as shown in Fig.1.

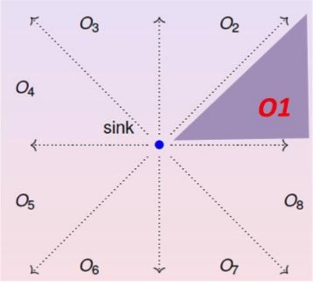


Figure 1. CGD Orientations

Definition3: In ***Ξ8***, ***O1***, the shape of the contour is determined by the values of x_{diff}, y_{diff}, xy_{diff}, and ***k*** where:

If $n_{(i,j)}$ *and* $n_{(q,r)}$ *are source and sink nodes, respectively, then:*

$$x_{diff}\,(n_{(i,j)},\, n_{(q,r)}) = |q-i| \quad (1)\ [11]$$

$$y_{diff}\,(n_{(i,j)},n_{(q,r)}) = |r-j| \quad (2)\ [11]$$

$$xy_{diff}\,(n_{(i,j)},n_{(q,r)}) = |x_{diff} - y_{diff}| \quad (3)\ [11]$$

$$k(n_{(i,j)},n_{(q,r)}) = \lfloor \frac{(xy\,diff)}{2} \rfloor \quad (4)\ [11]$$

Hence, in ***Ξ8***, ***O1***, values of $\mathbf{x_{diff}}$, $\mathbf{y_{diff}}$, $\mathbf{xy_{diff}}$, and **k** determine the general shape of a contour; six contour shapes are raised (***uni-path***, ***parallelogram***, ***symmetric and non-symmetric rectangle***, ***symmetric and non-symmetric hexagon***). In all of these six shapes, every contour is divided into three regions: ***expansion***, ***propagation***, and ***contraction*** [11]; fig.2 presents these shapes and their three regions.

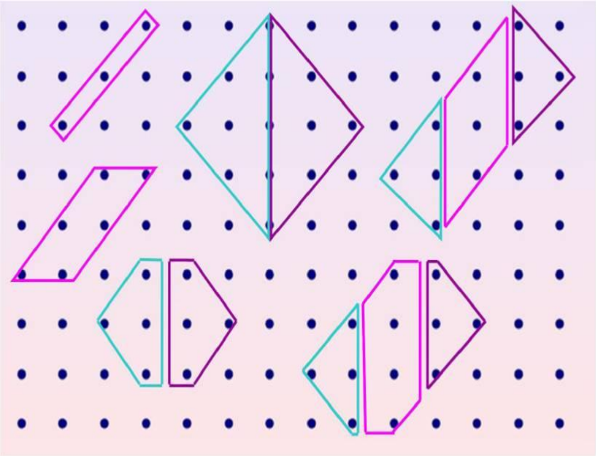


Figure 2. Contour Shapes and Regions in a 2D Grid Topology

Definition4: Within contour, three spreading schemes are suggested to disseminate the messages: ***uniform***, ***optimal***, and ***ISPT*** spreading [11]; all assume ***Ξ8***, ***O1*** contours where every node in row_i has a maximum of three reachable downstream nodes in row_{i+1}, a maximum of three reachable upstream nodes in row_{i-1}, and a maximum of two reachable sidestream nodes on row_i. In the ISPT spreading, the ***source*** node sends its messages to the ***sink*** node along a tree of shortest paths between the two: each of length equals $\boldsymbol{Max(x_{diff}, y_{diff})}$, i.e., the optimal number of hops between the source and the [11]. Fig.3 shows the ISPT spreading paths.

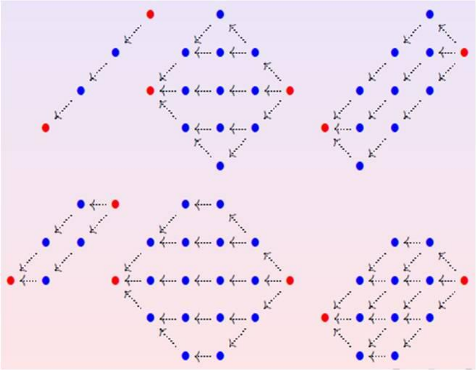


Figure 3. ISPT Spreading Paths

IV. THE PRESENTED PRM SCHEME

With no node(s) failure presented in the ISPT network, all data is successfully delivered from source to sink, but what happens when some random nodes fail; this certainly affects the ISPT performance. As a solution, ***data must be rerouted around the failed nodes***. Thus, this work presents a Proactive Reactive Mitigation (**PRM**) rerouting algorithm to extend the ISPT spreading strategy of the CGD dissemination protocol for a scenario that includes nodes failures using pre-established multiple paths between the source and the sink. The idea is to simply modify the ISPT spreading rules of expansion, propagation, contraction, and transition between regions in a way that as soon as a node in row_i (parent node) receives a failure notification(s) from its downstream children in row_{i+1}, it would select alternative pre-stated paths, called PRM paths, other than the previously used ISPT ones. Since the effects of failure are considered permanent, this method will keep use every new selected PRM path until the end of the network lifetime.

