Data Fusion Algorithms in Cluster-based Wireless Sensor Networks Using Fuzzy Logic Theory

WEILIAN SU
Naval Postgraduate School
Department of Electrical Engineering
833 Dyer Road, Monterey, CA 93943-5121
USA

THEODOROS C. BOUGIOUKLIS
Naval Postgraduate School
Department of Electrical Engineering
833 Dyer Road, Monterey, CA 93943-5121
USA

Abstract: Data Fusion in wireless sensor networks (WSNs) can improve the performance of a network by eliminating redundancy and power consumption, ensuring fault-tolerance between sensors, and managing effectively the available communication bandwidth between network components. In this paper, we develop a data fusion algorithm that combines the cluster-based design of WSNs using fuzzy logic methods. Our results show that the algorithm eliminates redundant sensor reports.

Key–Words: Data fusion, redundancy, fuzzy logic, aggregation, throughput

1 Introduction

Wireless sensor networks (WSNs) consist of large number of energy constrained sensor nodes, randomly distributed over a geographical region, to detect a physical phenomenon or event. The operation of a WSN is composed of sensing, processing, and communication tasks executed from sensor nodes that operate under conditions characterized by low signal-to-noise ratio, interference and multipath effects [1]. Due to environmental characteristics, limited power and processing capabilities of WSNs, it is essential to find techniques that improve the flow of information in the network. Data aggregation or fusion can de used to minimize the amount of information flowing and the energy spent during sensing, processing, and communication operations in the network [2]. Data aggregation or fusion is a process of combining data from multiple unreliable sources (sensor nodes) to extract useful and reliable information.

Many data fusion algorithms have been proposed. Dasgupta [3] presents a tree-based aggregation algorithm. A tree is constructed with sink node as a root and data can be fused at intermediate sensor nodes (edge of tree). The algorithm assigns a data gathering schedule in which collected data from all sensor nodes are fused and forwarded to the base station (user). The algorithm lacks speed since the running time of the tree building process is $O(n^3)$, where n is the number of sensor nodes in the network. SPIN (sensor protocols for routing information via negotiation) [4] [5] uses a type of negotiation between sensor nodes to reduce the amount of duplicate data flowing in the network. The algorithm uses application-based knowledge in making routing decision (cross-layer design). Such techniques require complex hardware implementation. Heidemann et al. [6] proposes a directed diffusion mechanism in which a user's queries are forwarded to an application aware sensor node based on a least-cost algorithm. Then data generated from sensor nodes are forwarded to the user following the reverse least-cost path. The mechanism uses filters to compress correlated data from different sensor nodes instead of combining correlated or uncorrelated data. *Cushion* [7] is an adaptive data aggregation algorithm, in which the sink node controls the message overhead of sensor nodes by measuring their reliability. Flow of information in the network is based on control messages generated by the sink node based on redundancy level p (reliability measure) of sensor nodes. The redundancy level depends on the number of received data packets at the sink node. Control messages can cause performance degradation due to congestion in cases where WSNs consist of large number of sensor nodes. Authors in [8] propose a data fusion scheme that distinguish collected data in emerged and usual sensed data. Collected data at each sensor node are stored for a time period and only the emerged or the changed information of usual sensed data is forwarded to the sink node. The algorithm consumes extra memory space for storing sensed data and processing power for each sensor node.

In this paper, we propose a data fusion algorithm that is based on fuzzy logic methods to reduce traffic and enhance the performance of the sensor networks.

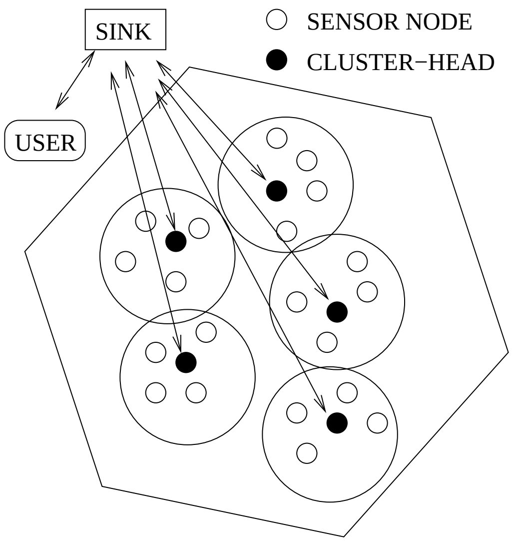

Figure 1: Clustering in WSNs.

Fuzzy logic methods are capable of fusing uncertain data from multiple sensor nodes to improve the quality of information. They require less computational power than conventional mathematical computational methods such as addition, substraction, multiplication and division. In addition, only few data samples are required in order to extract final accurate result. Finally, they can be effectively manipulated since they use human language to describe problems [9]. The rest of the paper is organized as follows. In Section 2, we present the data fusion algorithm. Simulation setup and performance of the proposed algorithm are described in Section 3. Lastly, we conclude in Section 4.

