Manuscript received April 12, 2011; revised June 2011. DOI: 10.2478/v10177-011-0040-9

Socionomic Modelling in Wireless Sensor Networks

Sourendra Sinha and Zenon Chaczko

***Abstract*—The performance and efficiency of a Wireless Sensor Network (WSN) is typically subject to techniques used in data routing, clustering, and localization. Being primarily driven by resource constraints, a Socionomic model has been formulated to optimize resource usage and boost collaboration among sensor nodes. In this paper, we present several experimental results to ascertain the underlying philosophy of the Socionomic model for improving network lifetime of resource constrained devices – such as, sensor nodes.**

***Keywords*—Routing, clustering, localization, socionomic.**

I. Introduction

ADVANCES in micro-electro-mechanical-systems (MEMS) have enabled the development of miniature sensor nodes with extremely low-power requirements [1]–[3]. Hence, fuelled by the reduced cost, size, and complexity of such devices has revolutionized the development and deployment of WSNs in a range of different domains. By definition, WSNs consists of a set of sensor nodes strewn across an ad hoc area and networked with wireless links. For the nodes to communicate with each other, one or more nodes utilize the in-built transceiver device to transmit messages to neighbouring nodes. The same principle is also used to relay messages to a base station (or sink) either by a direct or multi-hop communication path.

However, sensor nodes are typically deployed in vulnerable environments and are thus expected to function unassisted and unhindered for a length of time [4]. Also, by virtue of their limited power, computing capability, and storage, the design and management of sensor nodes impose many challenges. In order to circumvent these challenges, the design of the fundamental functions of sensor nodes needs to encapsulate aspects of energy-awareness and inter-node collaboration [5]. In fact, aspects of energy-awareness have already been achieved in the physical and link layers by dynamic voltage scaling, improved transceivers, optimized duty cycles, and energy-aware MAC protocols. However, for a more long-term and sustainable solution refinements are also necessary in the network layer. In order to achieve such refinements, this paper presents a new data routing algorithm called SNIPER that is based on a Socionomic model involving a banking system.

Typically, a society is composed of a disparate set of individuals who are able to react, learn and adapt to their environments, thereby leading to the formulation of complex social and economic phenomenon [4]. The resulting social entropy is best exemplified by the autonomous traits in human beings living together in groups to benefit from shared experience, mutual contributions and sharing knowledge. However, these autonomous traits also lead to differences in opinion, experience and resulting action. Therefore, no two individuals in a society are ever exactly alike, and one of the means of assessing the underlying dynamics is through a study of the socio-economic constraints, or Socionomics.

In this paper, we have considered the human society as a system driven by its resource constraints and gauged by the cost of harvesting those resources. Although, such a socio-economic model cannot necessarily be formalized, yet, in software intensive systems it establishes a new paradigm for resource management. The results of our experimentation work conducted in the application of the Socionomic model against traditional techniques of clustering and routing are presented in following sections.

The rest of this paper has been organizes as follows: Section II introduces the Socionomic model, and discusses its application to clustering, routing, and localization in WSNs. Section III presents some of the experimental work conducted in validating the Socionomic model. Finally, the paper is concluded with a summary of the research work along and prospects for further analysis in the future.

II. The Socionomic Model

Social systems are composed of complex entities and in order to apply it as a computational framework, it is important to account for the intricacies in the relationships of these entities. Thus, the framework utilizes software agents as the basic building block in modelling a system. By definition, agents are characteristically adaptive, reactive, proactive and autonomous and over time they tend to build up their knowledge base by virtue of their set of competencies.

In a typical society, human beings interact, communicate and adapt to changing circumstances, and thereby aid in continuously evolving the system as a whole. As expressed by Friedman, evolutionary mechanism result in systems composed of only those agents who employ high degrees of rationality and information processing skills [4]. However, unlike their human counterparts software agents are yet abstained from the abilities to judge and make decisions. Thus, in dealing with computational agents the adaptive and learning mechanisms employed must be driven by heuristics on human learning to in turn govern their learning and adaptation techniques.

A. The Minority Game Model

The socionomic framework, as discussed in the previous section, uses economics as a means of both monitoring and controlling the dynamics resulting from the interaction of various agents in a society. Naturally, it can be assumed that the dynamics of a society are resource driven and in order to

S. Sinha and Z. Chaczko are with University of Technology, Faculty of Engineering, Sydney, Australia (e-mails: souren.sinha@gmail.com, zenon@eng.uts.edu.au).

monitor the behaviour of the agents we have used the Minority Game Model.

According to this model, in a typical WSN consisting of N nodes, the lifetime of the network depends on the level of interaction of the constituting agents. Each agent is essentially resident inside a sensor node, and during the initial setup the nodes are likely to be placed in a virtual cluster formation. Each of these clusters is managed by a single sensor node – the Cluster Head. As the popularity of the Cluster Head increases, more sensor nodes become part of the cluster, thus increasing its size. In order to manage the total population of the cluster, it is important for the Cluster Head to maximize its lifetime as much as possible.