The suggested PRM scheme is ***Proactive*** for using pre-established paths (PRM paths) to reroute data and ***Reactive*** because it uses these paths only when node failures occur. The presented model assumes that periodically every node senses its own status and communicates it to its parent(s). If the received status report says that the child is healthy, e.g., its battery is still working; the parent continues to use the ISPT spreading paths. When the received report says that the child is about to be faulty (e.g. its battery is about to be drained), the parent should use the new PRM established path(s). Fig.4 shows the general behavior that every node within the contour should follow.

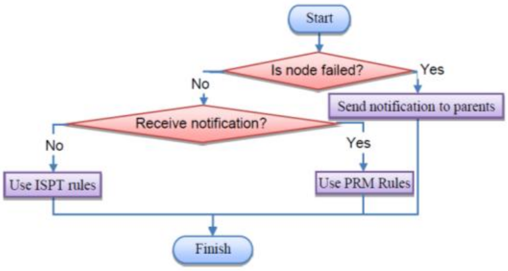


Figure 4. General Node Behavior

A. *PRM Rules in the Contour Three Regions*

To accomplish the work of the PRM scheme, three general spreading rules have been presented to suit the three contour regions: expansion, propagation, and contraction.

1) *PRM Rules in Expansion Region*

*If **w** (odd) is the width of some row in the expansion region of any shaped contour, and **w+2** is the width of the next row. If **N** is the number of messages handled by each node on the **w** width row. Let the nodes on either side of the middle node* $\left\lfloor \frac{w}{2} \right\rfloor + \mathbf{1}$*, in the row of **w** width, be labeled **1, 2, 3, . . .,*** $\left\lfloor \frac{w}{2} \right\rfloor$ *and the nodes on either side of the middle node* $\left\lfloor \frac{w+2}{2} \right\rfloor + 1$*, in the next row, be labeled **1, 2, 3, . . .,*** $\left\lfloor \frac{w+2}{2} \right\rfloor$*. Then, the diagonal node with **i=1**, has two children nodes **i, i+1** in the next row, sends all its messages to one of its children if and only if the other child is failed. Each node with **i≠1**, has one downstream node **i+1**, sends all of its messages to the node **i** in the next row if and only if its child node **i+1** is failed. The middle node* $\left\lfloor \frac{w}{2} \right\rfloor + 1$*, has one downstream node* $\left\lfloor \frac{w+2}{2} \right\rfloor + 1$*, evenly divides its messages between the two immediate neighbors* $\left\lfloor \frac{w+2}{2} \right\rfloor, \left\lfloor \frac{w+2}{2} \right\rfloor$ *of its child in the next row, if and only if **N** is even and the child is failed; if **N** is odd, it alternatively sends* $\frac{N+1}{2}, \frac{N-1}{2}$ *to the two immediate neighbors of the middle downstream node in the next row.*

```
w=input('row_width'); %w is the width of the row where
current node lies
N=input('No_of_messages') % N messages at each node
for i=1:floor(w/2)+1 % n is a node in current row, m is a
node in next row
if i==1; then
    if (mod(N,2))==0 then m(i)=n(i)/2 else m(i)=(n(i)/2)+1;
    if (mod(N,2))==0 then m(i+1)=n(i)/2 else
m(i+1)=(n(i)/2)-1;
elseif i==floor(w/2)+1; then
    if (mod(N,2))==0 then m(floor((w+2)/2))=n(i)/2 else
m(floor((w+2)/2))=n(i)/2+1;
else
m(i)=n(i);
end;
end;
```

Fig.5 compares the old ISPT rules with the new PRM rules in this region.

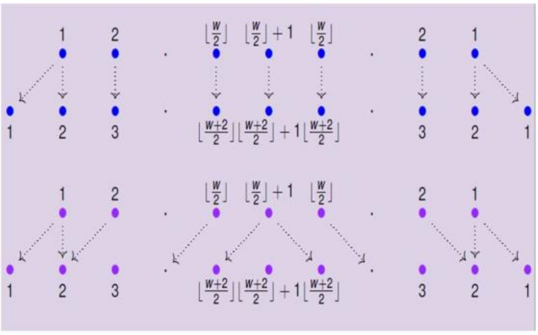


Figure 5. ISPT vs. PRM Rules in Expansion Region

2) PRM Rules in Propagation Region

*If **w** is the width of some row in the propagation region of any shaped contour, or of the last rows in the expansion and the propagation regions of the non-symmetric rectangle shaped contours. If **N** is the number of messages handled by each node on the row with **w** width. Let node **i** be labeled **1, 2, 3, .., w**. Then, the diagonal node with **i=1**, has one child node **i** in the next row, sends all its messages to its neighbor sidestream node **i+1**, in its row, if and only if its child is failed. Each node with **i≠1**, has one downstream node **i**, sends all of its **N** messages to the node **i-1** in the next row if and only if its child node **i** is failed.*

```
w=input('row_width'); %w is the width of the row where
current node lies
for i=1:w % n is a node in current row, m is a node in next
row
if i==1; then
    n(i+1)=n(i)
else
m(i-1)=n(i);
end;
end;
```

Fig.6 compares the old ISPT rules with the suggested PRM in the propagation region.

