



ISSN NO. 2320-5407

Journal homepage: <http://www.journalijar.com>

INTERNATIONAL JOURNAL
OF ADVANCED RESEARCH

RESEARCH ARTICLE

Robust Sink Failure Avoidance Protocol for Wireless Sensor Networks

¹ Abdul Razaque, ² Khaled Elleithy

University of Bridgeport, CT-06604, USA

Wireless and Mobile communication (WMC Laboratory)

Manuscript Info

Manuscript History:

Received: 26 July 2014

Final Accepted: 29 August 2014

Published Online: September 2014

General terms:

Algorithms, Design, Performance and theory.

Keywords:

WSNs, Sink failure avoidance (SFA), Error detection, Error recovery, Energy consumption.

*Corresponding Author

Abdul Razaque

Abstract

Sink failure is one of the critical issues in wireless sensor networks (WSNs). Prolonging the network lifetime of WSNs, the robust sink has a paramount significance to provide an interface between deployed sensors successfully. The sink failure highly affects the performance of several WSN applications. Hence, the Sink must have the capability to recover from the failure state immediately.

In this paper, we introduce a sink failure avoidance (SFA) protocol that improves the network lifetime by using sink fault tolerance algorithms. SFA helps determine the error detection and error recovery processes successfully. To demonstrate the effectiveness of algorithms, we use network simulator-2 (ns2.35). Based on the simulation results, it is validated that our algorithms highly support the error-detection and error-recovery process of the sink failure successfully.

Copy Right, IJAR, 2014. All rights reserved

Introduction

Wireless sensor networks typically involve the technology of small devices that comprise of tiny sensor nodes. Each sensor node does its job as a unit to monitor the critical data in the environment[1],[2],[3]. WSNs are self-configuring and self-healing networks comprising of homogenous and heterogeneous nodes. WSNs provide the promising solutions for numerous applications such as target detection, intrusion detection, industrial automation, airport surveillance systems, medical diagnosing systems, environmental monitoring, battlefield, etc. However, WSNs experience the challenge due to sink-failure for several applications.

The sink is of paramount importance to collect the data from different sensor nodes and forwards to the base station. The sink failure affects the network performance caused by lack of energy, limited storage capacity, computing power and security in the susceptible environment[4],[5]. As, these problems are not sufficiently addressed to avoid the sink failure in WSNs. Due to these restrictions, fault-affecting sensor nodes may lead to an unpredictable condition in which a region of interest is disconnected from the WSN. As a result, the forwarded data cannot be reached to sink node. A more complex situation in which a sink node fails is inappropriate selection of routing protocols for a specific application[6].

The sink is data-gathering node from all the nodes in the WSNs and is also responsible for transmitting to the end user. The sink node is selected to improve the data flows so that energy can be preserved[7]. Thus, generic two-tiered model was created to improve the network topology to augment the sink and application nodes[8]. The average bit-stream was used between sink and sensor nodes for prolonging the network lifetime. However, the energy efficiency was not considered [9]. Probability and missing probability (PMP) model was used to resolve the sink placement problem[10]. However, sink placement needs to be determined prior to selection of some points [11]. If sink functions correctly that achieves high probability of success for WSN applications.

On the other hand, the failed sink makes WSN as unusable. As a result, the quality of service provisioning is highly affected. Thus, balancing and improving the performance of WSNs, we introduce a sink failure avoidance protocol that improves the network lifetime by using sink fault tolerance algorithms that help determine the sink

failure and sink recovery process. This process is initiated by using backup sink (BS) to replace principal sink (PS) in case of failure. The research of art in this contribution depends on prolonging the network lifetime and improve the QoS provisioning. The remainder of the paper is organized as follows: The section I, presents the problem statement. Section II discusses the protocol that provides the sink with fault-tolerance capabilities. Section III gives the simulation setup and analysis of the result and section IV, finally concludes the paper.

I. PROBLEM STATEMENT

The wireless sensor networks comprise of tiny nodes with limited energy resources. They are scattered in the different regions of the network to collect the critical information from the physical environment[12]. These tiny nodes process the collected critical data to sink node in order to forward to end node. The WSNs experience the problem due to the sink failure. As a result, the sink failure causes the interruption and subsequently may compromise the network resources. Even if a sink node encompasses of additional resources as compared with other sensor nodes. However, it experiences the severe problem due to failures caused by the tough deployment phenomena.

The prerequisite of fault tolerance is inevitable with WSNs, particularly if the application running on WSN is of high significance. To make WSN functional, fault-tolerance capability fulfills its task despite faults[13]. There are two approaches are introduced to handle permanent and temporary sink failures, which are check pointing with active and passive replications. However, these both approaches fail to address the exact cause of sink failure and its recovery.