As per the notion portrayed by the Minority Game Model, a set of N agents participating for acquiring resources in an ad hoc sandbox environment. At every time step t the agents are given the opportunity to either take part in the game by investing in the resource pool or just staying idle. Thus, the performance of each agent at time t lies between [-1, 1], and therefore the social utility function can be represented as $u(t) \in [0, 1]$. Therefore, at each time step the profit $u_i(t)$ of an agent may be calculated as:

$$u_i(t) = a_i(t)R\left[\frac{A(t)}{N}\right] \tag{1}$$

According to Kets, based on the individual profit of each agent, the aggregated investment $A(t)$ can be recorded as [6]:

$$A(t) = \sum_{j=1}^{N} a_j(t) \tag{2}$$

Where, $a_j(t) \in [-1, 1]$ is the action taken by agent i at time step t.

In a typical WSN, the Minority Game Model is quite analogous to the behaviour of the sensor nodes in that they must either participate in relaying data to other nodes or remain idle in a SLEEP state with the radio turned off. Depending on the state of the agent the residual energy would vary over time and thus affect the lifetime of the network itself. Since the agents are allowed to behave in a stochastic manner and the information efficiency H can be derived by accounting for the probability that an action will be taken by an agent.

$$H(\pi) = \langle A\rangle^2 \ where \ \langle A\rangle = \sum_{i=1}^{N}(2\pi_i - 1) \tag{3}$$

The learning model described by Ken Wets [6], also mentions that the Minority Game enables agents to make strategic decisions based on patterns identified in the game's history of last m actions. Essentially, for all actions in the game's history H_m, a response mode s assigns to each set of information h_m that is given by:

$$h_m \in H_m = \{(x_k)_{k=1,2,\ldots,m}\} x_k \in \{-1, +1\} \tag{4}$$

Thus, at each time step tthe response mode s determines action $s(h_m(t)) \in [-1, +1]$ that can be taken. Since, for each action there are basically two possible responses [-1, +1], for a set of responses from 3 different modes, the resulting actions can be represented as:

History			Action			
H_m			s_i1	s_i2	s_i3	s_i4
-1	-1	-1	+1	-1	-1	+1
-1	-1	+1	-1	-1	+1	-1
-1	+1	-1	+1	-1	-1	+1
-1	+1	+1	-1	+1	-1	+1
+1	-1	-1	+1	+1	+1	+1
+1	-1	+1	-1	-1	+1	+1
+1	+1	-1	-1	-1	-1	+1
+1	+1	+1	-1	+1	-1	+1

III. SNIPER ROUTING

A WSN tends to be composed of a homogenous set of sensor nodes that are randomly distributed in a given environment [7]. In order for the nodes to efficiently route data among each other, relay points are nominated that are essentially the Cluster Head (CHs). However, in any form of data routing packet loss is a common phenomenon that needs to be accounted for in judging the effectiveness of the routing technique. Packet loss is typically caused by weather patterns, radio interference, or even local node hardware. To judge the effectiveness of the routing techniques, we have formulated a link-cost model based on the following set of parameters:

- The processing time (t_{pcs}),
- The channel acquisition time (t_{ch}),
- Time for transmission (t_{Tx}),
- Queuing time (t_Q),
- Propagation delay (t_{Pr})
- Retransmission timeout (t_{RT})
- Packet loss error rate (τ)

The link-cost model is based on a simple analogy that for every sensor node to successfully transmit a packet an initial amount of energy will be required [8]. Additional amounts of energy may be necessary for re-transmitting a packet, in the event of packet loss. The general equation for the link-cost model can be expressed as:

$$Cost_{ij} = (t_{pcs} + t_{PR}) + \left(\frac{\tau}{1-\tau}\right) \cdot (t_{pcs} + t_{RT}) \tag{5}$$

The processing time for each sensor node is a sum of the time required to assimilate the data into the right packet format, and acquire the right channel on the radio. In case the channel acquisition fails, the transmission must be postponed for a time period (t_{wait}). Therefore, the channel acquisition time is finally calculated as:

$$t_{ch} = \left(\frac{\vartheta}{1-\vartheta}\right) \cdot t_{wait} \tag{6}$$

Where, t_{wait} is the time to wait till the next retransmission ($\sim$3 secs), and ϑ is rate at which transmission failure occurs ($\sim$5%).

Using the channel acquisition time, the processing time for each sensor node may be calculated as:

$$t_{pcs} = t_{ch} + t_{Tx} + t_Q \tag{7}$$

Finally, the propagation delay involved in transmitting a packet from one node to another is dependent on the distance travelled, and the applicable latency in communication –

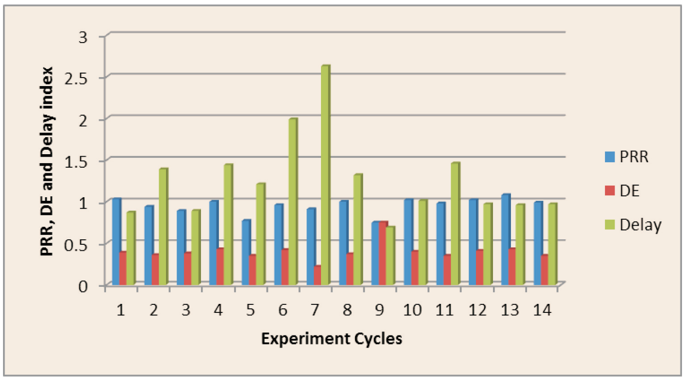


Fig. 1. Plot of the relay point analysis.

typically 67% the speed of light (3×10^8 m/sec).

$$t_{PR} = \frac{d}{s} \tag{8}$$

Where, d is the distance, and s is the latency in packet transmission.