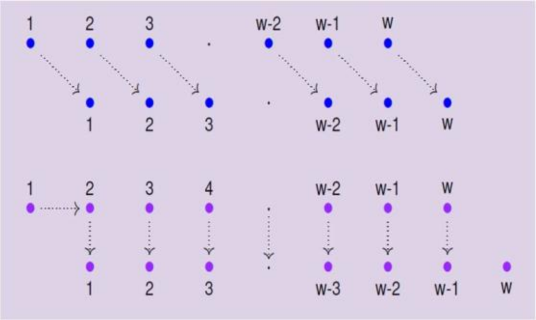


Figure 6. ISPT vs. PRM Rules in Propagation Region

3) PRM Rules in Contraction Region

*If **w** (odd) is the width of some row in the contraction region of any shaped contour and **w-2** is the width of the next row. If **N** is the number of messages handled by each node on the row with **w** width. Let the nodes on either side of the middle node* $\left\lfloor\frac{w}{2}\right\rfloor+\mathbf{1}$*, in the row of w width, be labeled **1, 2, 3, . . .,*** $\left\lfloor\frac{w}{2}\right\rfloor$ *and the nodes on either side of the middle node* $\left\lfloor\frac{w-2}{2}\right\rfloor+\mathbf{1}$*, in the next row, be labeled **1, 2, 3, . . .,*** $\left\lfloor\frac{w-2}{2}\right\rfloor$*. Then, the diagonal node with **i=1**, has one child node **i** in the next row, sends all its messages to its neighbor sidestream node **i+1**, in its row, if and only if its child is failed. Each node with **i≠1**, has one downstream node **i-1**, sends its entire **N** messages to the node **i** in the next row if and only if its child node **i-1** is failed. The middle node* $\left\lfloor\frac{w}{2}\right\rfloor+\mathbf{1}$*, has one downstream node* $\left\lfloor\frac{w-2}{2}\right\rfloor+\mathbf{1}$ *, evenly divides its messages between the two immediate neighbors* $\left\lfloor\frac{w-2}{2}\right\rfloor$*,* $\left\lfloor\frac{w-2}{2}\right\rfloor$ *of its child in the next row, if and only if **N** is even and the child is failed; if **N** is odd, it alternatively sends* $\frac{N+1}{2}$ *,* $\frac{N-1}{2}$ *to the two immediate neighbors of the middle downstream node in the next row.*

```
w=input('row_width'); %w is the width of the row where
current node lies
N=input('No_of_messages') % N messages at each node
for i=1:floor(w/2)+1 % n is a node in current row, m is a
node in next row
if i==1 then
    n(i+1)=n(i)
elseif i==floor(w/2)+1; then
    if (mod(N,2))==0 then m(floor((w-2)/2))=n(i)/2 else
m(floor((w-2)/2))=n(i)/2+1;
else m(i)=n(i); end; end; end;
```

Fig.7 compares the old ISPT rules with the suggested PRM in the contraction region.

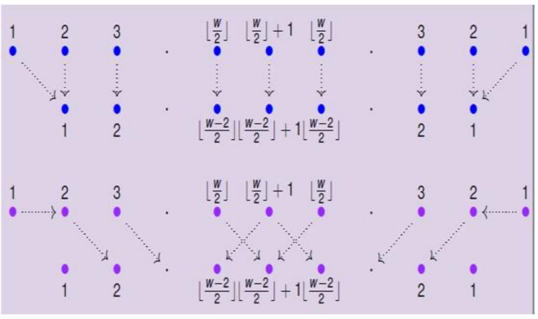


Figure 7. ISPT vs. PRM Rules in Contraction Region

Extra five rules are presented to deal with some special cases:

4) PRM Rules for Expansion to Propagation Transition in Non-symmetric Hexagon Contours

*If **w** (odd) is the width of the last row in the expansion region of non-symmetric hexagon contour, and **w+1** is the width of the next row, first row in its propagation region. If **N** is the number of messages handled by each node on the row with **w** width. Let the nodes, in the row with **w** width, be labeled **1, 2, 3, . . ., w**, and nodes, in the next row, be labeled **1, 2, 3, . . ., w+1**. Then, each node **i**, has two children nodes **i, i+1** in the next row, sends all its messages to one of its children if and only if the other child is failed. See Fig.8.*

```
w=input('row_width'); %w is the width of the row where
current node lies
N=input('No_of_messages') % N messages at each node
for i=1:w % n is a node in current row, m is a node in next
row
if (mod(N,2))==0 then m(i)=m(i+1)=n(i)/2
else m(i)=(n(i)/2)+1 && m(i+1)=(n(i)/2)-1;
end;
```