II. PROTOCOL PROVIDING SINK WITH FAULT TOLERANCE CAPABILITIES

The goal of this protocol is to make sink be highly robust for handling the error occurrences. In resulting, the fault-tolerance protocol prolongs the network lifetime by rendering the sink as fault-tolerant. The protocol follows three basic rules, which include fast failure detection, maintaining the reliable sink state, and finally resuming the sporadic task. We represent the sink (base station) by ' S ' that creates the interface between WSN and end users. The sink involves two types: principal sink ' P_s ' and backup sink ' B_s '. The ' B_s ' creates the relay for ' P_s '. Therefore, ' B_s ' functions rather than ' P_s ' and collects the critical information from sensing nodes and forwards to the end nodes depicted in Figure 1. The ' P_s ' is invoked to distribute the interested queries ' $\nabla\varepsilon$ ' periodically on the request of users during each unit waiting time ' Δt_1 '. The initial behavior of ' P_s ' is defined in Algorithm-1.

Algorithm 1: Initial behavior of Principal Sink

1. Initialization of variables (Δt_1 : waiting time for principal sink, ' \mathbb{R} ': sensed data table, $\nabla\varepsilon$: interested queries, and ' S_{Nj} ': sensor node j)
 2. P_s sending initialization request
 3. Repeat until all nodes are initialized
 4. Initializing (Δt_1 and \mathbb{R})
 5. if {Reception ($\nabla\varepsilon, P_s$)} || Timeout Δt_1 then
 6. Broadcasting ($\nabla\varepsilon$)
 7. endif
 8. If (Reception ($\nabla\varepsilon, S_{N1}$)) then
 9. Add ($\nabla\varepsilon, \mathbb{R}$)
 10. Compute (\mathbb{R}) //Aggregating and computing the data
 11. Send (ϖ, P_s) until (End of $\nabla\varepsilon$)
 12. endif
-

Let us assume that ' P_s ' may fail due to either by unexpected physical interruptions or death of energy. Let ' E_{min} ' be a minimum energy required for ' P_s ' to work properly. Once an error is detected at ' P_s ', then responsibilities of ' P_s ' are shifted to ' B_s '.

The protocol uses checkpoint process that helps determine the status of the principal node. Once principal node ' P_s ' saves its state and determines the checkpoint (level of energy), then it forwards to B_s '. Thus, this process occurs in the following three steps.

- It occurs after computing and data aggregation phase.
- It occurs once B_s inquires the check point request.

- After expiration of set-timer Δt_3

When B_s obtains the checkpoint ' $\gamma\Delta$ ' of P_s , then it stores in the checkpoint table ' $\gamma\Delta_t$ '. The behavior of principal sink and backup sink is elaborated in algorithm 2.

