PERFORMANCE EVALUATION OF LEAST DISRUPTIVE TOPOLOGY REPAIR ALGORITHM (LEDIR) USING NS-2

Niharika Gorre, Prudhvi Raj Kyatham and Mohamed El-Sharkawy

Purdue School of Engineering and Technology, Indianapolis, Indiana, USA

ABSTRACT

In wireless sensor networks, sensor nodes fail either when some critical event occurs at the node or when the battery of the nodes is completely drained. If the remaining nodes in the sensor network are not aware of the node failure, the network might under undergo significant broadcast delay and path loss. An alarm packet has to be broadcast throughout the network when any critical event occurs and the transmission path must be recovered to achieve better quality of service in the sensor network. This paper utilizes least disruptive topology repair (LeDir) algorithm to minimize the power consumption and to handle such critical events. LeDir algorithm is used to recover the transmission path and to ensure that the quality of service issues in the network is met. The variations in the throughput, roundtrip time, broadcast delays and packet delivery ratio of the network are observed by performing multiple NS-2 simulations. It is observed that the throughput of the network could be quickly regained when the transmission path is recovered by node replacement.

KEYWORDS

LeDir, NS-2, AODV.

1. INTRODUCTION

Wireless sensor networks contain a huge number of low power sensor nodes that are placed heterogeneously in the deployable area. Each node acts a transceiver and has processing capabilities. They detect sudden changes in the environment such as temperature, pressure, humidity and report the information to the end users immediately. They are employed in applications such as environmental monitoring, gas monitoring in coal mines, in military applications to broadcast any urgent information to the end users. They prevent further damage to the area of interest by giving status signals to the end users. Wireless sensor networks are limited by the number of nodes used and their battery power, energy and computational capacity.

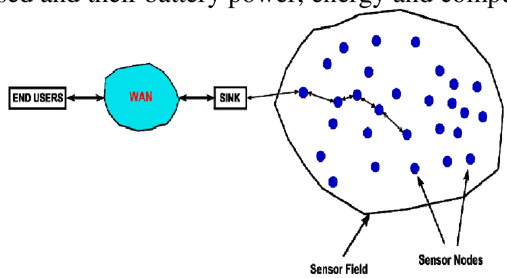


Figure 1. Wireless Sensor Network.

DOI : 10.5121/ijwmn.2014.6406

The source node sends the information to the destination node by choosing the shortest path based on the protocols and routing tables used in the network. The difference between wireless sensor networks and ad-hoc networks is that the former uses point to point communication whereas the latter uses broadcast communication. When a node in the network fails, all the other nodes in the network need to be aware of the node failure. The central node is made to broadcast an alarm packet that contains information about the node failure to the remaining nodes in the network [1, 2].

A single node failure might cause the entire network to become disjoint. As a solution to this problem we reposition the failure node and replace it with a working node already present in the network. Especially in applications like search and rescue operations, the timely coordination among the nodes is required. In this paper, network restoration algorithms-DARA (Distributed Actor Recovery Algorithm) and Least Disruptive topology repair algorithm (LeDir) are studied. LeDir maintains constant path even after the node failure. LeDir is implemented in NS2 simulator and various parameters are calculated. The paper considers the problem of the connectivity restoration [3, 5]. Throughput, End-to-End delay of the Packets, packet RTT and Packet Delivery Ratio are calculated from the trace files generated during the simulation. Data trace file contains information about the network. GNUPLOT tool is used to generate graphs and analyze the results.

1.1. System Model

The used system contains a central node and groups of other sensor nodes. The central node is placed such that it is the transmission range of all the other nodes in the network. This acts as the main hub for the all the other nodes in the network that is all the data transmission happen through this node. The remaining nodes are grouped together in the form of blocks such that the center node is positioned in the middle of the node blocks (node 9) as shown in Figure 2. The central node maintains a registry that has information about the nodes participating in the network such as routing tables, battery levels, etc.