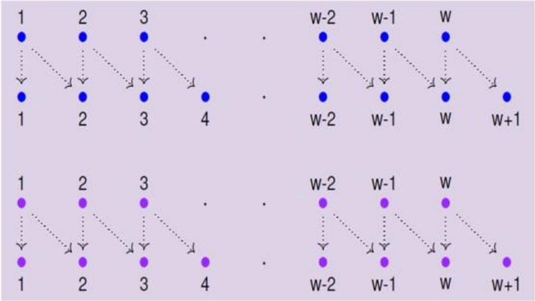


Figure 8. ISPT vs. PRM in Expansion to Propagation Transition in Non-symmetric Hexagon Contours

5) PRM Rules for Propagation to Contraction Transition in Non-symmetric Hexagon Contours

*If **w** (even) is the width of the last row in the propagation region of non-symmetric hexagon contour, and **w-1** is the width of the first row in its contraction region. If **N** is the number of messages handled by each node on the row with **w** width. Let the nodes, in the row with **w** width, be labeled **1, 2, 3, . . ., w**, and the nodes, in the next row, be labeled **1, 2, 3, . . ., w-1**. Then, node with **i=1**, has one child node **i**, sends its messages to its neighbor sidestream node **i+1**, in its row, if and only if its child is failed. Similarly, node with **i=w**, has one child node **w-1**, sends its messages to its neighbor sidestream node **i-1**, in its row, if and only if its child is failed. Every other node with **i≠ 1 ≠w**, has two children nodes **i, i-1** in the next row, sends all its messages to one of its children if and only if the other child is failed. See Fig.9.*

```
w=input('row_width'); %w is the width of the row where
current node lies
N=input('No_of_messages') % N messages at each node
for i=1:w % n is a node in current row, m is a node in next
row
if i==1; then n(i+1)=n(i)
elseif i==w; then n(i-1)=n(i)
else
    if (mod(N,2))==0 then
    m(i)=m(i+1)=n(i)/2
    else
    m(i)=(n(i)/2)+1 && m(i+1)=(n(i)/2)-1;
    end;
end;
end;
```

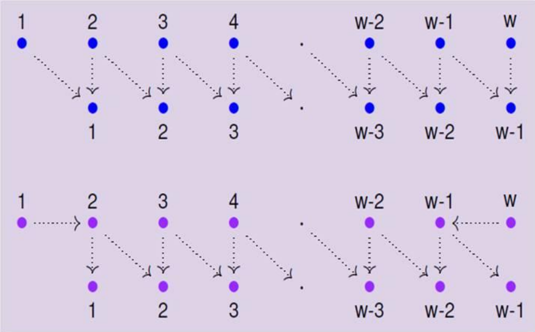


Figure 9. ISPT vs. PRM in Propagation to ContractionTransition in Non-symmetric Hexagon Contours

6) PRM Rules for Expansion to Contraction Transition in Symmetric Hexagon Contours

*If **w** (odd) is the width of the (last row in the expansion region) | (first row in the contraction region) of a symmetric hexagon contour. If **N** is the number of messages handled by the nodes in the last expansion region row. Let the nodes, in both rows, be labeled **1, 2, 3, . . ., w**. Then, every node **i≠w**, has one child node **i**, sends its messages to node **i+1**, in the next row, if and only if its child is failed. While node **i=w**, has one child node **i**, sends its messages to its neighbor sidestream node **i-1**, in its row, if and only if its child is failed. See Fig.10.*

```
w=input('row_width'); %w is the width of the row where
current node lies
for i=1:w % n is a node in current row, m is a node in next
row
if i==w; then
    n(i-1)=n(i)
else
    m(i+1)=n(i)
end;
end;
```

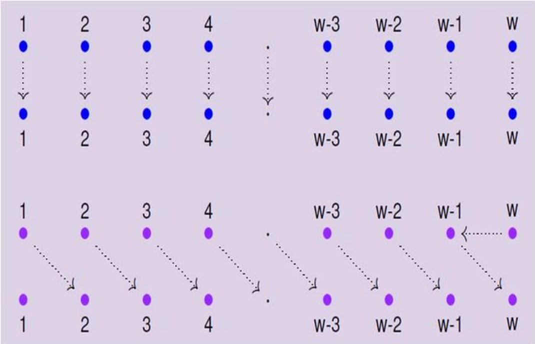


Figure 10. ISPT vs. PRM in Expansion to Contraction Transition in Symmetric Hexagon Contours

7) PRM Rule for Source Node

*If **N** is the number of messages handled by source node of any contour shape. Then, the source node will divide its **N** messages between the available alive children that reside inside or even outside the contour.*

8) PRM Rules for Uni-path Contour with w=1

Using rules of 3.2, node i, in a row with w=1 width of a uni-path shaped contour (compromises only a propagation region), sends its messages to node i-1, outside the contour, if and only if its child node i, in the next row, is failed. The receiving node i-1 keeps follow its own path outside the contour to deliver the messages to sink node.

To summarize the overall behavior of the source node, every intermediate node, and sink node in the PRM scheme, flowcharts A.1 is presented in Appendix A.