2 Data Fusion Algorithm

2.1 Clustering in WSNs

One practical design scheme in WSNs is clustering, in which sensors form clusters based on some predefined criteria [10]. For each cluster, a sensor node is chosen to be a cluster-head and is responsible to forward data to a base station (sink node) as shown in Figure 1.

Clustering divides a large-scale coverage area of a WSN to a number of clusters, thus transmission range of each sensor node in each cluster is much smaller than that before clustering [12]. The amount of power required to send data from a sensor node to a cluster-head is much smaller than the amount required if the same sensor sends data directly to a base station. In addition, clustering allows load balancing in each cluster which can improve the performance of the network [13].

Rule	Rule description using crisp values
1	if x is $A3$ or y is $B1$ then z is $C1$
2	if x is $A2$ and y is $B2$ then z is $C2$
3	if x is $A1$ then z is $C3$

Table 1: Rules for a Two-input One-output. problem

2.2 Fuzzy Logic Inference Methods

In general, fuzzy logic is a multivalued logic, by which intermediate values can be defined using expressions such as true/false, high/low, below/above, etc [14]. The most common fuzzy logic inferences are the Mamdani and Tsukamoto-Sugeno methods. Both fuzzy logic Mamdani and Tsukamoto inference methods used by the proposed data fusion algorithm are completed in four phases: fuzzification, rule evaluation, combination or aggregation of rules, and deffuzification [15]. For example, consider a simple two-input and one-output inference system, and assume that the rules describe the system are as shown in Table 1, where x and y represent the input values and z the output value. The two input values are the distance of a sensor node from a sink node and its available operational power. The output value z describes the reliability of the sensor's report. Each input is fuzzified using a triangular or trapezoidal function. The center and the width of each membership function depend on the range of crisp input values. The first step in the Mamdani fuzzy inference method is to assign a degree of membership for each input value to the appropriate defined fuzzy sets. The two intersection points assumed crisp input values of $x = 62m$ and $y = 34m$ are shown in Figures 2 and 3.

The next step is the rule evaluation, where the fuzzified inputs are applied according to certain appropriate rules. Usually, in cases where a fuzzy rule has more than one conditional element (antecedent), an AND (minimum) or OR (maximum) operator is used to estimate a number that describes the result after the rule evaluation, as shown in Figure 4. The union (maximum) of fuzzified inputs is picked using operator OR , and the intersection (minimum) is picked using operator AND. In cases where the input is zero, the resulting value is zero.

The third step of the Mamdani fuzzy inference method is the aggregation of all outputs. During aggregation, the outputs of each rule are combined to form a new fuzzy set as shown in Figure 5.

The final step is the defuzzification process, by which the aggregated new fuzzy set is converted to a number. The method used to implement this conversion is called the centroid technique, which is presented in Figure 6. The centroid method tries to deter-

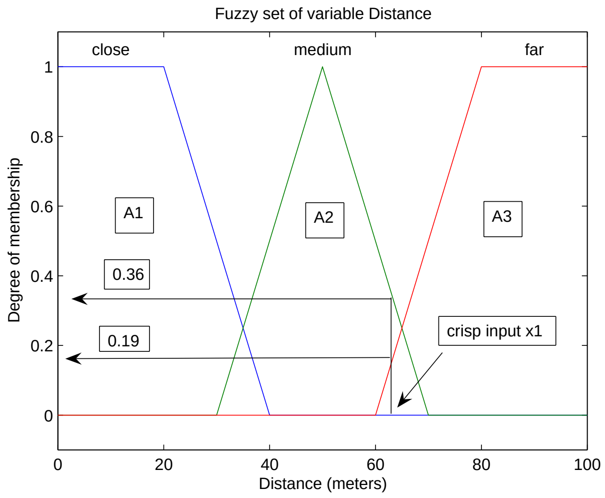

Figure 2: Fuzzification assignment for the Distance input variable.

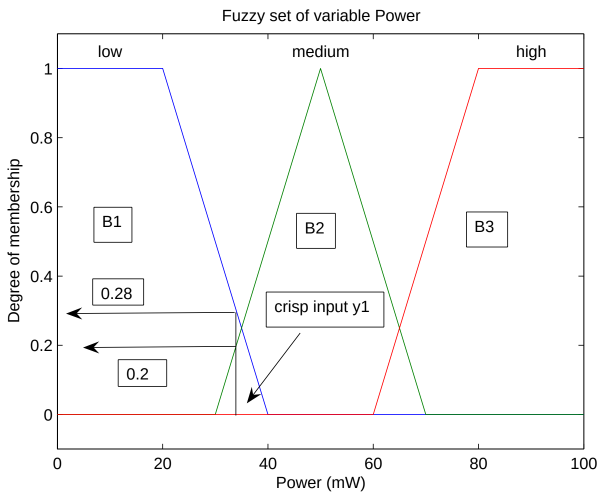

Figure 3: Fuzzification assignment for the Power input variable.

mine the point at which a vertical line slices the combined set into two equal parts. The center of gravity in Figure 6 is calculated as 31.67.