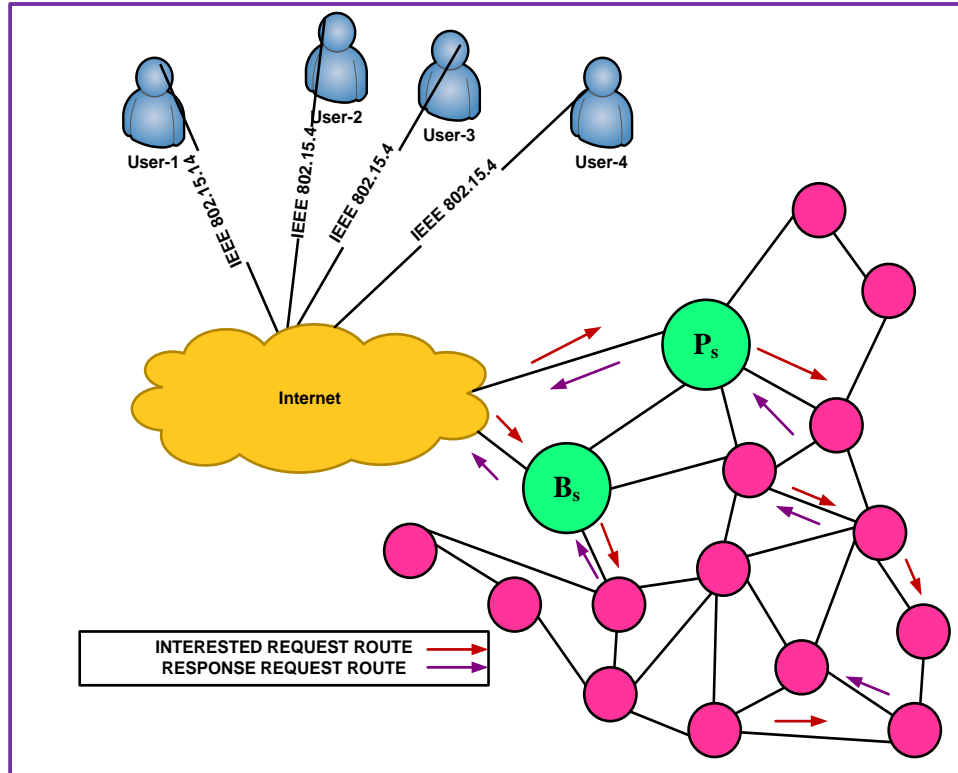


Figure 1. Fault tolerance robust architecture for a wireless sensor networks

Algorithm 2: New behavior of principal sink

1. Initialization of variables (Δt_1 : waiting time of principal sink, Δt_2 : waiting time of Backup sink to take responsibility ' \mathbb{R} ': sensed data table, $\nabla\epsilon$: interested queries, and ' S_{NK} ': sensor node k)
2. WSN sending initialization request
3. Initializing ($(\Delta t_1, \Delta t_2, \mathbb{R})$)
4. Repeat until all nodes are initialized
5. if {Reception ($\nabla\epsilon, P_s$) || Timeout Δt_1 then
6. Broadcasting ($\nabla\epsilon$)
7. endif
8. If (Reception ($\nabla\epsilon, S_{NK}$)) then
9. Adding ($\nabla\epsilon, \mathbb{R}$)
10. processing (\mathbb{R}) // aggregating and calculating the data
11. endif
12. Sending (ϖ : computed result, P_s)
13. Set ($\gamma\Delta = \gamma\Delta_s$)
14. Sending (M, B_s) // Measuring E_{min} of ' B_s ' to function instead of ' P_s '
15. Setting (E_{min}, B_s)
16. if {Reception ($\nabla\epsilon, B_s$) || Timeout Δt_2 then
17. set ($\gamma\Delta' = \dot{K}$)
18. Sending (\dot{M}, B_s) // Measuring E_{min} of ' B_s ' to function instead of ' P_s '
19. Setting (E_{min}, B_s)

20. endif
21. If Receiving message ("Hello", P_s) then
22. Responding message ("I am still alive" B_s)
23. Measuring (E_{min}, B_s)
24. Sending (E_{min}, B_s)
25. **Until** (End of $\forall \epsilon$)
26. endif

Furthermore, the protocol applies the sink error detection process that consists of the following steps.

- The backup sink node checks each Δt_3 time units periodically and also sends a "hello" messages to principal node to determine the status. Once B_s does not receive a response after fixed N number of attempts, and then P_s is declared as failure sink.
- Once backup sink inquires the minimum energy level of principal sink node, and if it finds an energy level of principal sink node is less than set threshold value ' δ ' of minimum energy. And then P_s is declared as failure sink node. This statement can be written as:
If $E_{min} \leq \delta$, thus P_s is failure ' \mathcal{R} '.
- Once, principal sink node is declared as a failure, then message is broadcasted in the entire network in order to restrict the nodes to stop sending messages to failure P_s .

If error is detected in principal sink node, then error recovery process is initiated. As a result, B_s starts to store the last state ' $\gamma \Delta_s$ ' of P_s to avoid the energy consumption in order to improve QoS provisioning explained in algorithm 3. Furthermore, Table 1 demonstrates the used parameters in the system.

TABLE 1: Showing used parameters with definition in the system model

PARAMETERS	SYSTEM DEFINITIONS
B_s	Backup sink node that replaces the principal sink node in case of failure of principal sink node
$\forall P_s$	MAC address of principal sink node
E_{min}	Minimum energy
ℓ	Initializing request
P_s	Principal sink node
\mathcal{R}	Sink failure
\mathbb{R}	Sensed data table
δ	Threshold value that is set to measure the minimum energy level of sink node
S	Sink (base station)
S_{Nj}	sensor node j in WSN
S_{Nk}	Sensor node k in WSN
$\Delta t_1, \Delta t_2, \Delta t_3$ and Δt_4	Different waiting times for principal node to detect its failure state
$\forall \epsilon$	Interested queries
ϖ	Computed result
$\gamma \Delta$	checkpoint
$\gamma \Delta_s$	Stored checkpoint is used when detecting the error state of the principal node

$\gamma\Delta_t$ checkpoint table is used when principal sink node gets failure; then backup sink node stores the current status of principal sink node in this table