V. THE USED QUALITY OF SERVICE METRICS

In the PRM scheme, the QoS is accepted as a measure of the service quality that the network offers to the applications or users; it is concerned with how the underlying communication network delivers the QoS-constrained sensor data to the sink while efficiently utilizing network resources and corresponding requirements. Our model proposes to define WSN QoS in terms of:

- **N**: The **N**umber of messages handled in each node is used as a base metric to evaluate PRM scheme. The scenario assumes the source node sends 15 messages to the sink; each message is 36 bytes in length.
- **MLR**: **M**essage **L**oss **R**ate is the percentage ratio of number of messages not received at the sink (lost in the network) to the number of messages generated at the source.

$$\text{MLR}=\left(\frac{(\text{Generated Source} - \text{Received Sink})}{\text{Generated Source}}\right) \text{x}100\% \qquad (5)$$

- **AL**: In a network, **A**verage **L**atency, a synonym for average delay, is the average end-to-end delay of every message that is sent from source to sink, measured in **msec**.

$$\text{AL}=\frac{\sum(\text{Sim.time}_{\text{message}} - \text{Creationtime}_{\text{message}})}{\text{No.of Messages}} \qquad (6)$$

- **TP**: Generally, **T**hrough**P**ut is the amount of data transmitted between source and sink in a given amount of time. In our model, we use the throughput parameter to represent the mean value of the total transferred bits within a specific simulation time period.

$$\text{TP}=\frac{\text{No.of Messages} * \text{Message Length}}{\text{Average Latency}} \qquad (7)$$

- **J**: **J**itter is a measure of the variability over time of the delay across a network, measured in **msec**.

VI. SIMULATION AND RESULTS

To get a better insight to the behavior and performance of the suggested PRM scheme, the OMNeT++ simulator [12] is used. The used simulation method models:

- At design phase:
    - The multihop propagation of messages as a multi-stage queuing network.
    - All nodes in the network are located at predetermined locations.
    - Both source and sink nodes are predefined to be the pair of nodes that the tree of paths is going to be built between them.
    - The source sends 15 messages to the sink.
    - All ISPT and PRM paths are pre-established but used when they'd be needed.
- At runtime phase:
    - Randomly, the user selects the nodes that are wanted to be faulty.

The simulation work has extensively investigated the performance of the ISPT spreading and the PRM spreading in face of nodes failures for different x_{diff}, y_{diff} cases in all of the six contour shapes. To exemplify the established work, Figures 11 through 16 show the Graphical User Interface (GUI) layouts as well as the MLR for the ISPT rules (with no failed nodes and with failed nodes), and the PRM spreading in two samples:

- The non-symmetric rectangle contour, with x_{diff}=10 and y_{diff}=4, to represent the even xy_{diff} contours.
- The symmetric hexagon contour with x_{diff}=9 and y_{diff}=0, to represent the odd xy_{diff} contours.

By rerouting data around every failed node using the PRM rules, the same amount of the generated number of messages at the source is received by the sink node. In Table.1, an average MLR is shown for some even and odd values of xy_{diff}.

The proposed PRM spreading scheme must not introduce an excessive overhead to the average latency **AL**, mean jitter **J**, and average throughput **TP** metrics in comparison with the ISPT spreading scheme into both even and odd $\boldsymbol{xy_{diff}}$ and under varying number of hops between source and sink. Referring to the performance curves in figures 17 to 19, it can be seen that the PRM approximates the ISPT in both even and odd $\boldsymbol{xy_{diff}}$ contours.

TABLE I. MLR IN ISPT, ISPT + FAILURES, & PRM FOR SOME XYDIFF VALUES

xy_{diff}	**MLR %**		
	ISPT (without failures)	**ISPT** (with failures)	**PRM**
0	0	100	0
1	0	72.14286	0
2	0	49.5	0
3	0	66	0
4	0	56.5	0
5	0	64	0
6	0	53	0
7	0	75.33333	0
8	0	73	0
9	0	68.66667	0
10	0	90	0
11	0	73	0

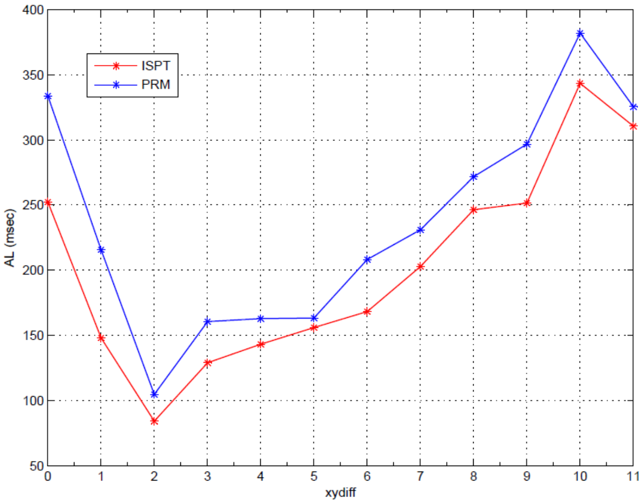


Figure 11. ISPT vs. PRM in Terms of Average Latency in different xy_{diff} Settings