The steps are the same for the Tsukamoto inference fuzzy method, but there are two basic differences compared to the Mamdani fuzzzy inference method. One difference is that in Tsukamoto fuzzy method rule outputs represented as single tones placed at a single particular point for which the membership function is maximum (20,50,70) as shown in Figure 7. The second difference is that a weighted average method is applied to calculate the defuzzified value as shown in Figure 8.

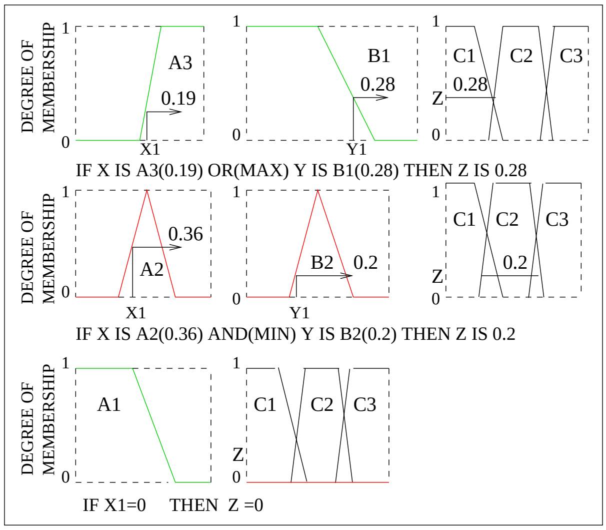

Figure 4: Mamdani Rule Evaluation Process for crisp $x1 = 62m$ and $y = 34mW$.

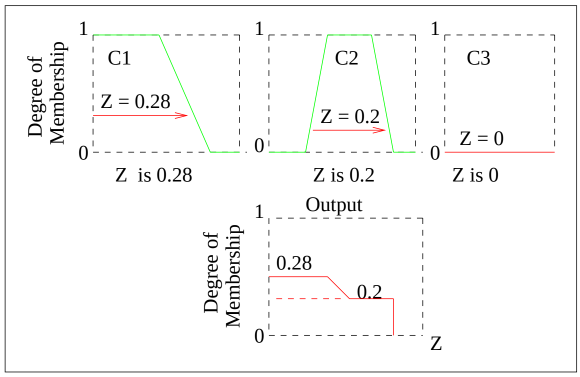

Figure 5: Aggregation of rule outputs.

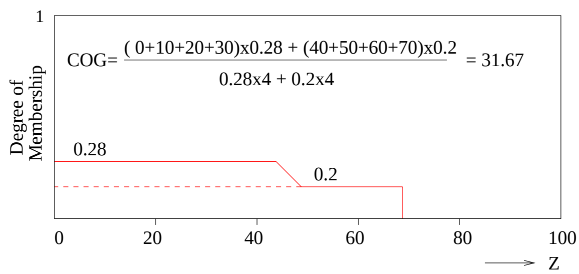

Figure 6: Mamdani Defuzzification Process. For the crisp values of $62m$ and $34mW$ the center of gravity is computed as 31.67.

2.3 Proposed Algorithm

The proposed algorithm is based on the fuzzy logic inference methods described in previous section. The

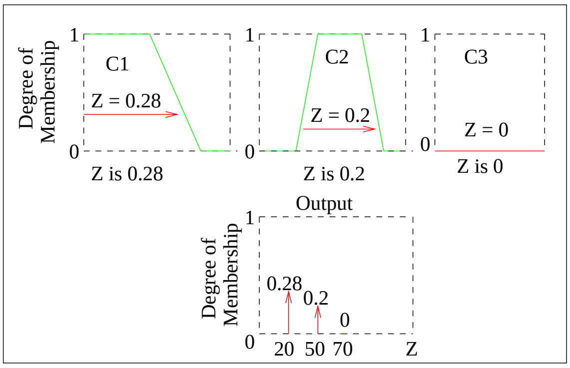

Figure 7: Aggregation of rule outputs.

