Energy Efficient Wireless Vehicular-Guided Actuator Network

Imene Boudellioua and Basem Shihada
Computer, Electrical and Mathematical Science and Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{imene.boudellioua, basem.shihada}@kaust.edu.sa

***Abstract*—In this paper, we present an energy-efficient vehicular guided system for environmental disaster management using wireless sensor/actuator networks. Sensor nodes within clusters are controlled by a master node that is dynamically selected. Actuators support mobility for every sensor node in the area of interest. The system maintains energy efficiency using statistical, correlation, and confidence for determining actuator actions and implements an adaptive energy scheme to prolong the system lifespan. Experimental results show that the system is capable of saving up to 2.7*Watt* for every 28KByte of data exchanged. We also show that actuator actions are correct with a 90% confidence.**

I. Introduction

Natural and man-made disasters have resulted in tremendous damages and loss of human life and environmental resources. The disaster response protocol currently used worldwide is based on a centralized, closed system that was originally derived from the military [1]. In large scale disasters, information centers quickly become overwhelmed and flooded with emergency input and inquiries. Wireless sensor/actuator network (WSAN) is one of the cutting edge technologies that can quickly report and respond to sudden events.

A WSAN is a geographically distributed set of sensor and actuator nodes that perform the phenomena of sensing and actuating in the network respectively. Sensor nodes are battery-equipped, low-power, low-cost devices with limited sensing, data processing, transmission range, memory, and communication capabilities [2]. Frequently, sensor nodes are densely deployed either inside or closed to an observed area. On the other hand, actuator nodes are richer in resources and are equipped with higher transmission power, better processing capabilities, and longer battery life. Therefore, actuators can act on wider areas and hence, only few of them are often needed. These actuators vary in size and complexity depending on the application requirements.

There are many applications that necessitate the use of actuators along with the sensors in the environment [3], [4], [5]. Particularly, there is a great interest in developing applications for monitoring, diagnosis, and control in the medical, industrial, environmental, and agricultural sectors. The goal of these applications is to improve social and environmental conditions of society, and to increase quality and productivity levels in industrial processes.

In this paper, we implement a reliable and energy efficient WSAN that provides movement-assisted services to all the sensor nodes. We formulate a system model that consists of a cluster of sensor and actuator nodes led by a master node. The guided movement of the nodes is achieved by using a range-based localization method, *Received Signal Strength Indicator, RSSI* as a measurement for distance estimation between any node and its master node. We propose an actuation reliability scheme that minimizes errors with regard to false movement decision due to potential presence of interference. We also integrate an adaptive power management for the wireless radio system component to enhance the overall system energy consumption.

The rest of the paper is organized as follows. A literature review is provided in Section II. Section III describes the design of our system model and the implementation platform. Section IV introduces our statistical approach for determining *RSSI* threshold and our adaptive power scheme. Our experimental study and performance analysis is discussed in section V. Finally, a summary of our work and directions for future research is highlighted in section VI.

II. Related Work

Wireless sensor networks have been extensively studied in the past decade while limited number of studies were done for WSANs systems. A real-time architecture for automated WSAN to bind the latency in applications was proposed in [6]. Shah *et. al.* [7] proposed a real-time coordination and routing framework that addresses the coordination of sensors and actuators. Durresi *et. al.* [8] designed a scalable geometric broadcast protocol for WSAN that imposed very low communication overhead. Boukerche *et. al.* [9] proposed a routing protocol with service differentiation for WSANs that provides low latency and reliable delivery in the presence of failures. Melodia *et. al.* [10] demonstrated the first distributed coordination framework for WSANs based on an event-driven clustering paradigm to make a tradeoff between energy consumption and delay for data transmission. Another interesting real-time communication framework was presented by Ngai *et. al.* [11]. It supports event detection, reporting, and actuator coordination.

While the above studies have explored the potentials of WSANs, action reliability and energy conservation for event reporting from sensors to actuators have yet to be addressed and studied thoroughly along with the real-time requirement of WSANs applications. The following section provides an overview of the system components and protocols.