Figure 2 shows the model of the used network. Node 9 is the central node and all the remaining nodes are arranged in the form of blocks. Nodes 0,1,2,3 form the first block, nodes 4,5,6,7,8 form the second block, nodes 10, 11,12,13,14 form the third block. We consider two data transmissions in the network scenario. During the first transmission node 0 sends data packet to node 7 with node 3 and node 9 being the intermediate nodes. Node 8 sends the data packet to node 12 with node 9 and node 11 as the intermediate nodes. The data transmission occurs successfully until node failure occurs at node 3.

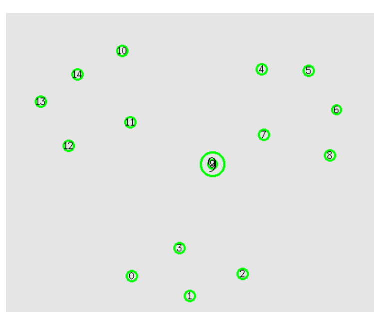


Figure 2. Network Model.

Figure 3. Network Example.

1.2. Distributed Actor Recovery Algorithm (DARA):

When a node fails, the best node used for replacement is selected from its one hop neighbors. The node having lowest degree and least distant from the failure node is selected. Any child nodes disconnected during this process are recovered by recursive relocation procedure. Thus shortest path between the nodes is not maintained [11].

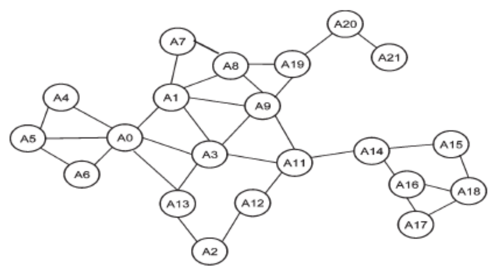


Figure 4. Network after the Implementation of DARA.

Let us assume that node A10 fails. Nearest neighbor with the lowest degree is selected. Thus A11 moves to the location of A10. The process is repeated by moving A12 to the position of A11, A2 to the location where A12 was earlier. Finally, A2 is replaced with A13. The final network after the implementation of DARA is shown in Figure 4. The shortest path length is increased by one-hop after implementing DARA. A0 and A3 that were one-hop neighbors of A2 earlier are moved far with the involvement of node A13. This algorithm may not be applicable for applications sensitive to delay. LeDir algorithm is a path restoration algorithm that is used to sustain the path lengths before the failure.

1.3. Least-Disruptive Topology Repair (LeDir) Algorithm:

LeDir involves cascaded movement only for the lead node. This algorithm helps the restoration of connection without extending the length of the shortest path among nodes. The LeDir algorithm allows the movement of nodes in blocks in contrast to other path restoration algorithms in which the nodes move in a serial fashion.

The algorithm performs the following steps:

- **Detecting the node failure:** The sensor nodes transmit messages to their one-hop distant neighbors periodically. When a node does not transmit the message it is understood it node has failed. The one-hop neighbors of the failed node find if the failed node is critical to the network. A node is considered to be important if it is on the cut-vertex. Let us assume that the node A10 has failed. It is important to recover as it is on the cut-vertex.

- **Identifying the smallest block:** The smallest block is the block with the least number of nodes. The reachable set of nodes for each direct neighbor of the failed node is identified and the set containing less number of nodes is selected. Thus the recovery overhead is reduced. Here, the block containing nodes 14, 15,16,17,18 is the smallest block.
- **Replacing the failure node:** The faulty node is replaced with the one-hop neighbor of the smallest block. The nearest neighbor of node 10 that belongs to the smallest block is node 14. Thus node 14 moves to the position of node 10.
- **Movement of child nodes:** The nodes that are at two hops distance from the faulty node are called children and those at a distance of three hops are called grandchildren. Once the child knows that its parent is moving to a new location, it tells its one hop neighbors that is grandchildren and follows the path of its parent. If a child has two parent nodes that are moving, it relocates to a position to maintain its connection with both the parent nodes.

In the example, node 14 informs its neighbor's node 15 and node 16 and moves to the position of node 10. The children of node 14 that is node 15 and node16 follow the path of node 14 to maintain the transmission link with node 14. This is highly advantageous as the routing tables of the children need not be updated. Before following the path of node 14, nodes 15 and 16 inform their children node 17 and node 18 about their movement. Since node 18 has three parent nodes, node 15, node 16 and node 17, it moves to a new location such that it is connected to these three nodes. Thus the node 17 does not relocate. Figure 5 shows how the LeDir algorithm is implemented [7].