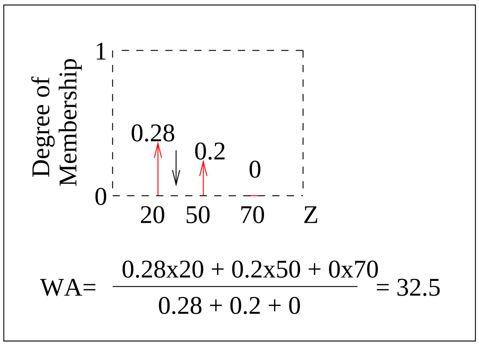

Figure 8: Tsukamoto Defuzzification Process.

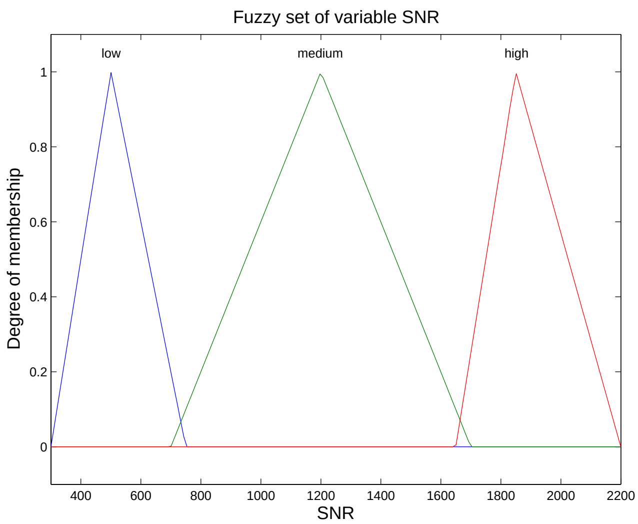

Figure 9: Fuzzy set assignment for the SNR variable.

input fuzzy values in our fuzzy inference system are the distance of each sensor node from the sink node and the quality of received signal-to-noise ratio (SNR) at the sink node. The fuzzy input *SNR* is fuzzified using a triangular function as shown in Figure 9, where

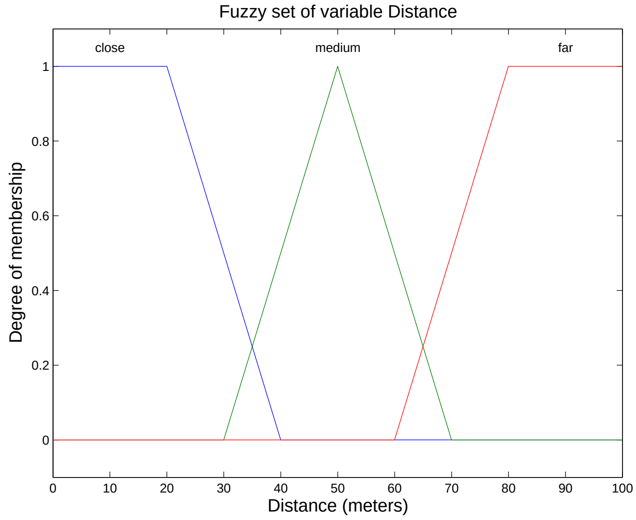

Figure 10: Fuzzy set assignment for the Distance variable.

the *low*, *medium*, and *high* components represent the magnitude of participation for that input. The fuzzy input *Distance* is fuzzified using a trapezoidal and triangular functions as shown in 10, where the components *close*, *medium*, and *far* represent the magnitude of participation for that input.

Both input variables are used to assign a weight factor to each sensor node that detects an event. All input fuzzy values are stored using array vectors. At the formed array vectors, the algorithm tries to eliminate common observations by searching for same equal reported values coming from the same sensor node. The center and width used in triangular or trapezoidal functions during fuzzification phase is chosen from numerical values such as minimum, maximum, mean, and standard deviation of input fuzzy values. The fuzzified inputs are applied to an appropriate set of rules. The number of constructed rules is k^n, where $n = 2$ is the two input variables and $k = 3$ represents the three different terms (low, medium, high) for each variable. Table 2 includes the nine applied rules in the algorithm. The expected fuzzy output described by the fuzzy variable *State* is shown in Figure 11, where the *LL*, *LM*, *LH*, *MM*, *MH*, and *HL*, *HM*, and *HH* represent the magnitude of participation for that output.

In the final step (4), defuzzification is applied to the aggregated output using the center-of-gravity method and the weighted average, respectively, as described in previous subsection, resulting in a single number output, which defines a weight factor.