III. System Overview

Our system is designed for a vehicular guidance application where actuators provide mobility to stationary sensor nodes whenever needed. The system satisfies three fundamental requirements which are real-time constraint, reliable actuation decisions, and efficient energy consumption. A set of control messages should be exchanged between the sensor nodes and their master in a given neighborhood in order to enforce these constraints while minimizing the signalling overhead. Moreover, we introduce a node discovery protocol which will allow nodes to join and leave neighborhoods freely. This assignment is handled by a dynamically selected master node for each neighborhood. The actuator node decisions rely on statistical measurement of the *RSSI* as a range-based localization scheme for estimating the distance between the nodes. Basically, a predefined maximum distance between any node and their master in the cluster is specified and whenever exceeded, actions will be taken. We assume a single-hop communication scheme, bi-directional wireless links, single channel, and omnidirectional radios. The proposed system is independent of the underlying communication protocols with no modifications required to lower layers in the network protocol stack. Finally, it is computationally cheap with only a small runtime overhead. This meets well the general WSAN design requirements with regard to resource constraints on data processing capacity and energy consumption in sensor and actuator nodes.

A. System Operation

Assume a cluster of four nodes $N1$, $N2$, $N3$, and $N4$ with $N1$ being the current master node. Nodes $N2$, $N3$, and $N4$ transmit their data to node $N1$ which later aggregates them and transmits them to the base station whenever requested as in Figure 1. Also, consider a minimum threshold *T* for *RSSI* determined based on a specific reference model for localization that reflects the maximum distance allowed in the application between any node and its master node. This threshold is application specific and varies depending on the system requirement and hence, must be defined by the user. Consider a request from the base station to $N1$ commanding it to move to a different location:

1) When *discovery period* timer is fired, slave nodes exchange discovery messages with the master node to update the status of the master node and other nodes in the neighborhood. If any of the slave nodes does not respond back to the master, then the node is considered out of the neighborhood.
2) Each slave node measures the *RSSI* value of the packet received from the master.
3) If the measured *RSSI* value is less than the minimum threshold *T*, the node sends a control message to the master node to ask about its status (*i.e.,* move or stop). Otherwise, if the measured *RSSI* is greater than or equal to *T*, no action is required and the node remains static.
4) If the slave node receives a reply for its control message, the node signals a command to the attached actuator and moves towards the master node. Otherwise, the node remains static.
5) The procedure from 1-4 is repeated for every discovery period.

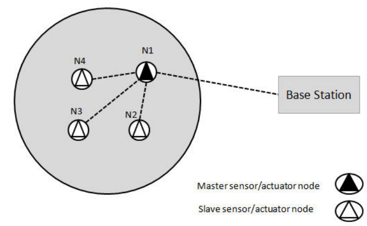

Fig. 1. Basic system topology

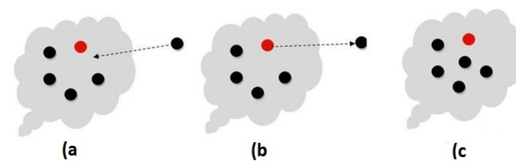

Fig. 2. a) Node broadcasts DIS packet b) Master replies with REP packet c) Node is now part of the neighborhood. (Red circle denotes master, black circle denotes slave)

B. Node Discovery Protocol

Node discovery is crucial to the system design and becomes challenging in the presence of mobility [12]. We implement a one-hop node discovery protocol in which nodes join a particular neighborhood in an incremental fashion. The master node assigns a unique numeric network ID to each node. Master node is the first to start a neighborhood and maintains the smallest network ID. Also, we assume fully connected neighborhoods where no sensor node is a member of more than one neighborhood at a given time. In our system, the criteria for ID assignment is that nodes are deployed with the richer node in resources being first so it gets a smaller network ID and so on. In certain region, if a node runs a discovery protocol and did not find any neighbors, it starts a neighborhood of its own and becomes the master. Generally, every predetermined discovery period (*e.g.,* 2000 *ms*), current member nodes of a given neighborhood broadcast discovery packets (DIS) including their network ID to the master node to ensure their availability as active nodes. The master node will then acknowledge receiving DIS packets by sending back a reply (REP) packet to each node in the network confirming their status.