Algorithm 3: Replacement of Principal Sink node P_s with Backup sink node B_s

1. Initialization of variables (Δt_3 : waiting time for principal sink, Δt_4 : waiting time for principal sink)
 2. Initializing the waiting time and checkpoint table ($\Delta t_3, \Delta t_4, M$)
 3. If receiving (l, PS) then
 4. Sending ($\forall P_s, P_s$)
 5. Repeat until message is delivered
 6. endif
 7. If Δt_4 occurs, then send ($\gamma\Delta, P_s$)
 8. endif
 9. If (Δt_3 occurs, then send a message ("Hello", P_s)
 10. endif
 11. If receiving ($\gamma\Delta, P_s$), then store ($\gamma\Delta, \gamma\Delta_t$)
 12. Until the end of the request
 13. endif
 14. If not receiving ("I am alive", P_s) $\parallel (\gamma\Delta, P_s) \parallel E_{min} \leq \delta$; then, error detection process starts
 15. Broadcasting the message P_s is failure ' R_s '
 16. Storing $\gamma\Delta_s(\gamma\Delta_t)$
 17. Installing $\gamma\Delta_s$ and Broadcasting ("I am P_s ")
 18. endif
-

III. SIMULATIONS AND RESULTS

We have simulated the performance of sink failure avoidance protocol using ns-2.35-RC7[14] with Ubuntu 13.10 operating system. We have created realistic scenario that reflects the real WSNs phenomenon. The nodes are randomly placed with uniform fashion in the area of 500 X 500 square meters. The initial energy of each sensor node is set 8 *joules*. The bandwidth of the node is 250 kb/sec, and maximum power consumption for each sensor node is set 13.6 *mW*. In addition, the sensing and idle modes have 12.4 *mW* and 0.45 *mW* respectively. Each sensor node has the capability to broadcast data from -16 *dBm* to 11 *dBm* power intensity.

The total simulation time is set to 35 minutes, and the pause time is set to 5 seconds for the initialization of phase to warm the nodes at the start of the simulation. The obtained results demonstrate an average of 12 simulation runs. The energy consumption pertaining to different radio modes and simulation parameters is summed up in Table 2.

Table 2. Summarized simulation parameters

Name of parameters	Description
Transmission Range	30 meters
Sensors	BT node sensors
Sensing Range of node	15 meters
Initial energy of the node	8 Joules
Bandwidth of node	250 Kb/Sec
Number of sensors	350
Number of sinks	2
Network size	500 X 500 m ²
Packet transmission rate	25 Packets/Sec
Data Packet size	256 bytes
Simulation time	35 minutes
Initial pause time	5 Seconds
Transmitter energy	13.6 mW
Receiver energy	12 mW,
Power intensity	-16 <i>dBm</i> to 11 <i>dBm</i>
Location of principal sink	(0, 550)
Location of backup sink	(0, 530)

The selected criteria for our protocol involve following metrics.

- Initialization time
- Energy consumption
- Recovery time

A. Initialization Time

We determine the initialization time for sensor node, principal sink node and backup sink node depicted in Figure 2 and 3. In Figure 2, we use our sink failure avoidance protocol to detect the consumed time for initialization of different number of sensor nodes. We have observed that when network size increases then initialization time increases that prove the direct trade-off between initialization time and network size. Based on the result, we also noticed very important point that the consumed time for first 50 nodes is slightly different from last 50 nodes.