The relay point routing technique adopted in SNIPER involves three different routing techniques, namely:

- *Nearest Neighbour* – the node located nearest to the source node is automatically selected to serve as a relay node,
- *Lowest Cost* – at each hop the recipient node selects a relay node for the next hop based on the associated cost,
- *Round Robin* – the simplest method of all whereby at each hop the recipient node sequentially selects a new relay node from its neighbour list.

The experiment carried out involved a deployment of 250 homogenous sensor nodes in a simulated environment, conducted over several hours. Following the completion of the cluster formation, three different static event sources were deployed in the environment. Three sensor nodes located closest to the event sources were configured to transmit the event data at a rate of 4 packets every hour. The results obtained were plotted along with the respective Packet Reception Rate (PRR), Delivery Efficiency (DE), and delay.

In routing, a relatively high PRR is important because it is indicative of the efficiency of resource utilization in the network. Similarly, a low PRR results from packet loss occurring in communication between two nodes in the routing path – it does not account for timeouts or retry attempts made by the source node. In order to determine the PRR between any two nodes that lie in the path selected for routing data between the source node and the $SINK$ the following equation is used:

$$PRR_{BA} = \frac{DN_R(A)}{DN_S(B)} \tag{9}$$

$DN_R(A)$ and $DN_S(B)$ imply the number of distinct packets received by node A, and transmitted by node B, respectively. To measure the energy efficiency of a WSN, the DE between two nodes AB in the network can be calculated as:

$$DE_{AB} = \frac{DN_R(A)}{DN_S(A)} \tag{10}$$

TABLE I
STATISTICAL SUMMARY ON PRR OF THE THREE NODES

Node ID	Mean	Median	Std. Dev.
1	72.02	71.43	16.45
2	50.94	50	38.31
3	72.35	100	35.44

After analyzing the data we observed that although the same reading and data packet was used for each of the nodes, yet the mean varied considerably between the three nodes (Table I). The difference in reading can be attributed to the variance in weather conditions enforced in the simulation, which affects the antenna gain.

IV. SNIPER CLUSTERING

A WSN must be organized into a set of clusters so as to manage the available resources in the most efficient manner possible. However, there are many different techniques that may be involved in the clustering process, and hence, it is essential to formulate a method of assessing the quality of clustering. As per the Socionomic principles, we have adopted the Small-World model to develop a method for calculating the Clustering Coefficient – a measure of the degree to which the Cluster Head is connected. The connectivity may be determined either as global or local, where [9]:

- Global clustering is an indication of the clustering of the whole network itself,
- Local clustering is the fraction of pairs of neighbours of a node that are themselves neighbours.

Therefore, for a neighbourhood graph G, consisting of a set of vertices V, and edges E, the graph can be represented as $G = \{V, E\}$. An edge connecting nodes i and j is represented as e_{ij}. Therefore, the neighbourhood of the vertex v_i can be represented as:

$$N_i = \{v_j : e_{ij} \in E \Lambda e_{ji} \in E\} \tag{11}$$

The degree k_i of node i is the number of vertices N_i in its neighbourhood. The number of possible links between the neighbours of node i is given by:

$$K = \frac{k_i(k_i - 1)}{2} \tag{12}$$

Therefore, the local clustering coefficient C_i is given by:

$$C_i = \frac{|\{e_{jk} \in E(G) : e_{ij} \in E(G) \Lambda e_{jk} \in E(G)\}|}{K} \tag{13}$$

And, the global clustering coefficient can thus be calculated as a sum of the local clustering coefficients, as follows:

$$C = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{14}$$

In order to study the variation in cluster formation, for this experiment, a comparative study of the cluster formation has been performed between the SNIPER algorithm and the HEED algorithm (Figure 2 and Figure 3).

The SNIPER algorithm being based on a social framework, exhibits a relatively more complex set of interconnections

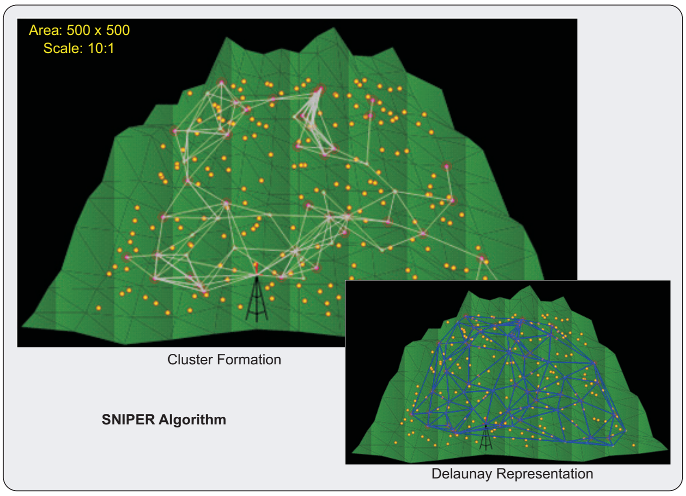


Fig. 2. SNIPER cluster formation overlayed with its Delaunay Graph.

between the nodes, than is evident for HEED. In order to assess the strength, efficiency, and collaboration of these algorithms, a hierarchical clustering coefficient has been used.