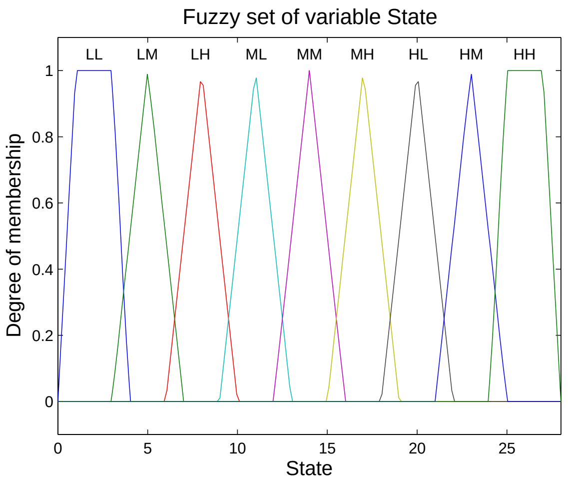

Figure 11: Fuzzy set of output variable State.

RULE	SNR	DISTANCE	State
1	low	close	StateLowLow
2	low	medium	StateLowMedium
3	low	far	StateLowHigh
4	medium	low	StateMediumLow
5	medim	medium	StateMediumMedium
6	medium	far	StateMedimHigh
7	high	low	StateHighLow
8	high	medium	StateHighMedium
9	high	far	StateHighHigh

Table 2: The constructed fuzzy rules used for simulation.

3 Performance Evaluation

3.1 Simulation Setup

For our simulations, we use J-Sim [16]. J-Sim uses three types of nodes: target nodes (generate events and phenomena), sensor nodes (detect events), and sink nodes (process the collected information). An example of a network topology is shown in Figure 12, where there are six target nodes and one sink node. The task of the target nodes is to detect a physical event (phenomenon) and report it to the sink node (sensor or user). The physical phenomenon is an earthquake event started at a predefined location and characterized with an initial magnitude. The magnitude of the earthquake is decreased following an exponential distribution with a predefined attenuation factor of $\lambda = 0.0015$ richter/meter. When target nodes detect the phenomenon, they send data reports (packets) to the sink node. The data are generated using an *on/off* scheme, with *on* (5 and 10 seconds) and *off* (15 and 20 seconds) periods of time. During the *on* and *off* periods, packets are sent at a fixed rate or are not sent, respectively.

In the following subsection we evaluate the proposed algorithm for both Mamdani and Tsukamoto fuzzy methods using two different wireless topologies consisting of six and fifty target nodes.

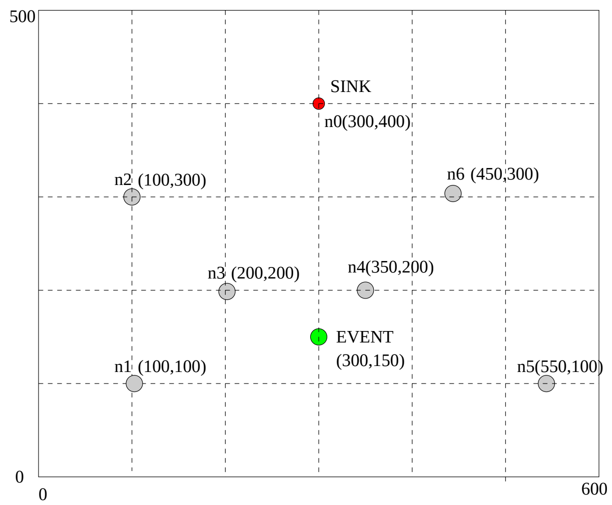

Figure 12: Example of a topology for a wireless sensor network.

3.2 Evaluation of proposed scheme

Our proposed algorithm is applied to two different wireless topologies, each of which has different number of target nodes (6 and 50) and different values for the initial magnitude of the earthquake event (seven and eight Richter respectively). For each topology, the two techniques are compared using the following characteristic values: (1) the reported magnitude value based on the highest and weighted defuzzified value, and (2) the error between the initial and the estimated magnitude value based on the highest and weighted average defuzzified value. At every *epoch*, which is five minutes, the algorithm executes data fusion using the above two fuzzy methods if there are data to be fused, otherwise it waits for the next epoch, and so forth.

Figure 13 shows the reported magnitude while Figure 14 shows the reported magnitude error for the first scenario (6 target nodes) using the highest and weighted average value.

Comparing the above figures we can make the following observations:

- The mean of magnitude error for Mamdani method using the highest and weighted average defuzzified value is smaller than Tsukamoto method.
- The deviation of the magnitude error for Mamdani method using the highest and weighted average defuzzified value is smaller than Tsukamoto method.