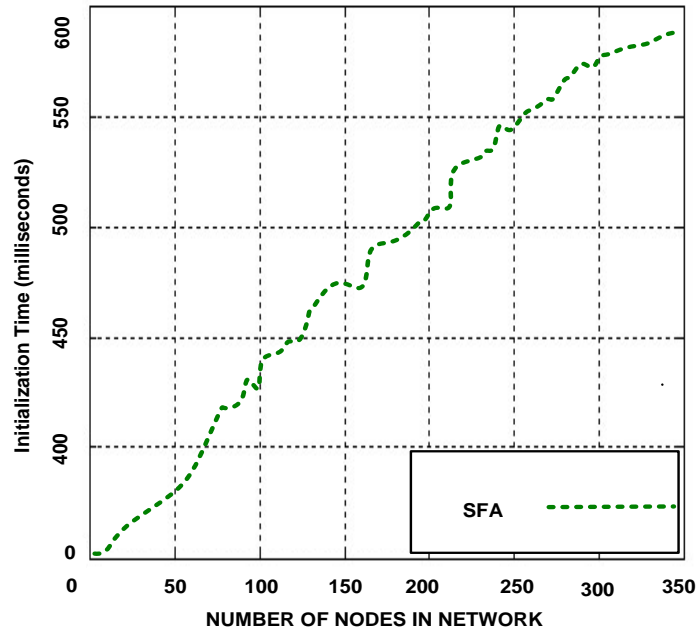


Figure 2. Initialization time for different number of sensor nodes

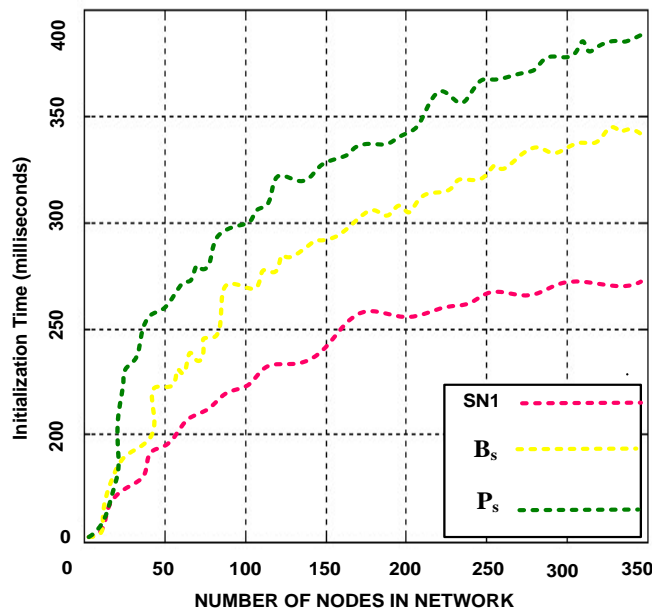


Figure 3. Initialization time for Principal sink node, backup sink node and sensor node at different topologies

The reason is this, the nodes have more energy for initializing the first 50 sensor nodes, but nodes run out their energy for initializing the last 50 nodes.

However, time is not changed for each simulation runs. In Figure 3, we have calculated the initialization time for sensor node, principal sink node and backup sink node at the different size of the network. We have observed that network size increases, then initialization time for sensor node, principal sink node and backup sink node increases in a different way. The principal sink node takes slightly higher initialization time as compared with backup sink node and sensor node. The reason of the increase in initialization time of principal sink node is its additional functionalities.

B. Energy Consumption

The sensor nodes in WSN are powered by limited battery resources. Thus, they require using the limited energy budget. We have computed consumed energy for different components of sensor node e.g. CUP, LED, EEPROM and Radio transceiver depicted from Figure 4 to 7. Each component of the sensor node consumes different energy in network. We have observed that in Figure 4 and 5, principal sink node consumes more energy followed by backup sink node, sensor node j and sensor node k for CPU and Radio transceiver components.

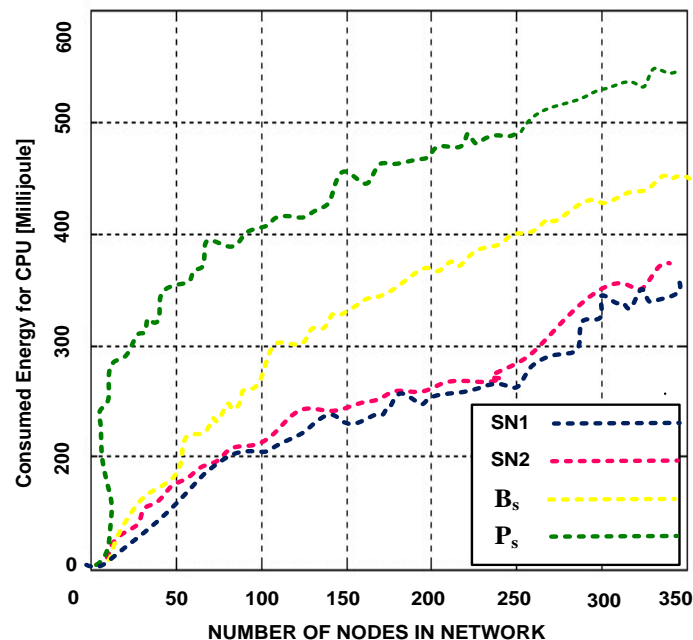


Figure 4: Energy consumed for CPU on different sinks

The reason for the consumption of additional energy in case of principal sink node is to transmit additional data and control packets. Whereas, backup sink node has responsibility to keep on monitoring the principal sink node on the regular basis for replacement of principal sink node in case of its failure occurs that causes the energy consumption.