One of the key assumptions made in this experiment made is that while SNIPER is capable of altering its communication range, HEED constantly uses the maximum possible range. Therefore, to inter-link all the Cluster Heads in the network separated by their geodesic distance, d, the number of hops required for SNIPER will be greater than HEED. Since, the connectivity of the network also depends on the number of inter-relationships for each node, the following elements must also be considered:

- Node Degree (n_d),
- Hierarchical Degree (h_d)
- Edges (e),
- Divergence (D),
- Clustering Coefficient (CC).

The method adopted to calculate the Clustering Coefficient involved the following:

- Determining a count of the number of neighbours the reference node possesses at each successive hop count, as applicable,
- The hierarchical degree of the reference node at each hop count,
- The divergence measurement of the reference node at each hop count, and finally,
- The Clustering Coefficient itself.

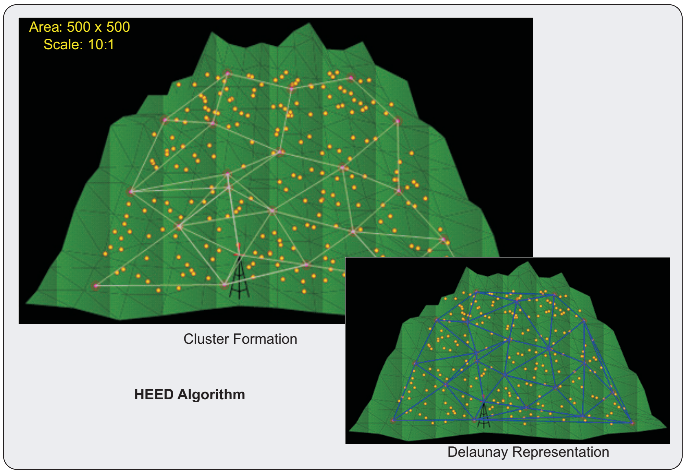


Fig. 3. HEED cluster formation overlayed with its Delaunay Graph.

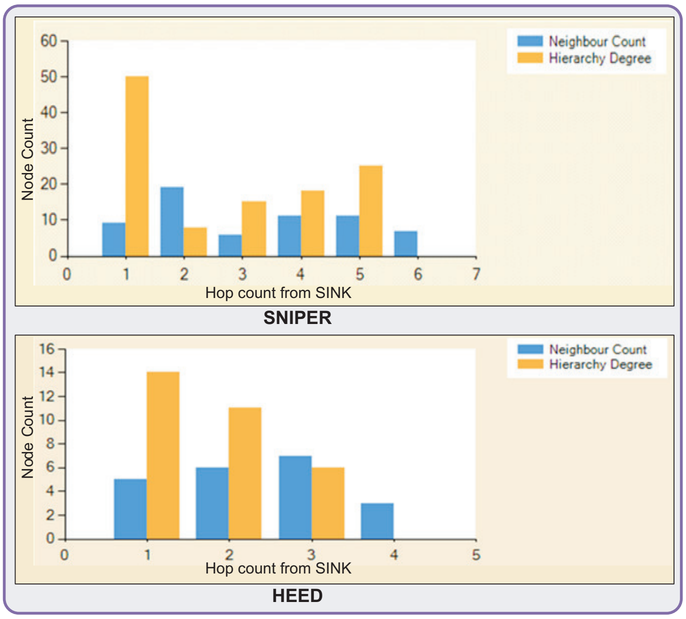


Fig. 4. Cluster formation analysis of SNIPER and HEED.

A study of the plot of the Cluster Formation Analysis and Divergence analysis revealed that although the number of hop counts for SNIPER was greater than HEED, it still offered a better Hierarchy Degree, implying a better level of inter-node connectivity across the network. Similarly, a study of the Divergence Analysis also showed a higher number of connectivity links for SNIPER (Figure 4 and Figure 5).

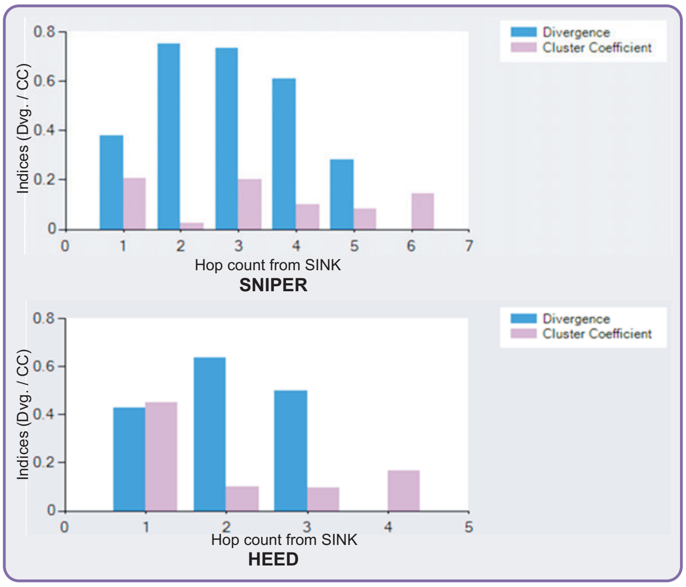


Fig. 5. Comparison of divergence and cluster coefficient between SNIPER and HEED.