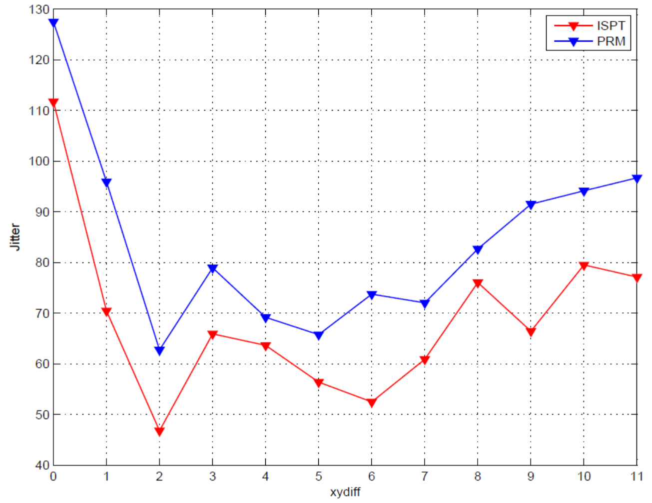


Figure 12. ISPT vs. PRM in Terms of Mean Jitter in different xy_{diff} Settings

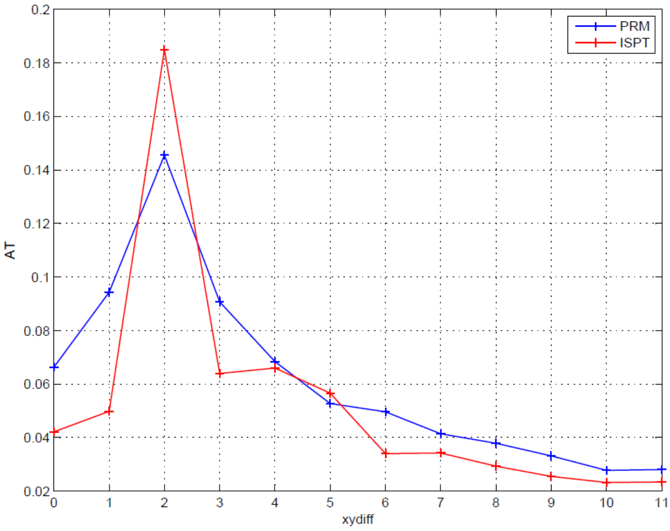


Figure 13. ISPT vs. PRM in Terms of Average Throughput in different xy_{diff} Settings

Observation 5.1*: The AL measure in PRM introduces ≅ 85 msec delay compared with the ISPT.*

Observation 5.2*: The PRM mean jitter measure shows approximately a 23.5 msec variance upon the ISPT.*

Observation 5.3*: The PRM and the ISPT throughput measurements tend to be the same.*

VII. CONCLUSION

The ability to keep the dissemination protocols work smoothly in face of disturbances is important. This work investigates a method to mitigate the effect of node failures in the ISPT spreading of the WSN CGD protocol. It sets some rules to reroute data around failed sensors. The conducted simulation results show that the PRM scheme is straightforward and leads to significant results; the new PRM rules do not put excessive overhead on the original ISPT rules quality of service in terms of message loss rate, average latency, jitter, and throughput.

REFERENCES

[1] Akyildiz, I.F., et al., "Wireless sensor networks: a survey," Computer Networks, Vol. 38, No.4, pp. 393-422, ISSN 13891286, 2002.

[2] Zory J. and Vanzago L., "Introduction to Wireless Sensor Networks," the ST Journal of Research Vol.4, No. 1, 2007.

[3] Björkstad P., "Reliable Data Delivery over Wireless Sensor Networks," Master Degree Thesis, EE Department, KTH Royal Institute of Technology, Stockholm, Sweden, Jan. 18, 2007.

[4] J. Deng, R. Han, and Sh. Mishra, "A Robust and Light-Weight Routing Mechanism for Wireless Sensor Networks," 1st Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS 2004), Florence, Italy, June 2004.

[5] L. Wang, "Survey on Sensor Networks," Department of Computer Science and Engineering, Michigan State University, 2004.

[6] H. Alwan, A. Agarwal, "A Survey on Fault Tolerant Routing Techniques in Wireless Sensor Networks," 3rd International Conference on Sensor Technologies and Applications SENSORCOMM '09, pp: 366-371, 2009.

[7] M. Radi, B. Dezfouli, K. Abu Bakar, and M. Lee, "Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges," Sensors, Vol.12, No.1, Basel, Switzerland, 2012.

[8] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," Mobile Computing and Communications Review, Vol.1, No.2, pp: 251-254, 2001.

[9] Heinzelman W. R., Kulik J., and Balakrishnan H., "Adaptive Protocols for Information Dissemination in Wireless Sensor Network," In Proceedings of ACM MOBICOM 1999, pp. 174-185, 1999.

[10] Intanagonwiwat C., Govindan R., Estrin D., Heidemann J., and Silva F., "Directed diffusion for wireless sensor networks," IEEE/ACM Transactions on Networking, Vol. 11, No. 1, pp. 2–16, 2003.