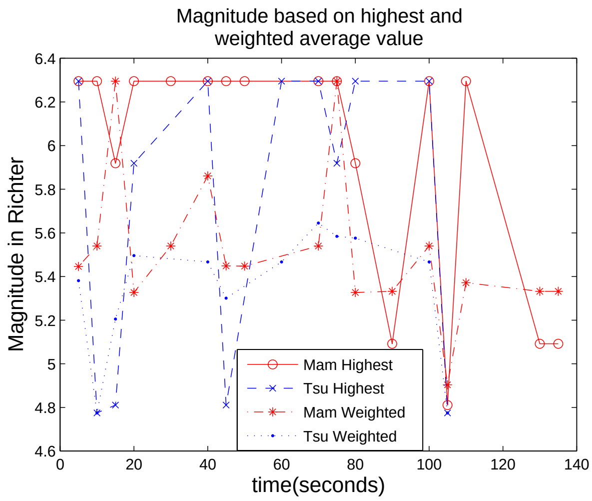

Figure 13: Reported magnitude using the highest and weighted average value for Mamdani and Tsukamoto techniques for the six target node scenario.

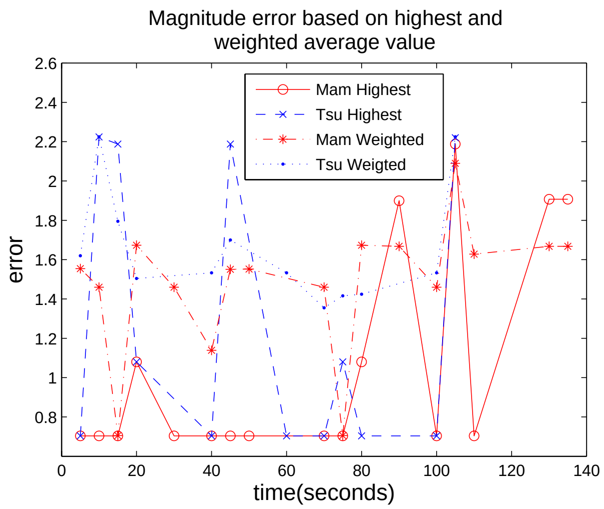

Figure 14: Magnitude error using the highest and weighted average value for Mamdani and Tsukamoto techniques for the six target node scenario.

- The differences in the mean and deviation of magnitude error are smaller in case of using the highest defuzzified value.
- The Mamdani method gives more accurate result compared to Tsukamoto method using either highest or weighted average defuzzified values.

Next, we want to explore the impact of increasing the number of sensor nodes on the fusion techniques. Figures 15 and 16 show the reported magnitude and reported magnitude error for the second scenario (50

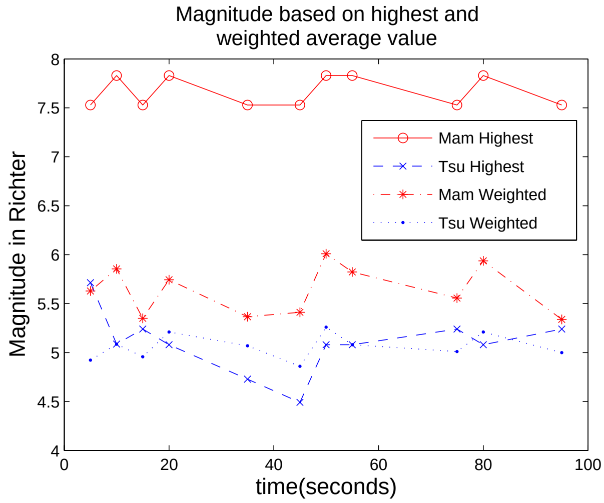

Figure 15: Reported magnitude using the highest and weighted average value value for Mamdani and Tsukamoto techniques for the fifty target node scenario.

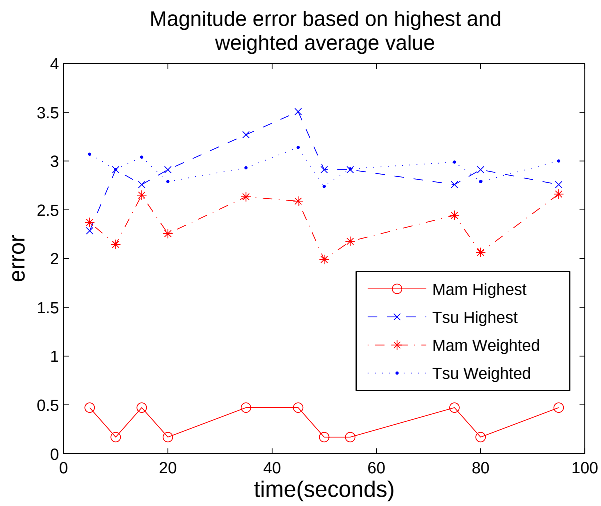

Figure 16: Magnitude error using the highest and weighted average value for Mamdani and Tsukamoto techniques for the fifty target node scenario.

target nodes) using the highest and weighted average value, respectively.