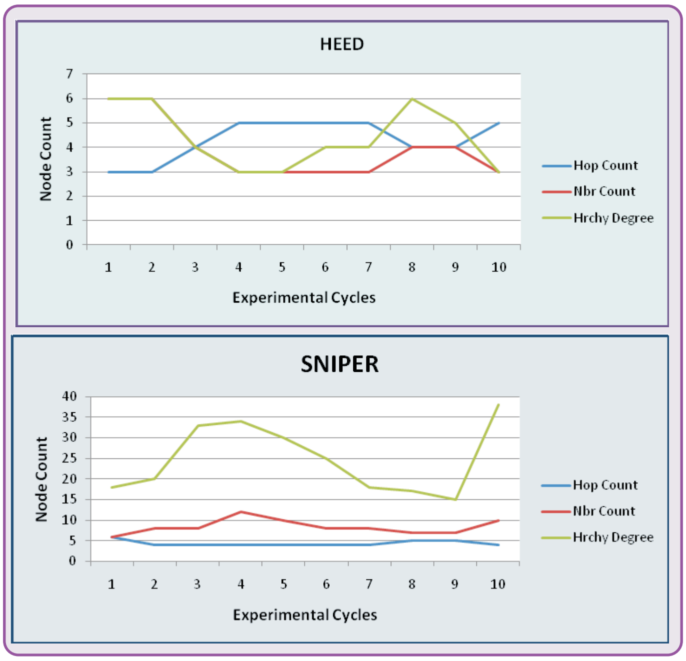
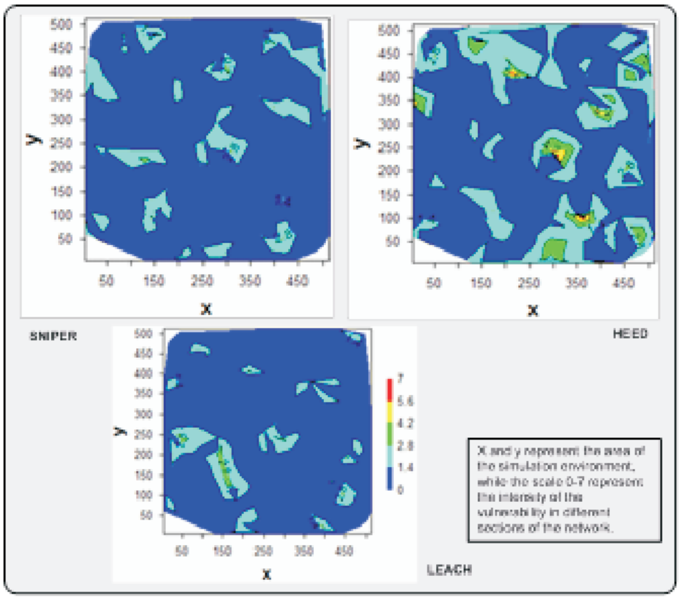


Fig. 6. A graphical comparison of the readings.

Also, the higher number of hop count for SNIPER is further evidence that it offers a greater depth in the links originating from the Sink, and therefore much better network coverage. At the same time, the load on individual Cluster Heads is also reduced because of the fewer number of member nodes. The average connectivity level (or load) of the Cluster Heads is evidenced by the lower Divergence factor for SNIPER, implying more uniform resource utilization across the network. Therefore, it not only ensures better connectivity between the nodes, but also a higher coverage across the network.



X and y represent the area of the simulation environment, while the scale 0-7 represent the intensity of the vulnerability in different sections of the network.

Fig. 7. Vulnerability analysis of different algorithms in cluster formation.

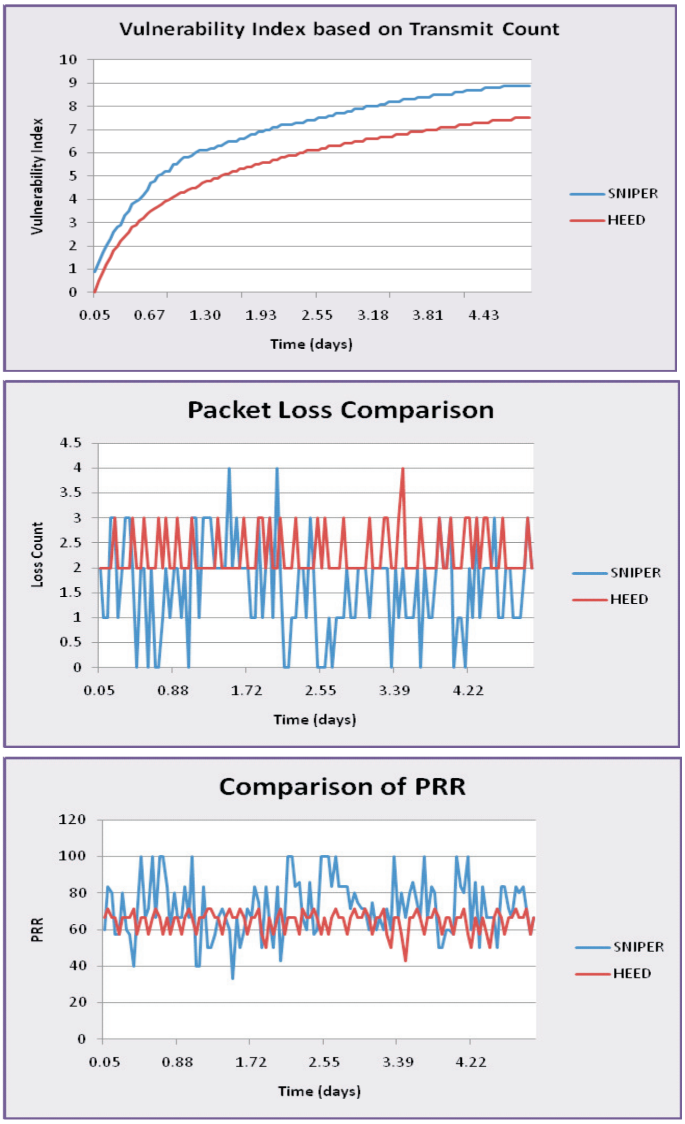


Fig. 8. Vulnerability analysis using node degree and transmit count as the factor.