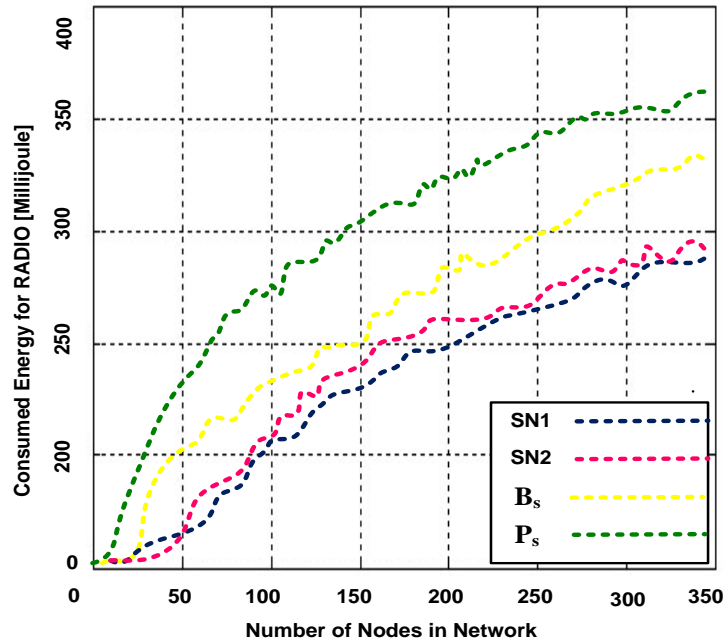


Figure 5: Energy consumed for RADIO on different sinks

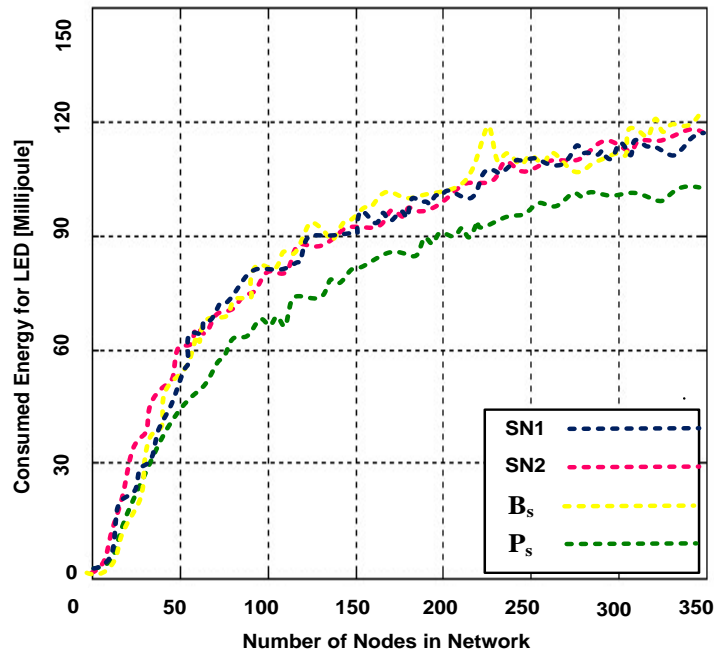


Figure 6: Energy consumed for LED on different sinks

Figure 6 demonstrates the consumed energy for light emitting diode (LED) for principal sink node, backup sink node, sensor node j , and sensor node k . In LED, principal sink node consumes less energy as compared with other nodes. We have observed from Figure 4 to 6 that more energy is consumed for Radio transceiver and CUP. In Figure 7, consumed energy for electrically erasable programmable read-only memory (EEPROM) is measured only for backup sink node because principal sink node and other nodes do not require EEPROM. The reason of consuming the energy for EEPROM is to monitor the principal sink node periodically.

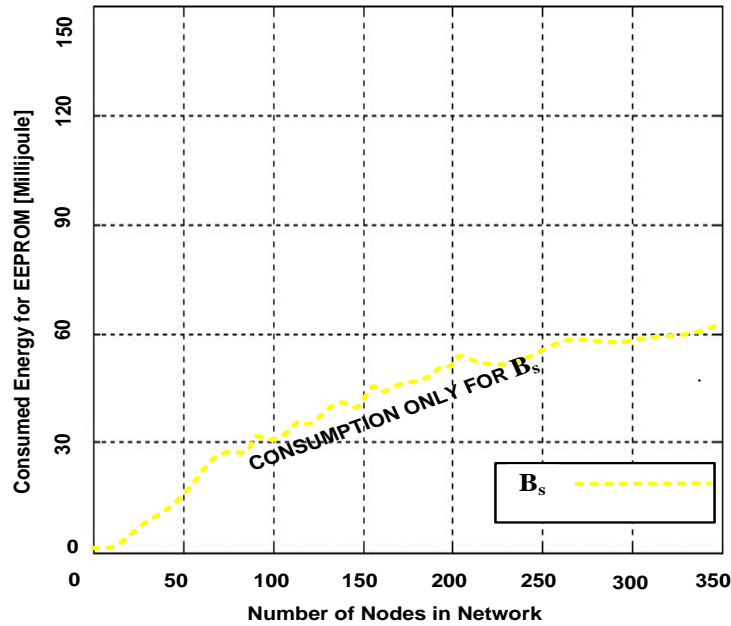


Figure 7: Energy consumed for EEPROM on different sinks

In Figure 8, we show the total energy consumed for all components of principal sink node, backup sink node, sensor node j , and sensor node k . Based on the outcomes, we have observed that principal sink node consumes more energy that is around 992.4 millijoules for CPU, but in our case, Radio does not consumes less energy in our case.