In the second scenario, the observations confirm the superiority of the Mamdani fusion method over the Tsukamoto data fusion approach using the highest or the weighted average defuzzified value. In addition, the following conclusions extracted based on the algorithm can be made:

- The deviations for the magnitude report, and the magnitude error are smaller than the correspond-

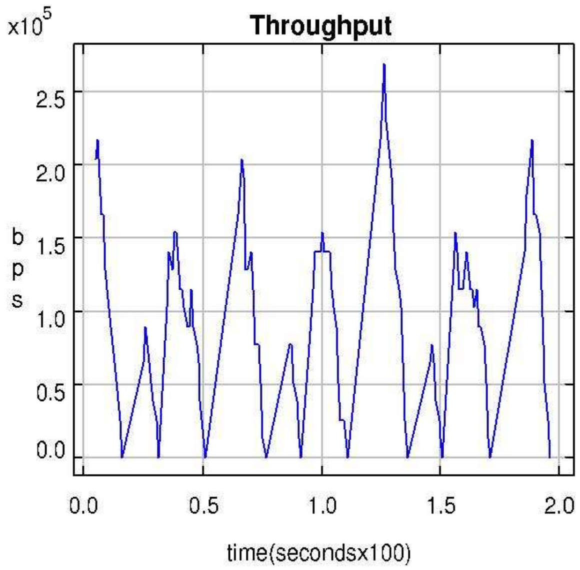

Figure 17: Aggregated throughput(bps) at the sink node.

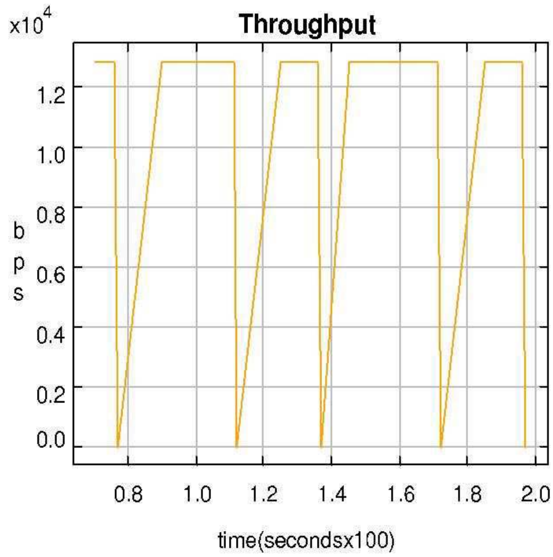

Figure 18: Throughput at sink node after data fusion for the six node scenario.

ing deviation values of the first scenario.

- There is a significant difference between all the reported values and the corresponding errors using the highest and the weighted average defuzzified value, especially for the Mamdani data fusion technique.

- Differences between all the reported values and the corresponding errors using the highest and the weighted average defuzzified value are not significant for the Tsukamoto data fusion technique. Although, they still have close deviations.

- For both scenarios, the two methods using the weighted average defuzzified method give the least accurate results and the maximum errors.

Based on the previous observations, we apply our data fusion algorithm using Tsukamoto inference method in both topology scenarios (6 and 50 target nodes). Figure 17 shows the aggregated throughput at the sink node (n0) in bits per second (bps) for the six target nodes. Figure 18 shows the throughput after data fusion from the sink node to the next-hop component (user or a central sensor node), assuming a constant payload during each epoch for the first scenario topology (6 target nodes). Note the substantial improvement in performance as compared to Figure 17. There is a traffic reduction of up to 94.8% and an average traffic reduction of 92%.

The aggregated throughput at the sink for the second scenario topology (50 target nodes) is shown in Figure 19, and Figure 20 shows the throughput after data fusion at the the sink node. By comparing Figures 19 and 20, we can observe that there is a traffic reduction of up to 98.7% and an average traffic reduction of 97%.

4 Conclusion

In this paper, we proposed a new data fusion algorithm for wireless sensor networks (WSNs). The proposed algorithm is based on fuzzy logic theory, and it applies two different fuzzy logic inference techniques (Mamdani and Tsukamoto). The algorithm was implemented and tested for two different wireless sensor topologies. The simulation environment considered in this work was an earthquake phenomenon, and the two proposed methods were applied to fuse data that were generated from the target nodes. Results show that the Mamdani method gives better results than the Tsukamoto approach for both wireless network topologies, and can reduce traffic by up to 94.8% and 98.7%. As a result, it can be applied to WSNs to provide optimal data fusion and ensure maximum sensor lifetime and minimum time delay.