It can be also observed that there is indeed a direct relationship between the hop count, average hierarchy degree, and average neighbour count for both HEED and SNIPER (Figure 6). The average hierarchy degree is inversely proportional to the maximum hop count. Although the pattern is more obvious in case of HEED, yet it is also evident for SNIPER, suggesting the presence of this relationship is subject to parameters for each clustering technique.

Therefore, the Clustering Coefficient serves as a good measure for analysing two different clustering techniques sharing a common deployment of nodes. However, yet another noble concept also capable of assessing the quality of network cluster formation is the Vulnerability Coefficient. The technique is not only useful in assessing the vulnerability of a WSN, but can also be incorporated in making critical routing decisions at Cluster Heads.

Typically, during cluster formation and data routing certain sensor nodes in a network are exercised more than others by virtue of their location or placement. As a result, over time re-

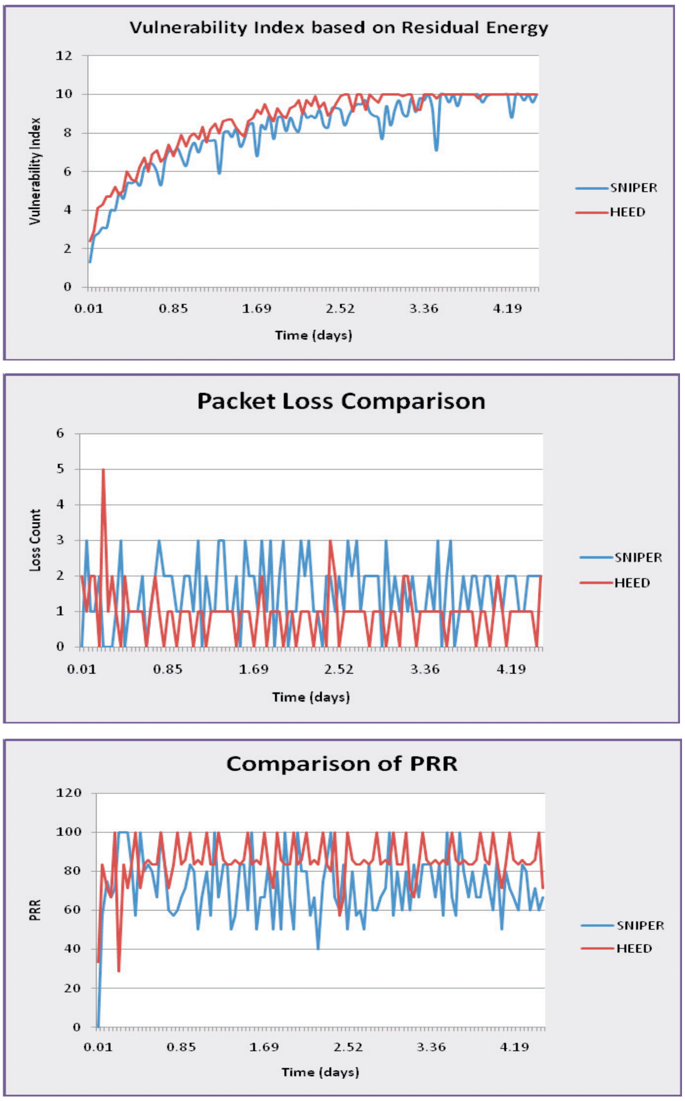


Fig. 9. Vulnerability analysis using residual energy as factor.

source vacuums tend to develop among the over-exercised set of sensor nodes, thereby affecting inter-node communication. In other words, the stability of a WSN is inversely proportional to its vulnerability coefficient, and directly proportional to the lifetime of the network.

In order to calculate the vulnerability coefficient, the WSN needs to be analysed in two stages. In the first stage, the vulnerability of the network is determined following the cluster formation, and in the second stage the effect of data routing was studied on the whole network. Both stages were repeated for the SNIPER, LEACH, and HEED protocols.

For the purpose of this study, the vulnerability is essentially the square root of the product of the respective level of threat and risk that a node faces, represented by:

$$v_{ij} = \sqrt{t_{ij} \times r_{ij}} \tag{15}$$

The level of threat faced by a node is determined by:

$$t_{ij} = \frac{d_{ij}^2 - 1}{d_{ij}^4} \times F \tag{16}$$

Where, t_{ij} is the threat level, d_{ij} is the distance, and F is the vulnerability factor between node i and j.

Similarly, the level of risk a node is subject to at discrete points in time is given by:

$$r_{ij} = \frac{d_{ij}^2 + (7 \times d_{ij}) + 3}{d_{ij}^2} \times 5 \tag{17}$$

Where, r_{ij} is the risk level faced by node i with respect to node j, and d_{ij} is the distance between node i and j.