During the discovery period, if a node discovers a neighborhood, it will ask to join the neighborhood. Figure 2 depicts this case. On the other hand, a node wishing to leave a neighborhood would simply leave without sending a request.

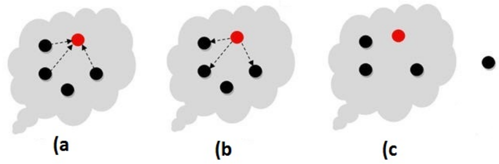

Fig. 3. a) Node does not broadcast DIS packet b) Master assumes node had left the neighborhood c) Node is removed. (Red circle denotes master, black circle denotes slave)

Basically, at the next discovery period, the master node will determine that a node with a certain network ID left the neighborhood. Then, it updates its table with regard to the currently active nodes of the neighborhood. Figure 3 depicts the scenario when a node leaves a neighborhood.

C. Master Node Selection

The role of a master node is vital for effective management of the neighborhood. The master node is responsible for initiating movement actions while slaves follow. We implement a master handshake mechanism for selecting a master node whenever necessary. Algorithm 1 highlights the procedure of master handshake in a given neighborhood *N* with *O(n)* iteration, where *n* is the number of nodes in the neighborhood. Hence, cluster size should remain relatively small. It is also important to note that any metric could be used for master determination. Remaining battery, resource capabilities, lifetime, are examples of metrics that could be used. The scenarios that initiate this mechanism are the following:

- *Establish a new neighbourhood*
  When a node is placed in a region and could not find any neighbourhood to join, it will select itself as a master node and start a new neighbourhood. No handshake is required as the only candidate is one node and thus, it will automatically become the master node.
- *Replace an out-of-service master node*
  As explained before, nodes in a neighbourhood exchange periodic discovery packets with the master node whenever they wake up. If the master node fails to reply with REP packet to the node's DIS packet, it means it has left the neighbourhood. In this case, we need to nominate a new master among the remaining active nodes.

Algorithm 1 Master Selection Procedure

1: Each node in cluster N becomes a candidate for master node role.
2: Candidate nodes exchange with each other their network metric via a broadcast communication fashion.
3: Node with the best metric will win and become the new master node.
4: Winner node will announce it to all nodes in N.
5: Other candidate nodes will go back to their previous state as slave nodes.
6: Regular communication will resume with the new master node in N.

IV. System Analysis

In this section, we focus on two challenging aspects of the system design. First, we propose a statistical approach for determining *RSSI* threshold. Second, we introduce an adaptive power scheme for different values of *RSSI*. These factors are essential for the long-term system sustainability and functionality.

A. RSSI Threshold (T)

When the master node triggers a movement action, it will immediately signal its attached actuator and start moving. The communication schemes between the nodes in the neighborhood will follow the same pattern. We introduce *RSSI* measurement that should be taken by each slave node from the REP packet received from the master node. This measurement is compared against an estimated threshold value for a maximum allowed distance between any node and its master. When a node finds that its measured *RSSI* value is beneath the specified threshold, a control packet will be send to the master node to ensure taking proper decisions. This is because wireless medium may suffer from interference that may degrade the link quality and hence the *RSSI* measurement will induce an inaccurate distance estimation that may lead to false actuation. Therefore, we use a statistical scheme for selecting a proper *RSSI* threshold for actuation decision on the fly. We introduce two parameters M and N. The parameter M is the number of consecutive *RSSI* values for a specific distance measured for the long-term statistics. The parameter N is the number of consecutive *RSSI* values measured prior to an actuation event for the short-term statistics for a specific distance (chosen to be 2 meters). The average of N values, denoted by $avgN$, is compared with the average of M values, denoted by $avgM$ in order to determine how likely an actuation event will occur in the system. If an actuation event happens, then $avgN$ is expected to be smaller than $avgM$, whereas when $avgN$ is close to $avgM$, this indicates a clear communication medium with minimal interference.