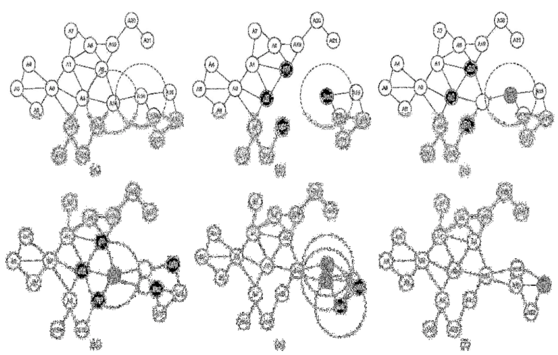


Figure 5. Proposed Algorithm.

Table 1. Comparison of DARA and LeDir [3].

Property	DARA	LeDir
Maximum no. of nodes required	N-3	0.5(N-1)
Maximum no. of messages to be sent	5N-3	1.5(N-1)
Maximum distance travelled by the node	r	r
Maximum distance travelled by other nodes engaged in the path recovery	rN	0.5(rN)

From table 1, DARA involves N-3 nodes whereas LeDir involves N-1/2 nodes for path restoration, where N is the number of nodes. In our network model, 15 sensor nodes are used. DARA requires 12 nodes whereas LeDir needs 7 nodes for path recovery. LeDir needs to send 21 messages and DARA needs to send 72 messages. The less number of messages provides less overhead. Thus, LeDir performs better than DARA. While the distance travelled by the peak node is the same in both the algorithms, the maximum distance travelled by the children is more in DARA i.e. in our system the child nodes need to travel a distance of 15r for DARA and 7.5r for LeDir, where r is the transmission range. Thus, LeDir if found to outperform in many ways and tries to recover the network with less recovery overhead while maintaining the shortest path lengths. This paper implements LeDir in NS-2 for the system model considered. In the used network model, it is assumed that the node 3 is the faulty node. The one hop neighbors of node 3, node 0, node2 and node 9, identify the failure when node 3 stops sending periodical status messages. LeDir algorithm identifies the smallest blocks as blocks containing node 0 and node 2. Node 2 is assumed to be least distant from node 3. Thus node 2 relocated to the position of node 3 and node 3 is moved to the previous location of node 2. Thus, the data transmission path is recovered and the network starts functioning normally. The quality of service parameters could be met.

2. IMPLEMENTATION

2.1. NS-2 Simulator:

NS-2 version 2.35 is used to simulate the network. NS-2 is an object oriented, discrete event simulator. NS-2 supports the use of TCP protocol is primarily useful for simulating wireless networks. NS-2 can be programmed using the Tool Command Language (TCL) and uses C++ in the backend of the system for interpretation. The TCL script is used to invoke the event scheduler, setup the network topology, set the start and stop of data transmission, call functions in the inbuilt NS-2 library [8, 14].

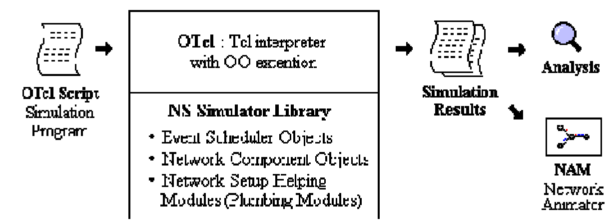


Figure 6. Network simulator.

When the TCL file is run in NS-2 the simulation results could be either in the form of a trace file or a Network Animator (NAM). The results are analyzed to estimate the performance of the communication network in the real world. For the simulator, a wireless sensor network containing 15 sensor nodes is considered. They are assumed to have full energy (100%) at the beginning of the simulation. Nodes are arranged in the form of blocks with each block having nodes placed at different levels. In the topology, node 9 acts as the central node for the all the other blocks. All the transmission occurs through the central node.