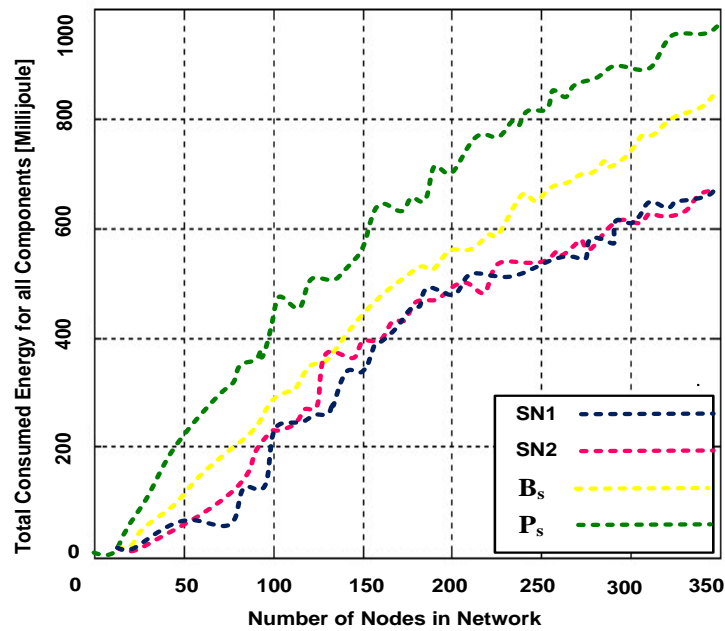


Figure 8: Total energy consumed for different components of the sensor

C. Recovery Time

One of the key metrics is to determine the recovery time, when principal sink node got failure and replaced by backup sink node. Therefore, simulation time and consumed energy are two significant constraints. We observe in Figure 9, the recovery time that is the trade-off between the failure and recovery occurrences. We have observed that principal sink node gets failure at 13.2 minutes, then backup sink node replaces the principal sink node for improving the quality of service provisioning. In the case of replacement, much recovery is reduced. We further

observe that the curve for principal sink node is balanced, and no information is retrieved because of the injection of failure. The consumed energy for backup sensor node remains stabilized when it starts relaying rather than principal sink node.

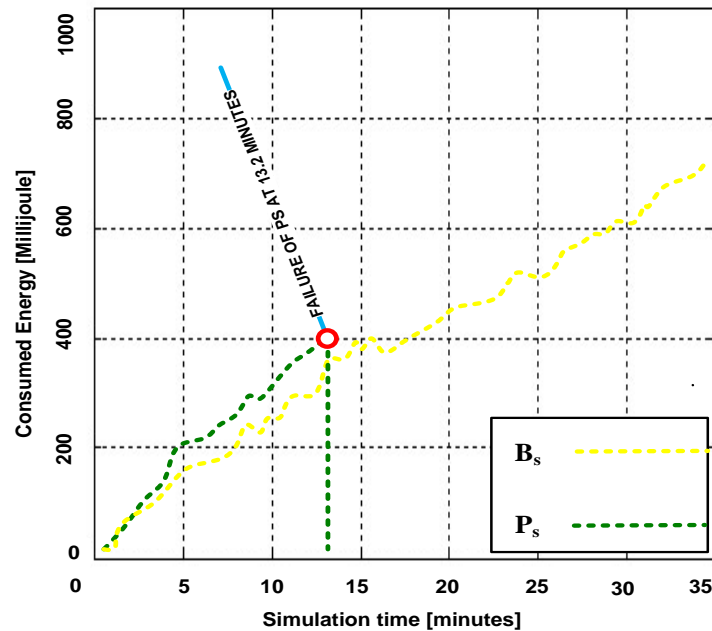


Figure 9. Failure of the principal node and consumed energy using the different period

IV. CONCLUSION

In this paper, we have introduced sink failure avoidance protocol for the detection and recovery of the sink errors. This contribution aims to determine the normal activity and energy consumption level of sink nodes using SFA. The SFA protocol provides the robust fault tolerance functionality to improve the quality of service provisioning by replacing failure primary sink node with backup sink node. To demonstrate the validity of SFA protocol, we have used network simulator-2.

Based on the simulation results, we observed that CPU consumed more energy as compared with other components of sensor nodes that are very interesting discovery. Furthermore, another significant discovery is to determine the EEPROM that is only used for backup sink node because it only monitors the failure of principal sink node. We also validated that SFA also reduces the recovery time and energy consumption.