We further quantify the relation between the long-term and short-term statistics in order to define the confidence with which a threshold value is accurate for actuation decision. We assume that the M consecutive values are random with a mean $avgM$ and a variance $varM$ which resulted into approximately Gaussian distribution as in Figure 4.

Selecting a proper value of N is fundamental, since N values are expected to provide sufficient information about the short-term network status when an actuation event is needed. If N is chosen too small or too larger, the short-term network status may not be accurately represented. Our approach in selecting N employs an autocorrelation function:

$$R(0,N) = \frac{1}{N} \cdot \sum_{i=0}^{N} RS(i) \cdot RS(i+N),$$

where *RS(i)* is the *ith RSSI* reading. The numbering of *RSSI* values is shown in Figure 5.

In our scheme, the value of N is selected such that $R(0,N) = R(0,0) \cdot \gamma\%$, where γ is the threshold that determines N based on the autocorrelation function. In order

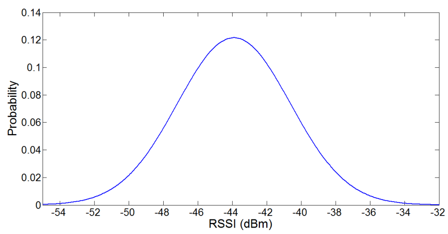

Fig. 4. RSSI values distribution histogram.

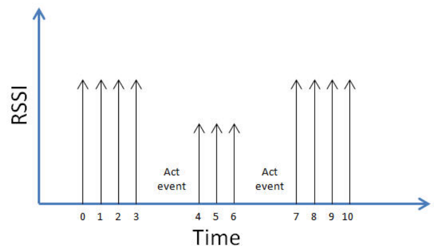

Fig. 5. Numbering of RSSI values over time

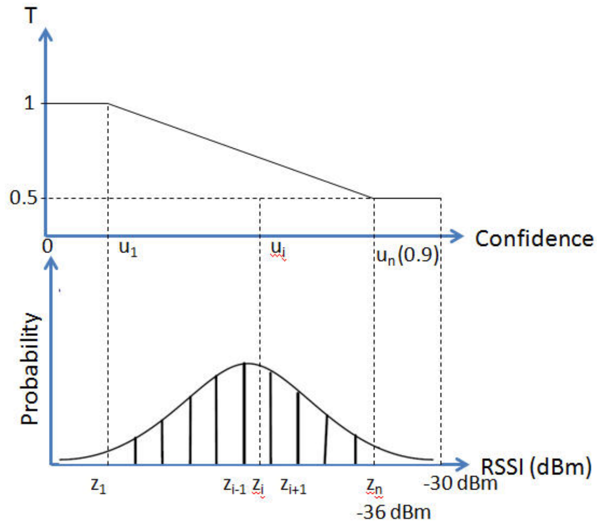

Fig. 6. The relation between z_i, u_i, and T

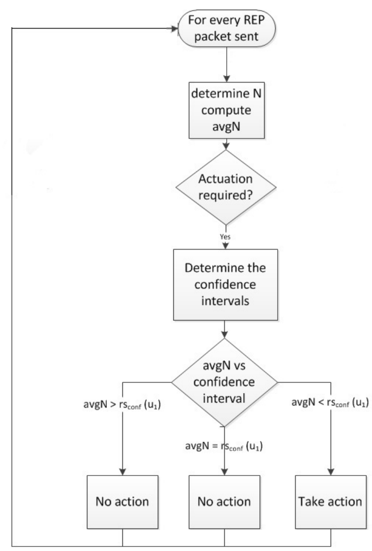

Fig. 7. The proposed policy.