Cluster formation is typically the most resource intensive operation in a WSN, and hence, the efficiency of a network is dependent on the number of packets that get exchanged during this operation. The surface plot presented in Figure 7 is reflective of the vulnerability faced by 250 nodes in the network. As can be observed from the plot, the SNIPER algorithm by virtue of its pseudo-election scheme of cluster formation based on Socionomics shows the least impact on sensor node distribution.

Similarly, to study the effect of data routing on node vulnerability, the impact of 1000 data transmissions on network vulnerability was analysed based on network vulnerability, packet loss, and PRR. The results of the experiment presented in Figure 8 and Figure 9, show that the network was subjected to a packet loss of 10%. A statistical representation of the data acquired on vulnerability analysis of data routing using SNIPER and HEED further demonstrate the contrast in the vulnerability index.

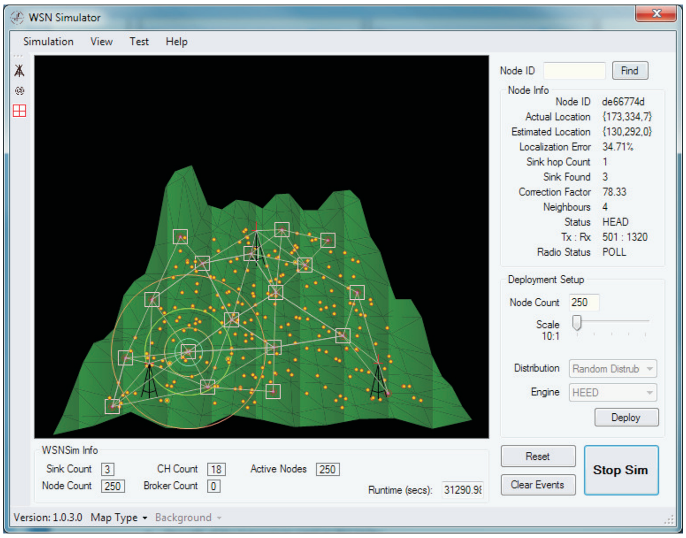
Therefore, the stability of a WSN can be defined by the strength of each individual edge in the directed graph $G = \{V, E\}$. As the vulnerability of a node rises, the cost of maintaining the link with its neighbouring nodes separated by the geodesic distance, d, also increase.

V. SNIPER Localization

While a number of optimizations have been introduced in the data routing for WSNs. However, information collected from the network by a Sink is of little value if the source of certain events cannot be determined. Although, the simplest technique for including location information is by using a GPS, yet, it is also very resource intensive and hence avoided. Other techniques such as the centroid and triangulation algorithm serve as alternate means of estimating a node location within a degree of localization error. The precision of a localization algorithm is typically affected by:

- Number of neighbours,
- Number of anchor points,
- Density of the network,
- Strength of the transceiver device on nodes,
- Packet loss,
- Incorrect calculation of geodesic distance between nodes, and
- Localization algorithm.

In this paper, we introduce a new localization technique called the MP-RSSI algorithm. Drawing from the concepts presented in the DV-Hop algorithm [10], the MP-RSSI algorithm introduces significant improvements by using a single phase transmission from the anchor nodes, thus substantially



Node ID

Node Info

Node ID de66774d
Actual Location {173,334,7}
Estimated Location {130,292,0}
Localization Error 34.71%
Sink hop Count 1
Sink Found 3
Correction Factor 78.33
Neighbours 4
Status HEAD
Tx : Rx 501 : 1320
Radio Status POLL

Deployment Setup

Node Count 250
Scale 10:1
Distribution Random Distrub
Engine HEED

WSNSim Info

Sink Count 3
CH Count 18
Active Nodes 250
Node Count 250
Broker Count 0
Runtime (secs): 31290.9£

Version: 1.0.3.0

Fig. 10. Sample localization using HEED.

improving the energy requirements. By using this technique, the sensor nodes do not need to rely on the anchor nodes being aware of each other, and instead rely on the distance estimation between neighbouring nodes by determining the hop count from respective anchor nodes. The distance between each node is determined based on the RSSI of the packets received from neighbouring nodes using the following equation:

$$d = \frac{\lambda}{4 \times \pi} \times \sqrt{\frac{P_{tx}(mW)}{P_{rx}(mW)}} \qquad (18)$$

It is assumed that nodes in a WSN are likely to be scattered in a random manner, and routes from the Sink to a particular node may not necessarily be a straight line. Therefore, the MP-RSSI algorithm uses a scaling factor whereby a confidence level metric is applied in determining the distance between each node, calculated using the following equation:

$$\vartheta = \frac{1}{1 + (A \times \tau) - (B \times \tau^2) + (C \times \tau^2)} \qquad (19)$$

Where, τ is the hop count, and A, B and C are constants used to control the level of impact.