[11] Chu I-H., Duan M., and Sastry S., "Contour Guided Dissemination for Networked Embedded Systems," International Journal of Distributed Sensor Networks, Vol. 5, No. 5, pp. 502-530, 2009.

[12] Varga A., "OMNET++ - Discrete Event Simulation System Version 3.2 User Manual," http://www.omnetpp.org.

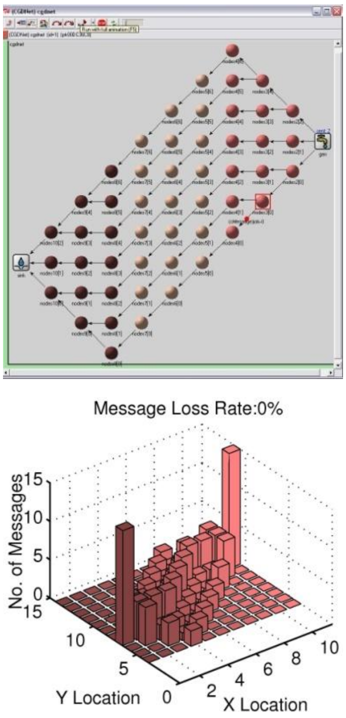


Figure 11. ISPT spreading and MLR with no nodes failed in non-symmetric rectangle contour with $x_{diff}=10$ and $y_{diff}=4$

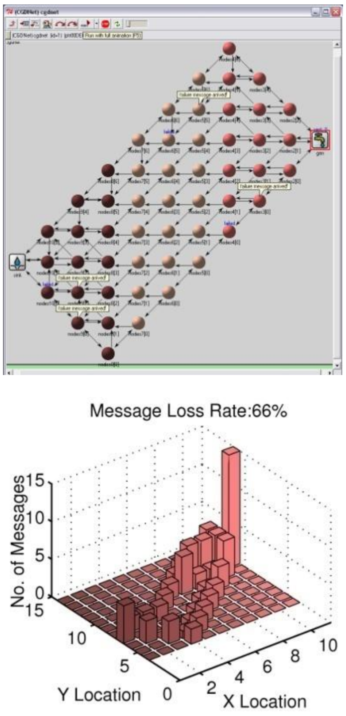


Figure 12. ISPT spreading and MLR with three failed nodes in non-symmetric rectangle contour with $x_{diff}=10$ and $y_{diff}=4$

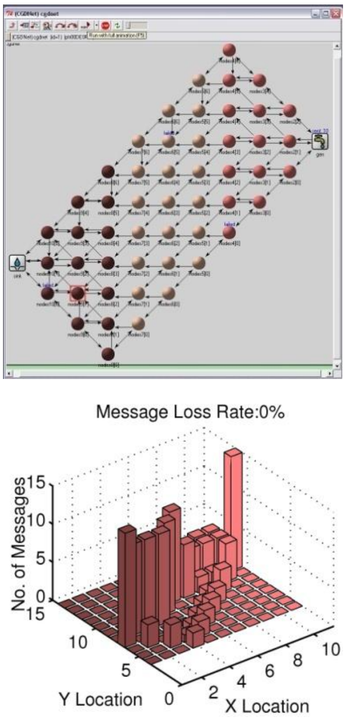


Figure 13. PRM spreading and MLR with three failed nodes in non-symmetric rectangle contour with $x_{diff}=10$ and $y_{diff}=4$

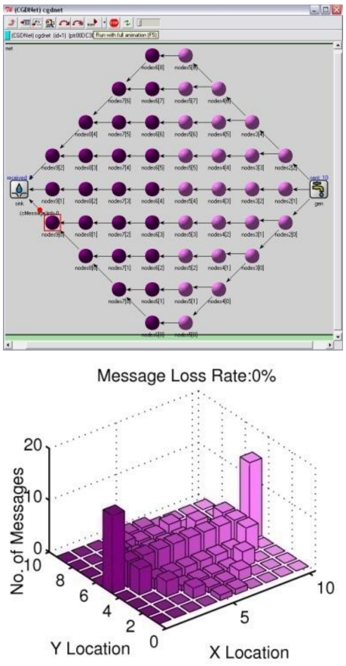


Figure 14. ISPT spreading and MLR with no nodes failed in symmetric hexagon contour with $x_{diff}=9$ and $y_{diff}=0$

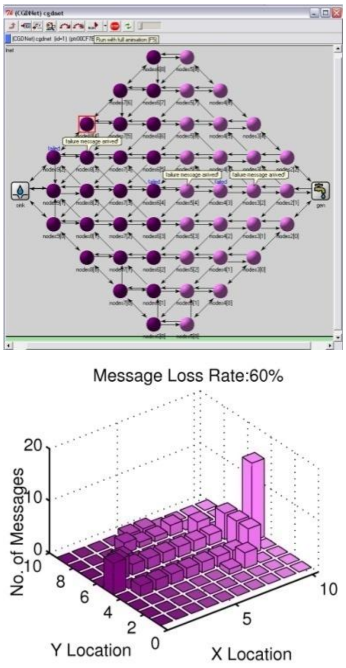


Figure 15. ISPT spreading and MLR with three failed nodes in symmetric hexagon contour with $x_{diff}=9$ and $y_{diff}=0$