to well-represent the short-term network status, the N values should have a strong correlation with each other. In our study, γ is taken as 90%. After a proper N is selected based on the approach above, the value of $avgN$ can be obtained. Then, we define the confidence with which the current *RSSI* value indicates an actuation event by positioning $avgN$ in the distribution spectrum of the M values collected. The derived confidence is used to dynamically adjust T as a threshold for actuation decision. A function $z_i = rs_{conf}(u_i)$ is defined for positioning $avgN$ in the Gaussian distribution spectrum, where u_i is the confidence level. The $rs_{conf}(u_i)$ returns an *RSSI* value (denoted by z_i) which is larger than a proportion u_i of all *RSSI* values in the Gaussian distribution curve. A one-to-one mapping between u_i and z_i exists, as per,

$$u_i = cdf(z_i) = \sum_{j=0}^{i} pmf(z_j) \tag{1}$$

The $cdf(z_i)$ and $pmf(z_i)$ denote the cumulative density function (CDF) and probability mass function (PMF) in the *RSSI* spectrum given the *RSSI* value of z_i. The one-to-one mapping between u_i and z_i is shown in Figure 6. In this figure, for example, if the *RSSI* value is higher than the mean *RSSI* values with $u_i = 90\%$ confidence, then the *RSSI* value of z_i is in the range of -36 to -30 dBm.

We define the following policy for determining the value of T. When $avgN$ is smaller than $z_1 = rs_{conf}(u_1)$, an actuation event is indicated. On the other hand, when $avgN$ is greater than or equal to $z_n = rs_{conf}u_n$, it is a strong indication of good network communication. Hence, no actuation is performed. This policy is summarized in the flow chart illustrated in Figure 7.

B. Adaptive Power Scheme

WSAN systems often suffer from high communication overhead. Therefore, we implement an adaptive transmission power scheme to reduce the system power consumption. Assume a network of K nodes. For the K_i node, $i \in \{1, ..., K\}$, the location of the nodes follow a uniforml distribution over the distance range$[d_{min}, d_{max}]$. d_{max} is the maximum distance with acceptable $0.01 > PER > 0.001$, where PER stands for packet error rate. Then, the corresponding PDF of each node's location at certain distance d_i is expressed by,

$$Pr\{D_{M_i} = d_i\} = \frac{1}{d_{max} - d_{min}} \tag{2}$$

From Eq. 2, we obtain the expression for the average energy consumption without implementing the adaptive power scheme as,

$$E_{ave} = \sum(E(d_{max})Pr\{D_{M_i} = d_i\}) = \frac{N}{d_{max} - d_{min}} E(d_{max}) \quad (3)$$

On the other hand, the average energy consumption while implementing adaptive power scheme is,

$$E_{ave} = \sum(E(d_{max})Pr\{D_{M_i} = d_i\}) = \frac{1}{d_{max} - d_{min}} E(d_{max}) \quad (4)$$

Then, the energy saving using adaptive power scheme compared to the non-adaptive power scheme is,

$$E_D = Pr\{D_{M_i} = d_i\}[NE(d_{max}) - \sum(E(d_i))] \quad (5)$$

We experiment with the adaptive power scheme energy saving using TelosB data sheet. Assume $K = 8$, and $d \in \{6.25, 12.5, 18.75, 25, 31.25, 37.5, 43.75, 50\}$. Then,

- $E_{ave} = 8 * (17.4 * 3.3) = 459.36$ mW
- $E_{(ave-PAS)} = \{17.4+16.5+15.2+13.9+12.5+11.2+9.9+8.5\} * 3.3 = 346.83$
- $E_D = \frac{112.53}{43.75} = \frac{2.57mW}{transmission}$

Thus, assuming a maximum packet size of 28 Bytes, transmitting 1000 packets of this size saves 2.57 *Watt*. We implemented the adaptive power scheme to obtain $P_{(Rx-th)}$ and determine the threshold to decide the desired transmission power. Our system uses the RTS signal to obtain the new transmission power P_{new}, and send it back to the slave using CTS.