Software requirements:

Programming language: TCL, C++, PERL, GAWK
Simulator: NS-2 2.35
User interface: NAM, Text Editor
Operating System: Ubuntu 12.00

Hardware Requirements:

Minimum RAM requirement: Min 1 GB
Minimum Hard Disk space needed: 20GB
Minimum Processor Speed: 1.8 GHz

Other parameters used in the Network:

Channel type: Channel/Wireless Channel
Radio-propagation model: Two-way Ground
Network Interface type: Physical/Wireless Interface;
MAC type : Mac/802.11
Interface queue type: Drop Tail/Priority Queue
Link layer type: LL
Antenna model: Omnidirectional Antenna
Queue length: 50
Number of nodes: 15
Routing Protocol: AODV
Topography: 3000 X 3000
Initial Energy: 100 Joules
Transmission Protocol: TCP/New Reno
Application: CBR (Cluster based routing) application

AODV protocol is used for the routing the packets and is used to find the shortest path in shorter duration of time [12]. Transmission protocol (TCP) along with new Reno Algorithm is used. TCP helps in the avoidance of congestion in the network. The new Reno algorithm uses congestion window techniques to control the congestion in the network by reducing the flow of data packets. The application is set to have constant bitrate to maintain constant speed of flow of data.

The network has two TCP connections- node 0 to node 7 and node 8 to node 12. Node 3 is made to fail after certain period of time. LeDir algorithm finds the best node to replace the faulty node and the connection is recovered. We used the agent trace information is used to calculate the throughput and mac trace information is used to calculate packet latency. Agent trace information is obtained while performing simulations for the packet delivery ratio and end-to-end delay of the network. Particle extraction report language (Perl) is used for calculating the throughput and GAWK is used for the remaining calculations. The graphs are plotted using the GNUPLOT tool. It is a 3-D graph generating tool used for the research purposes.

2.2. Nam:

Nam is an animation tool used to view the trace file data produced during the network simulations in the form of animations. Layout of the topology, data transmissions and other changes occurring in the network with time are shown graphically. Nam contains various controls such as play, pause, fast forward etc. which are useful to observe the simulations occurring at the required instant of time. The step size of time could be varied to change the speed of simulation [11].

Nodes are represented using circles. Different colors are used for nodes to identify the nodes belonging to a particular block. The inner circles represent the energy level of a particular block. They are differentiated as-green color for 100% energy, yellow for less energy, red for no energy. The central node contains information about all the nodes participating in the sensor network. All the nodes update their status periodically to the central node.

The data transmission begins at 5 seconds. The network is made to perform two data transmissions-one from node 0 to node 7 with node 3 and node 9 being the intermediate nodes, the other from node 8 to node 12 with node 7 and node 11 being the intermediate nodes. When the data packet reaches the destination, it sends an acknowledgement about the packet reception to the source. If any node in the path fails, the transfer of packets comes to a halt due to disconnection in the network. There is no transmission of data between node 0 and node 7 until the path is recovered.

The LeDir algorithm identifies node 2 as the nearest node in the smallest block. Node 2 is moved to the location of node 3 to take the responsibilities of node 3. This is illustrated in Figure 7. The data flows from node 0 to node2 and then to node 7 through the central node. The functionality of the sensor network is regained and hence the quality of service parameters of the network could be achieved by using the LeDir algorithm. As the data transmission take place continuously, the battery levels of the nodes participating in the data flow keeps decreasing from time to time. The decrease in battery level is indicated by inner yellow circles. Node 9 gets exhausted first since it participates in both the transmissions.

Figure 7. Network Animation (Nam).

3. RESULTS AND DISCUSSION

3.1. Throughput:

Network throughput is the rate of successful delivery of data packets over a communication channel. The data might be transmitted either using a physical channel or a wireless channel. It is measured in bits per second (bps) or data packets per second. It is the most important quality of service measure of a communication network [6].

The trace file generated while simulating the main source code is given as input to the source code that generates throughput. The throughput code is written using the Perl language. The following equation is used to calculate the throughput based on the information in the trace file.

$$\text{Throughput} = \text{total number of bits received} / \text{time duration}$$

Perl script utilizes the agent trace information. Agent trace has information about the data packets only. The information about the routing and MAC packets is not considered in the calculation of throughput.