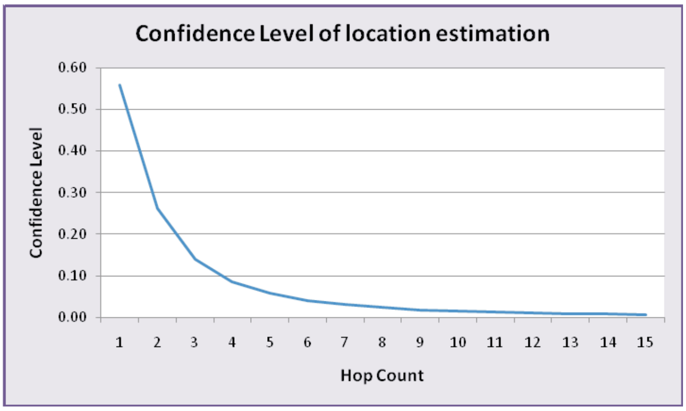
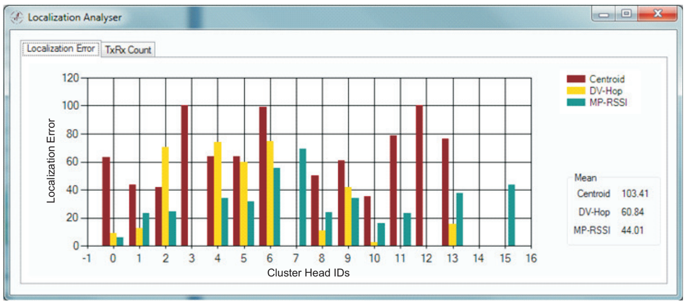


Fig. 11. Confidence level trend line for MP-RSSI.



Mean

Centroid 103.41
DV-Hop 60.84
MP-RSSI 44.01

Fig. 12. Localisation of 250 nodes using 4 anchor points.

The distribution of the confidence level at each hop count is calculated by setting the constants A, B, and C, to be 0.168, 0.134, and 0.755, respectively. A plot of the confidence level for Eqn. 15 shows that with increasing distance the confidence level steadily drops (Figure 11).

In our experiments, the MP-RSSI algorithm was compared against the DV-Hop and Centroid algorithms, and was resulted in a localization error of approximately 7%, which is similar to the DV-Hop algorithm (Figure 12). The results were further ascertained through further experiments involving a variance in the number of nodes and anchor points and in each case, MP-RSSI and DV-Hop presented comparable levels of accuracy. The localization error, δ, was calculated using the following equation:

$$\delta = \frac{\sqrt{(x_{est} - x)^2 + (y_{est} - y)^2 + (z_{est} - z)^2}}{\sqrt{x^2 + y^2 + z^2}} \qquad (20)$$

Therefore, it can be concluded that while the MP-RSSI algorithm does not necessarily improve the precision in localization, it does improve the resource usage, and is hence an improvement on existing techniques.

VI. Conclusion and Future Work

In this paper, we have used the Socionomic model to present a few novel concepts in executing some of the fundamental operations of a WSN, namely, clustering, routing, and localization. In each case, we have not only formulated a new method for calculating the respective coefficients, but also showed that the SNIPER algorithm is able to perform better, if not similar to existing techniques, such as LEACH and HEED.

The socioeconomic framework has allowed us to attempt an efficient approach towards clustering and routing in a WSN. The proposed algorithm is both flexible and dynamic and can thus be adopted in a range of different domains. In order to justify the application of the model, we used a specifically designed simulation platform to assess its functionality in a range of different scenarios. In the experiments that were conducted we demonstrated the capabilities of the SNIPER protocol and also provided results that are comparable to other known algorithms, such as, LEACH, HEED and GAF.

The future work will involve the adaptation of the simulation framework onto real sensor nodes and thus realize the potential of the model in an actual WSN application. There are several potential areas of application of the SNIPER framework - primarily in areas that rely on energy efficiency and dynamic route calculation. Some of the potential areas of application include the deployment of sensor nodes in remote forests, near harbours to study the level of tides and aid in navigating ships, tunnel and bridge monitoring. We have already observed that the SNIPER algorithm scales well in large sensor networks, and thus brings forth optimization in computing and resource requirements.

REFERENCES

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proceedings of IEEE Computer and Communications Societies*, vol. 2, pp. 775–784, March 2000.

[2] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, January 2001, p. 10.

[3] H. C. Chu and R.-H. Jan, "A GPS-Less Positioning Method for Sensor Networks," *The 1st International Workshop on Distributed, Parallel and Network Applications*, vol. 2, pp. 629–633, July 2005.

[4] S. Sinha, "Towards a Socionomic Framework for Collaborative Data Routing in Wireless Sensor Network," in *Doctorate Thesis*. Sydney: University of Technology, 2010.

[5] D. Niculescu and B. Nath, "Ad-Hoc Positioning System (APS)," *Global Telecommunications Conference*, vol. 5, pp. 2926–2931, November 2001.

[6] W. Kets, *The Minority Game: An Economics Perspective*. SSRN eLibrary, 2007.

[7] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Scheduling Sleeping Nodes in High Density Cluster-Based Sensor Networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825–835, 2005.

[8] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," *22th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, 2003.

[9] A. K. Naimzada, S. Stefani, and A. Torriero, *Networks, Topology and Dynamics: Theory and Applications to Economics and Social Systems*. Berlin: Springer-Verlag, 2009.

[10] Y.-H. Gau, H.-C. Chu, and R.-H. Jan, *A Weighted Multilateration Positioning Method for Wireless Sensor Networks*. Taiwan: National Chiao Tung University, 2008.

[11] M. Friedman, *Essays in Positive Economics*. University of Chicago Press, 1996.

[12] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, 2001.

[13] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network," in *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, 2003.