V. System Evaluation

Our system is evaluated using IEEE 802.15.4-compliant devices. In our experiments, we used TelosB motes from Crossbow Technology which employ *CC2420* radio chip. This chip offers more accurate measurements for *RSSI* compared to *CC1100* for estimating wireless link quality. TelosB motes are powered from an external battery pack containing 2 AA batteries. TelosB also has a 10-pin expansion connectors that can be configured for additional units and devices. The 10-pin connector is the primary connector that provides digital input/output as well as analog inputs. We used TinyOS 2.1.1 to program our system components. Actuation is provided via modified wheeled servo motors that are wired to every sensor node and will provide mobility whenever needed. The experimental study was conducted indoor in a scientific lab environment where interference is present. Unless stated otherwise, the default TinyOS and operating parameter values were used. Also, in some experiments, we used channel 11 as it overlaps with the first channel of 802.11 to analyze the system behavior in the presence of high interference. The experimental parameters are summarized in Table I.

TABLE I
PARAMETERS CONSIDERED IN THE EXPERIMENT

Parameter	Value
Max payload size	28 bytes
Radio data rate	250 kbps
Duty cycle	100 %
Discovery period	1000 ms
Transmission power	0 dBm
Channel	11

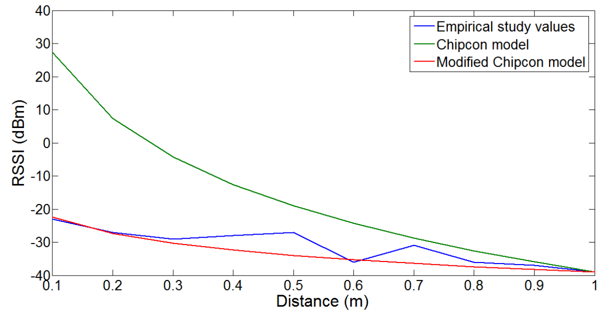

Fig. 8. Comparison between different RSSI (dBm) vs. Distance (m) experimental measurements and model estimation

A. Distance Estimation

In order to define a reference model, an empirical study of *RSSI* and distance is needed. For this purpose, we measured the values of *RSSI* obtained from a sensor node communicating with another node over various distances in our indoor environment averaged over 5 runs. We compared the above measurements with Chipcon formula values for estimating *RSSI* values with distance. However, we found that this formula does not satisfy our empirical study values due to the characteristics of our indoor environment with potential external WiFi interference. Hence, we propose our own reference model as,

$$RSSI = \frac{-10}{n} log_n(d) + A \quad (6)$$

Where *n* is propagation exponent, *d* is the distance from the sender in meters and *A* is the received signal strength in dBm at one meter of distance. In the study, the propagation exponent in free space is set to 2. Figure 8 highlights the comparison between our empirical study values, Chipcon formula, and our modified Chipcon reference model. We can clearly see that our model is much more suitable for our empirical study as it is very close to the measurements we obtained from the sensors.

For this experiment, we assume that the maximum acceptable distance between any given node and its master node in a certain cluster should not exceed 2 meters. Hence, with a maximum distance of 2 meters, under our experimental environment, the *RSSI* values should be not less than -45 dBm according to Eq. 6.

We used three different initial topologies for our performance evaluation. Note that these topologies will eventually change to a random topology as the actuation implemented is 1-dimensional. These topologies are shown in Figure 9. We

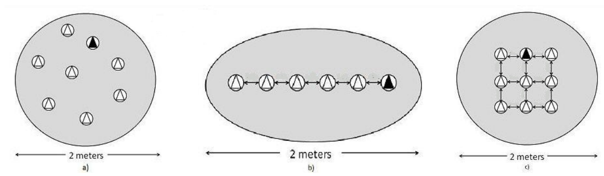

Fig. 9. a) Random topology b) Parking lot topology and c) Grid topology. (Circle with black triangle denotes the master)

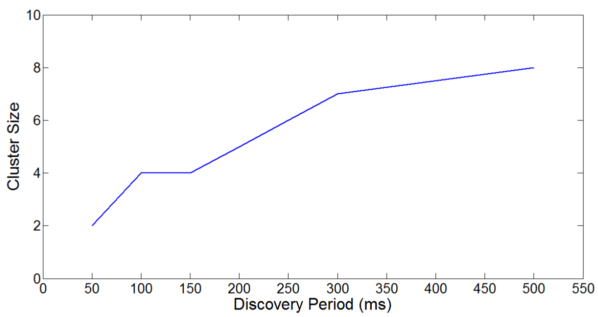

Fig. 10. Minimum discovery period vs. different cluster sizes

selected these well known topologies in order to maximize the coverage and minimize the interference caused by packets forwarding between the nodes.