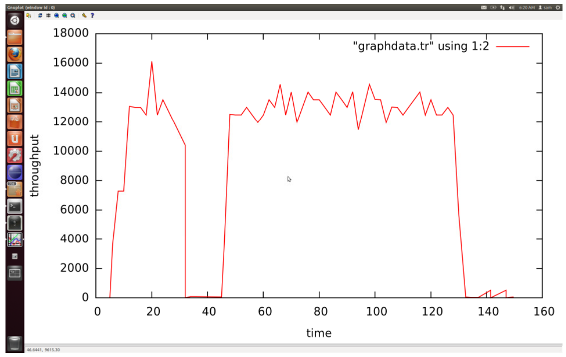


Figure 8. Throughput.

Figure 8 shows the graph of Throughput for the proposed network. The x-axis indicates time and y-axis indicates throughput of the network. The data transmission starts at 5 seconds. The throughput of the network changes from 5 seconds to 25 seconds due to the traffic and transmission delays in the network. When the node 3 fails at 26^{th} second, it is observed that the throughput drops to zero. This is because the network becomes disjoint as there is connecting path between node 0 and node 7.

Later LeDir algorithm is applied and node 3 is replaced with node 2. The repositioning of the nodes is done at the 32^{nd} second. Now, node 2 starts to take the responsibilities of node 3. The total simulation time is 120 seconds. At the 45^{th} second the data transmission starts again and thus the throughput is found to increase from 45 seconds to 120 seconds. The throughput gradually falls to zero again after few seconds. The maximum throughput of 16000 bytes is observed at time of twenty seconds.

3.2. End-to-End Delay:

End to end delay is the average time taken by a packet that is transmitted from the source to reach the destination. End-to-end delay includes all the delays in the network along with the propagation delay i.e. delays caused in the network for finding the shortest path and queues are also considered. The network is found to perform better when the end-to-end delay is low. The following equation is used in the calculation of end-to-end delay in the network:

End to end Delay = arrival time of packet – send time of the packet

The source code for calculating the end-to-end delay is written using the Gawk scripting language. We calculate packet delay using their packet ID. Packet ID contains information about the sending time and receiving time of the packet. The data trace file is given as an input to the gawk code to produce a trace file containing the delay information. The packet delays at different time intervals are contained in the trace file. A graph of time versus end-to-end delay is plotted using the GNU plot. The graph is shown in Figure 9. When the data transmission starts at 5^{th} second the end-to-end delay is found to be the lowest, 0.04 seconds. As the traffic in the network varies the delay is found to vary till node 3 fails where the end-to-end delay falls to zero. After the node recovery at 45^{th} second the delay is found to have the highest value of 0.11 seconds.

When the network has recovered from failure all the nodes take time updating the routing tables. Since the end-to-end delay considers all the delays in the network along with the transmission delay, the peak of end-to-end delay is found immediately after the node recovery.

3.3 Average Round trip time:

Average Round-trip time is the time duration between transmission of the packet to destination and reception of the acknowledgement signal. Round trip time depends on the number of nodes between the source and destination, traffic in the network, the number of other requests that are handled by the intermediate nodes. The average round trip time could vary from a few milliseconds to seconds. The lower the round-trip time the better is the throughput in the network.

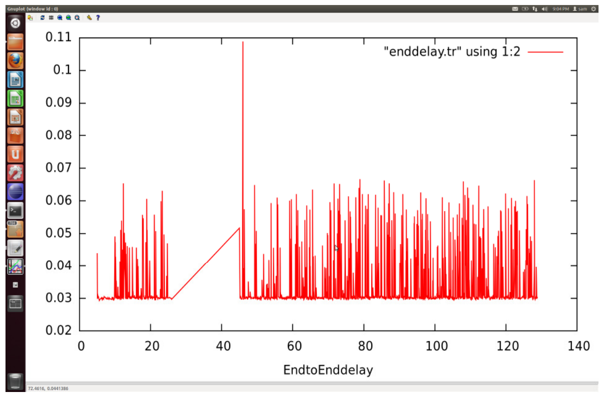


Figure 9. End-to-End Delay of Network.