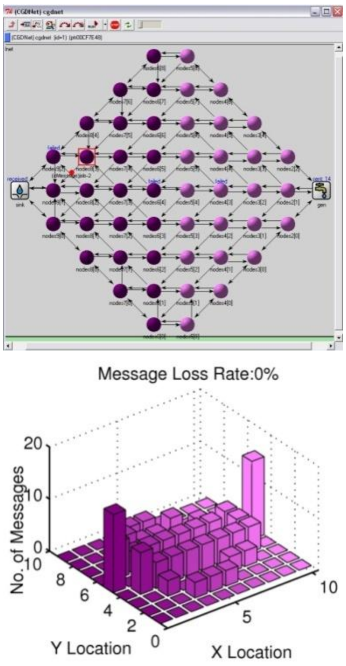
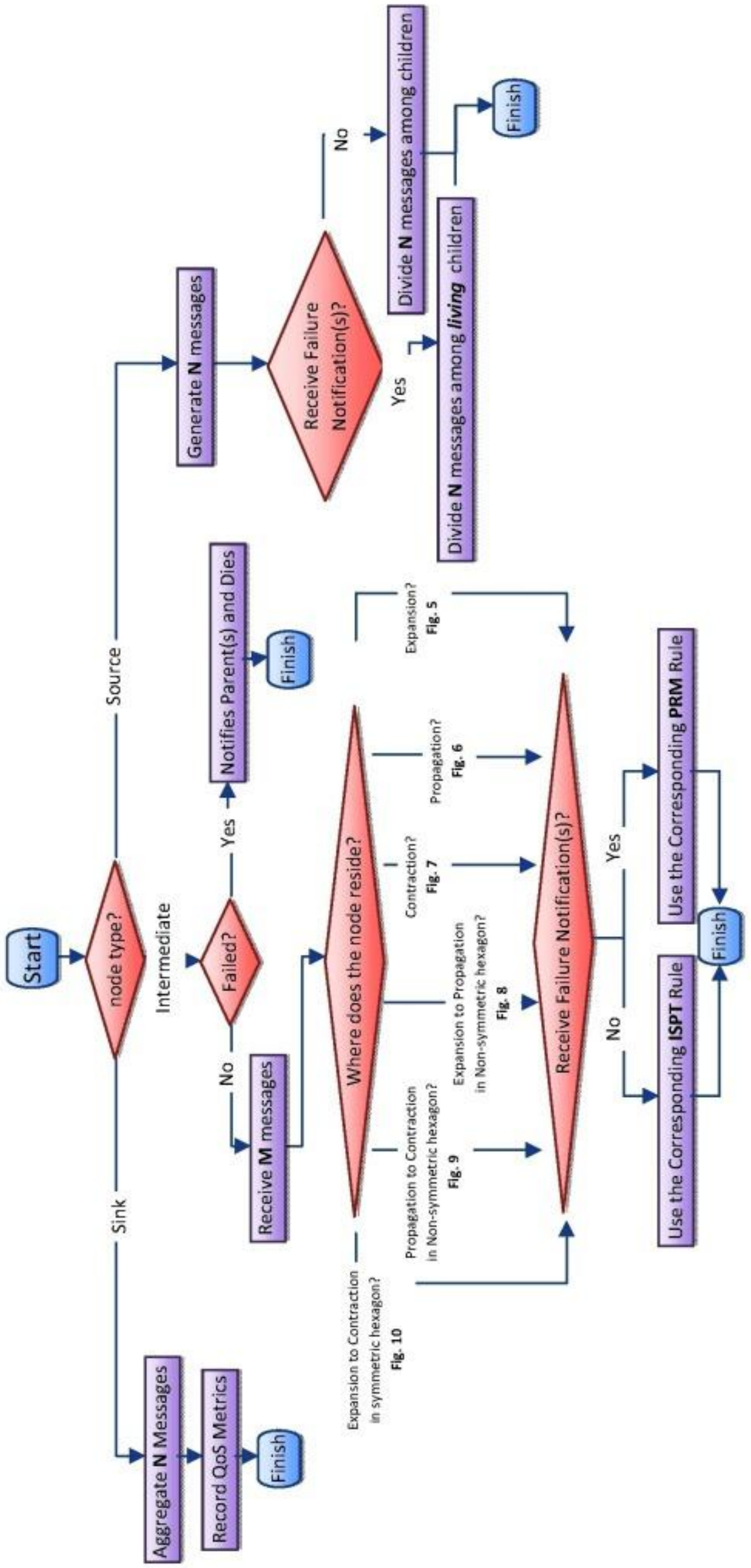


Figure 16. PRM spreading and MLR with three failed nodes in symmetric hexagon contour with $x_{diff}=9$ and $y_{diff}=0$

APPENDIX A



Flowchart A1. General Behavior of Nodes in PRM Scheme