B. System Convergence

Determining a sufficient discovery period in which a sensor node is awake and able to communicate with the other nodes is critical for the network fast convergence in node discovery phase. We studied the behavior of our system with different number of nodes using various discovery periods averaged over 5 runs. Observing our results, we found that the minimum sufficient discovery period generally increases as the number of nodes present in a given cluster increase. Figure 10 illustrates this finding. For example, the discovery period should be set to a minimum of 500 ms in order to construct a converged cluster of 8 nodes.

C. Actuation Reliability

In this experiment, we compared the actuation error rate of the system when control packets are used in comparison to when the decision is solely made based on *RSSI* upper bound. The actuation error rate basically stands for the percentage of false decisions taken by a node at a given time. Figure 11 depicts our results averaged over 3 runs with each run lasting 300 seconds for the above three different topologies. We observe that the control packet exchange scheme enhances the actuation reliability of the system significantly as the error rate is much smaller than the rate of a typical *RSSI-based* actuation approach. Reducing the actuation error rate will contribute to energy saving of the node as faulty actuation commands are avoided. We also found that initial parking lot topology lead to the best overall actuation rate in comparison to the other topologies. This is due to the property of the omnidirectional antennas integrated on the sensor nodes which cause variance

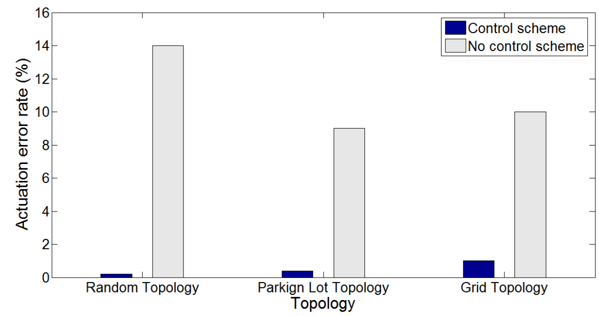

Fig. 11. Actuation error rate vs. controlled/non-controlled schemes

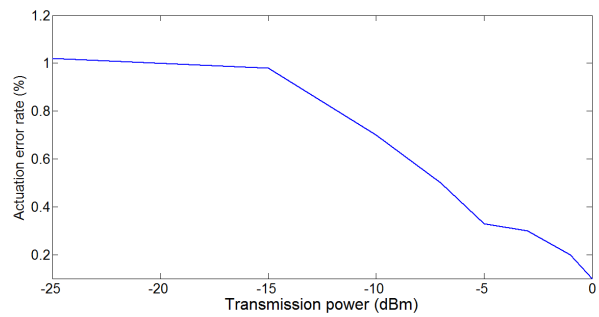

Fig. 12. Actuation error rate vs. transmission power

in *RSSI* measurements depending on the placement of the node.

D. Adaptive Transmission Power

Transmission power affects the quality of the radio communication between the sensor/actuator nodes. We use our random topology to measure the actuation error rate of our system with the proposed adaptive transmission power scheme. Figure 12 displays the results averaged over 3 runs with each run lasting 5 minutes. Basically, the actuation error rate decreases as the transmission power increases. This is logical as increasing the transmission power results in stronger *RSSI* measurement and minimizes potential control packet loss which eventually enhances the network performance.