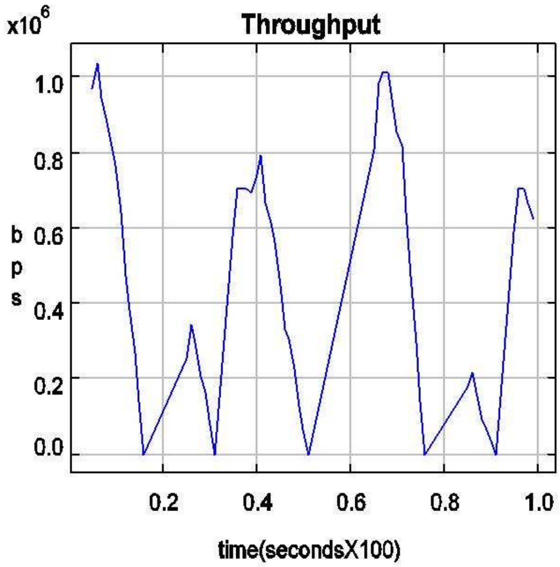

Figure 19: Aggregated throughput(bps) at the sink node before data fusion for the fifty node scenario.

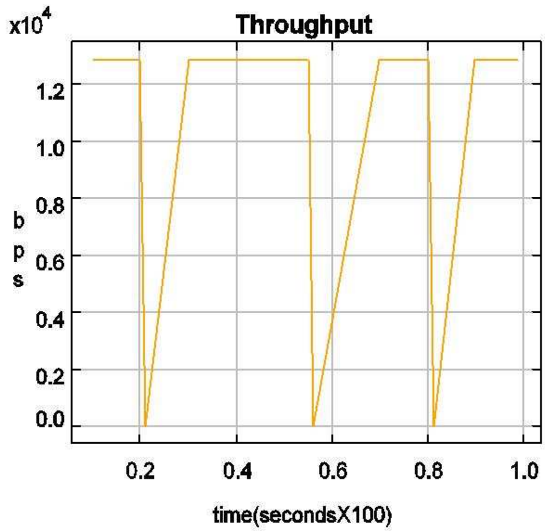

Figure 20: Throughput at sink node after data fusion for the fifty node scenario.

References:

[1] A. Sinha and A. Chandrakasan, Dynamic Power Management in Wireless Sensor Networks, *IEEE. Design and Test.* 18, 2001, pp. 62–74.

[2] D.–L. Hall and S.–A.–H. McMullen, *Mathematical Techniques in Multisensor Data Fusion* Artech House –Boston–London 2004

[3] K. Kalpakis K. Dasgupta and P. Namjoshi, An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks, *IEEE. WCNC.*, 2003, pp. 1948–1953.

[4] W. Heinzelman, J. Kulic and H. Balakrishnan, Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, *5th. ACM. MOBICOM*, 1999, pp. 178–185.

[5] J. Kulic, W. Heinzelman and H. Balakrishnan, Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks, *Wireless Networks.* 8, 1999, pp. 169–185.

[6] J. Heidemann, Building Efficient Wireless Sensor Networks with low-level naming, *18th. Sym. Syst. Princ.*, 2001, pp. 146–159.

[7] J. So, J. Kim and I. Gupta, Cushion: Autonomically Adaptive Data Fusion in Wireless Sensor Networks, *IEEE. MASS.*, 2005, pp. 1–3.

[8] Y. Zhenyu, Z. Hai, L. Kai, S. Peigang, G. Yishan, Z. Yongqing, and X. Ye, Multi-sensor Data Fusion in Wireless Sensor Networks, *CESA*, 2006, pp. 1690–1694.

[9] J. Wallace, D. Pesch, S. Rea and J. Irvine, Fuzzy Logic Optimisation of MAC Parameters and Sleeping Duty-Cycles in Wireless Sensor Networks, *IEEE. Veh. Tech. Conf.* 62, 2005, pp. 1824-1828.

[10] D. Estrin, R. Govindan and J. Heidemann, Embedding the Internet, *Commun. ACM.* 5, 2000, pp. 39–41.

[11] M. Ilyas and I. Manhgoub, *Habdbook of Sensor Networks: Compact Wireless and Wired Sensor Systems,* CRC Press –Florida 2005

[12] W.–R. Heinzelman, A. Chankdrakasan and H. Balakrishnan, Energy Efficient Communication Protocol for Wireless Microsensor Networks, *Int. Conf. Sys. Scien.* 8, 2000, pp. 3005–3014.

[13] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach,* Morgan Kaufmann Publishers –San Francisco 2004

[14] G.–J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications* Prentice Hall –New Jersey 1995

[15] J.–S. R. Zang, C.–T. Sun and E. Mizutani, *Neuro-Fuzzy and Fost Computing* Prentice Hall –New Jersey 1996

[16] A. Sobeih, W.–P. Chen, J.–C. Hou, L.–C. Kung, N. Li, H. Lim, H.–Y. Tyan and H. Zhang, J-Sim: A Simulation Environment for Wireless Sensor Networks, *IEEE. Wire. Comm. Magaz.* 13, 2006, pp. 175-187.