REFERENCES

- [1] A. Razaque and K. M. Elleithy, "Low Duty Cycle, Energy-Efficient and Mobility-Based Border Node—MAC Hybrid Protocol for Wireless Sensor Networks," *Journal of Signal Processing Systems*, pp. 1-20, 2014, 10.1007/s11265-014-0947-3.
- [2] A. Razaque and K. M. Elleithy, "Energy-Efficient Border Node Medium Access Control Protocol for Wireless Sensor Networks," *Sensors*, vol. 14, pp. 5074-5117, 2014, doi: 10.3390/s140305074.
- [3] A. Razaque and K. Elleithy, "Automatic energy saving (AES) model to boost ubiquitous wireless sensor networks (WSNs)," *arXiv preprint arXiv:1309.4504*, 2013.
- [4] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "A discrete-time battery model for high-level power estimation," in *Proceedings of the conference on Design, automation and test in Europe*, 2000, pp. 35-41.
- [5] A. Razaque and K. Elleithy, "Efficient Search (RES) for One-Hop Destination over Wireless Sensor Network," *arXiv preprint arXiv:1310.1129*, 2013.
- [6] R. Sumathi and M. Srinivas, "A Survey of QoS Based Routing Protocols for Wireless Sensor Networks," *JIPS*, vol. 8, pp. 589-602, 2012.

- [7] A. Bogdanov, E. Maneva, and S. Riesenfeld, "Power-aware base station positioning for sensor networks," in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004.
- [8] J. Pan, L. Cai, Y. T. Hou, Y. Shi, and S. X. Shen, "Optimal base-station locations in two-tiered wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 4, pp. 458-473, 2005.
- [9] J. Li and P. Mohapatra, "Analytical modeling and mitigation techniques for the energy hole problem in sensor networks," *Pervasive and Mobile Computing*, vol. 3, pp. 233-254, 2007.
- [10] W. Wang, V. Srinivasan, K.-C. Chua, and B. Wang, "Energy-efficient coverage for target detection in wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, 2007, pp. 313-322.
- [11] D. Luo, H. Zhao, Y. Bi, K. Lin, X. Zhang, Z. Yin, and P. Sun, "Multiple sink nodes' deployment based on PMP in WSNs," in *Pervasive computing and applications*, 2007. ICPCA 2007. 2nd international conference on, 2007, pp. 675-678.
- [12] A. Razaque and M. Elleithy, "Least Distance Smart Neighboring Search (LDSNS) over Wireless Sensor Network," in *Proceedings of the IEEE Conference on European Modeling Symposium (EMS)*, Manchester, UK, 2013, pp. 20-22, doi: 10.1109/EMS.2013.91
- [13] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, pp. 171-190, 2007.
- [14] T. Issariyakul and E. Hossain, *Introduction to network simulator NS2*: Springer, 2011.

BIOGRAPHY



Abdul Razaque is Editor-in-Chief for International Journal for Engineering and Technology (IJET), Singapore and also associated with Computer Science and Engineering Department, University of Bridgeport, USA. He holds fellowship form Higher Education Commission (HEC) Pakistan, and Common Wealth, UK. He served as Head of computer science department in Model colleges setup Islamabad, Pakistan from 2002 to 2009. He also led the projects as project Director for promoting the trend of information technology (IT) in Pakistan funded by United Nation organization (UNO) and World Bank during 2005 to 2008. He is currently active researcher of wireless and Mobile communication (WMC) laboratory, UB, USA. Abdul Razaque has also been working as Chair, Strategic Planning Committee for IEEE SAC Region-1. USA and Relational Officer for IEEE SAC Region-1 for Europe, Africa and Middle-East. Abdul Razaque has chaired more than dozen of highly reputed international conferences and also delivered his lectures as Keynote Speaker. His research interests include the wireless sensor networks, design and development of learning environments, TCP/IP protocols, multimedia applications and ambient intelligence.



Dr. Elleithy is the Associate Vice President of Graduate Studies and Research at the University of Bridgeport. He is a professor of Computer Science and Engineering. He has research interests are in the areas of wireless sensor networks, mobile communications, network security, quantum computing, and formal approaches for design and verification. He has published more than three hundred research papers in international journals and conferences in his areas of expertise. Dr. Elleithy has more than 25 years of teaching experience. His teaching evaluations are distinguished in all the universities he joined. He supervised hundreds of senior projects, MS theses and Ph.D. dissertations. He supervised several Ph.D. students. He developed and introduced many new undergraduate/graduate courses. He also developed new teaching / research laboratories in his area of expertise.

Dr. Elleithy is the editor or co-editor for 12 books by Springer. He is a member of technical program committees of many international conferences as recognition of his research qualifications. He served as a guest editor for several International Journals. He was the chairman for the International Conference on Industrial Electronics, Technology & Automation, IETA 2001, 19-21 December 2001, Cairo – Egypt. Also, he is the General Chair of the 2005-2013 International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering virtual conferences.