E. Effect of Interference

Introducing effective interference is important to capture the behavior of the system under real-world conditions. Concurrent packet transmission is a suitable way of jamming the communication between two transmitter and receiver nodes. Our strategy is to disable the CSMA/CA in the jammer node so that it will keep on transmitting packets regardless of what is going on in the neighborhood. However, normal nodes in the neighborhood with the default setting of CSMA/CA will not be able to transmit their packets due to the jamming presence. The *Packet Reception Rate (PRR)* is approximated as the probability of successfully receiving a packet between two communicating sender and receiver nodes. If *PRR* is high, it implies a high link quality and vice versa. For simplicity, we considered a cluster of two nodes, one as a master node

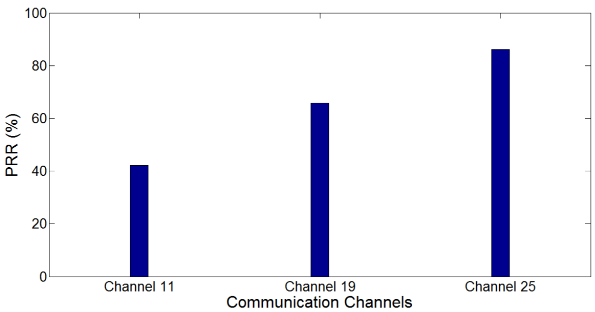

Fig. 13. PRR in presence of jamming signal

and the other as a slave node, initially positioned 1 meter away from each other with the jammer node in between them. We calculated the *PRR* at the slave node side in the presence of the jamming node. We studied this layout using three different channels and averaged the results of 3 experimental runs with each one lasting 5 minutes. The results are shown in Figure 13. We observe that channel 11 perform the worst as it interferes with channel 1 of the 802.11 band present in the lab environment. On the other hand, channels 19 and 25 both have better performance than channel 11 with channel 25 has a slightly better *PRR* of around 86%.

VI. Conclusion

In this work, we presented a vehicular guided wireless sensor and actuator network system for disaster management. The network consists of a set of clusters of sensor and actuator nodes interacting with each other. Actuation component in this system provides mobility to the sensor nodes. An actuation reliability scheme is introduced via exchanging control packets to ensure reliable and correct actions. Our experimental study shows that it is not efficient to rely solely on RSSI measurements for actuation decisions especially in critical applications with low fault tolerance. We overcome this deficiency by using the statistical approach.

Future enhancements would include testing the system using different localization schemes, such as range-based or range-free ones. Also, considering multi-hop communication pattern between the sensor and actuator nodes can be interesting yet challenging. Data aggregation methods and processing between nodes should be implemented efficiently to comply with the challenging characteristics found in WSANs. Another interesting improvement is to extend the movement process of the actuation to 3-dimensional space.

References

[1] D. A. McEntire, *Disaster Response and Recovery*. Wiley Pathways, 2006.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[3] R. K. A. K. M. Srivastava, "Distributed control over ad-hoc sensor actuator networks," Department of Electrical Engineering, UCLA, Tech. Rep., 2004.

[4] S.-F. Li, "Wireless sensor actuator network for light monitoring and control application," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 2, jan. 2006, pp. 974 – 978.

[5] P. Pagilla, "Real-time scheduling of sensor and actuator networks," Master's thesis, Oklahoma State University, 2005.

[6] Y. Zeng, C. Sreenan, and G. Zheng, "A real-time architecture for automated wireless sensor and actuator networks," in *Wireless and Mobile Communications, 2009. ICWMC '09. Fifth International Conference on*, aug. 2009, pp. 1 –6.

[7] G. A. Shah, M. Bozyiğit, O. B. Akan, and B. Baykal, "Real-time coordination and routing in wireless sensor and actor networks," in *Proceedings of the 6th international conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, ser. NEW2AN'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 365–383.

[8] A. Durresi and V. Paruchuri, "Geometric broadcast protocol for sensor and actor networks," in *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, vol. 1, march 2005, pp. 343 – 348 vol.1.

[9] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "Communication and coordination in wireless sensor and actor networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 1116–1129, 2007.

[10] ——, "A distributed coordination framework for wireless sensor and actor networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '05. New York, NY, USA: ACM, 2005, pp. 99–110.

[11] E. Ngai, M. Lyu, and J. Liu, "A real-time communication framework for wireless sensor-actuator networks," in *Aerospace Conference, 2006 IEEE*, 0-0 2006, p. 9 pp.

[12] V. Dyo and C. Mascolo, "Efficient node discovery in mobile wireless sensor networks," in *Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, ser. DCOSS '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 478–485.