As the transmission of packets starts at 5th seconds, the round trip time is the lowest of 0.3 seconds. This is because the traffic and other delays are limited at the beginning of the simulation. As time progresses, the round trip increases and varies based on number of other concurrent transmissions. When node 3 fails at 24th second, the round trip time falls to the lowest value. After the recovery of the data path between node 0 and node 7, the simulation begins again at the 45th second round trip time starts increases again. The variations in the round trip time from 45 seconds to the end of the simulation at 120 seconds are shown in Figure 10. When the round trip time is the highest, 0.08 seconds, the network performance degrades.

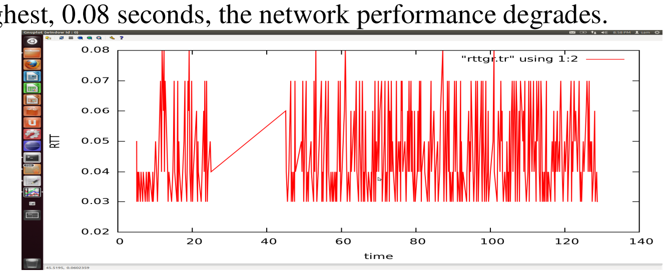


Figure 10. Graph of Time vs Round Trip Time.

3.4. Packet Delivery Ratio:

Packet deliver ratio is the ratio of successful packet delivery in the network. It is defined as the ratio of packets that successfully reached the destination to the packets produced by the source nodes. Higher packet delivery indicates better performance of protocols used in the network. It could be defined by using the equation below:

$$\text{Packet Delivery Ratio} = N1/N2$$

Where N1 is the number of data packets received at the destination node and N2 is the number of data packets generated at the source node. Gawk is used for the calculation of packet delivery ratio. Figure 11 shows the packet count at the source and the destination. 1238 packets are received at the destination while 1253 packets are originally transmitted from the source. The packet delivery ratio of the network is 98.88%. This shows that the network has better performance. Thus, LeDir algorithm could be applied in wireless sensor networks to maintain a constant performance of the network even in the presence of node failures and other critical events in the network.

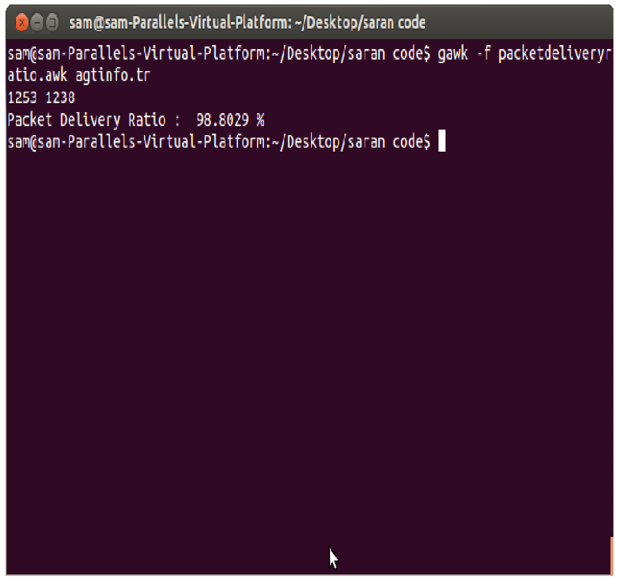


Figure 11. Packet Delivery Ratio.

4. CONCLUSION

Two different path restoration algorithms for wireless sensor networks-DARA and LeDir are studied. LeDir algorithm could be used to restore the wireless sensor network connectivity without extending the short path of nodes in the network and hence is more beneficial compared to DARA. It helps in improving the quality of the sensor network all times. LeDir is implemented on NS-2 to study the throughput and other parameters of the network. The algorithm could be used to recover only one transmission path at a time. In case of multiple node failures the procedure cannot be applied. But the probability that many nodes fail at some time is less and occurs when the deployment area is subject to a very hazardous event.

REFERENCES:

[1] P. Guo, T. Jiang, Q. Zhang, and K. Zhang, ”Sleep Scheduling for Critical Event Monitoring in Wireless Sensor Networks,” IEEE Transactions On Parallel And Distributed Systems, Vol. 23, No. 2, February 2012.

[2]. F. Akyildiz and I. H. Kasimoglu, “Wireless Sensor and Actor Networks: Research Challenges,” Ad Hoc Netw. J., Vol. 2, No. 4, PP. 351–367, Oct. 2004.

[3] A. Abbasi, M. Younis and U. Baroudi, “Recovering from a Node Failure in Wireless Sensor-actor Networks with Minimal Topology Changes”, IEEE Transactions on Vehicular Technology, Vol. 62, No. 1, January 2013.

[4] T. Zheng, S. Radhakrishnan and V. Sarangan, “PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” Proc. 19th IEEE Int’l Parallel and Distributed Processing Symposium, PP. 224-231, Apr. 2005.

[5] G. Wang et al., "Sensor Relocation in Mobile Sensor Networks," Proc. of 24th Annual Joint Conf. (INFOCOM), Miami, FL, Mar. 2005, PP. 2302–2312.

[6] P. Rohal, R. Dahiya and P. Dahiya," Study and Analysis of Throughput, Delay and Packet Delivery Ratio in MANET for Topology Based Routing Protocols (AODV, DSR And DSDV)," International Journal for Advance Research in Engineering and Technology, Vol. 1, Issue II, Mar. 2013.

[7] R. J. Egusha, B. Mathivanan and E. Konguvel, "A Distributed LeDiR Algorithm for WSA Networks," IRACST, 2250-3498, Vol.4, No.3, June 2014.

[8] "Introduction to Network Simulator NS-2" by T Issariyakul and E Hossain.

[9] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. Stankovic and T. Abdelzaher, "Real-Time Analysis of Tracking Performance in Wireless Sensor Networks," IEEE Real-Time Applications Symposium, May 2006

[10] M. Ismail and M. Y. Sanavullah "Security Topology in Wireless Sensor Networks with Routing Optimization", WCSN 2008.

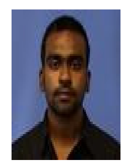
[11] A. Abbasi, M. Younis and K. Akkaya, "Movement-assisted Connectivity Restoration in Wireless Sensor and Actor Networks," IEEE Trans. Parallel Distrib. Syst., Vol. 20, no. 9, PP. 1366–1379, Sep. 2009.

[12] http://nsnam.isi.edu/nsnam/index.php/Nsnam.

[13] http://en.wikipedia.org/wiki/Ad_hoc_On- Demand_Distance_Vector_Routing.

[14] http://www.isi.edu/nsnam/ns/tutorial.

Niharika Gorre was born in India in 1991. She received the B.S. in Technology, ECE major, from Jawaharlal Nehru Technological University, India, in 2012 and M.S degree in Electrical and Computer Engineering from Purdue University, Indianapolis, Indiana in 2014. Her main areas of research interests are Multimedia and Networking.



Prudhvi Kyatham was born in India in 1991. He received the B.S. in Technology, ECE major, from Jawaharlal Nehru Technological University, India, in 2012 and M.S degree in Electrical and Computer Engineering from Purdue University, Indianapolis, Indiana in 2014. His main areas of research interests are Multimedia and Networking.

Mohamed A. El-Sharkawy: received the Ph.D. degree in electrical engineering from Southern Methodist University, Dallas, TX, in 1985. He is a Professor of Electrical and Computer Engineering at Purdue School of Engineering and Technology. He is the author of four textbooks using Freescale's and Motorola's Digital Signal Processors. He has published over two hundred papers in the areas of digital signal processing and communications. He is a member of Tau Beta Pi and Sigma Xi. He received the Outstanding Graduate Student Award from Southern Methodist University. He received the Abraham M. Max Distinguished Professor Award from Purdue University. He received the US Fulbright Scholar Award in 2008. He received the Prestigious External Award Recognition Award from Purdue School of Engineering and Technology, 2009. He is a reviewer for the National Science Foundation and Fulbright.