Data Fusion Utilization for optimizing Large-Scale Wireless Sensor Networks

Mohammadreza Soltani, Michael Hempel, Hamid Sharif
Advanced Telecommunications Engineering Laboratory, Dept. of Computer and Electronics Engineering
University of Nebraska - Lincoln
Omaha, NE, USA
msoltani@huskers.unl.edu, {mhempel, hsharif}@unl.edu

Abstract*—Wireless Sensor Networks (WSN) continue their tremendous growth acceleration. WSNs have found their way into a wide range of domains, from military and transportation applications to medical and environmental monitoring. Some of these applications can include a very large number of nodes, which poses significant challenges to network lifetime, data transmission, and overall reliability. Recently, data fusion approaches are gaining traction in WSNs for improving reported data accuracy and help predict future events. They are used to improve the reliability of delivered information. While this addresses data accuracy, it does not address the inefficiencies caused by very large nodes and high data redundancy. Data aggregation is a simple way of streamlining data flow, but does not fully address the issue. The large WSN size causes congestion and increases the traffic load in the network; plus, decreasing the performance of the WSN and potentially disrupting its operation altogether. In this paper, we therefore explore Kalman Filters (KF) based data fusion as a technique to reduce the number of active sensor nodes in a very large WSN to conserve network resources while preserving the required data reliability and accuracy. Our results show that there is great potential for improving WSN operations utilizing our proposed approach.

***Keywords*—data fusion; large-scale WSN; network size reduction; dynamic node reduction; Kalman Filter; network reliability; data accuracy**

I. Introduction

In recent years, WSN applications have been growing at a tremendous pace. They have found their way into numerous domains. In most such applications, there are networks with a very large number of sensor nodes involved. In these networks, each sensor observes events and report its measurement information to the gathering center for additional processing and decision-making. At the same time, a critical issue arises with regards to the reliability of the delivered information and the accuracy of the sensed data reported by each sensor node. One of the most prevalent topics of discussion in sensor networks for increased reliability in data delivery to the sink node is the use of data fusion techniques. Data fusion is defined as a method that can combine different information from various sensors to achieve improved accuracy and reliability compared to what could be attained by the evaluation of individual sensor readings. This information can be of the same type - in homogeneous sensor networks - or it can be of varying type in the case of heterogeneous sensor networks.

The central idea of data fusion is to leverage the collaboration of different sensor nodes in terms of complementary information by involving the redundancy within their reported data. That is, some sensors can observe one aspect of the event while the others cannot, and hence the fusion of this complementary information can improve overall accuracy. This implies that the joint information has a tendency to reduce data ambiguity and uncertainty [1].

Although using data fusion is an efficient tool for improving reliability, in very large sensor networks it can also cause significant problems. After a large group of sensor nodes have measured the specific phenomena they still need to report it to a central processing site for specific application. This can significantly increase the load of the network, cause congestion, and severely degrade the state of the network. Therefore, one key challenge is to manage the high network load without sacrificing the increased data accuracy provided through data fusion. In this paper, our objective is to demonstrate that data fusion not only can be used for achieving the desired accuracy, but it can be employed to also dynamically reduce the number of active nodes in large sensor networks. In recent years, several mathematical models for data fusion have been proposed. One of the most important and dominant approaches is based on the Bayesian method, which uses Bayesian estimation in probability and statistics theory for achieving optimal state estimation in dynamic systems [2]. In linear systems and using a Gaussian assumption for the process noise and observation noise, the Bayesian method is represented by the Kalman Filter (KF), which is one of most successful estimation tools in control applications and data fusion methods [3].

For the purpose of this paper and illustrating the capabilities of node reduction in large-scale WSNs, we have chosen target tracking as our sample application. We assume that there are a large number of sensor nodes distributed in an area, and their responsibility is to sense a moving object. Hence, the object's state changes over time. Our reason for selecting target tracking for demonstrating our work is the existence of numerous mathematical models for describing the target state and observation, which are the two key components in data fusion [4]. However, it should be noted that our method for active node set reduction can equally be applied to any other WSN application.

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

Our investigation is to present our approach and results which show data fusion can reduce the number of active nodes in large sensor networks while maintaining reliability and accuracy compared to the original number of nodes before the reduction.

The remainder of this paper is organized as follows. In section II we discuss papers related to data fusion in WSNs. Our node reduction model and algorithm are introduced in section III. The simulation results are presented in section VI and section V concludes this paper.

II. RELATED WORK

Data fusion is a topic that has been researched extensively, particularly for the task of data accuracy improvement. It is also slowly finding its way into related application domains.

In [5], the authors present and analyze the general idea of data fusion. The authors in [6] introduce the concept of multi-sensor data fusion structures in autonomous systems. Based on this paper, the different architectures for fusing data can be categorized as centralized, decentralized, or distributed. The paper also discusses the advantages and disadvantages for each category. In [7], different architectures are discussed in more details and mathematical models for these structures are proposed.

Data Fusion can be used for a variety of different purposes. In terms of the overall purpose of data fusion, however, it can be employed for data fusion estimation, data fusion detection, and data fusion decision [5]. However, most of the time these areas cannot be clearly distinguished from each other and all of them can be considered in a single application. Furthermore, there are different mathematical approaches and algorithms for each of these areas in data fusion. For example, Bayesian approaches are among the most common methods for data fusion estimation. Several representative Bayesian algorithms can be found in [8-9]. In these papers, in addition to describing the state equation, different measurements from different sensors are modeled as observation equations. Using the concept of Bayesian estimation, the conditional probability of the interest state given the sensors' observation sequences up to the current moment is calculated. After that, the estimate of the state in each time instance will be available. Because of the widespread application of Bayesian methods in various applications, there are different kinds of Bayesian filters that have been proposed over the years. For instance, the authors in [10-11] have used Extended Kalman Filters (EKF) [12] and Unscented Kalman Filters (UKF) [13] for data fusion.

Approaches for data fusion detection are most commonly based on the Dempster-Shafer method or the Evidence Theory. In [14], the author introduces this method and also provides a comparison between this method and the Bayesian approach. In [15], the author has used the Dempster-Shafer algorithm in autonomous mobile robots for fusing the data from multiple sensors in order to detect sensing anomalies. In [16], the authors have presented a Dempster-Shafer data fusion approach for data association [17] in multi-target tracking applications, and they compare their results with those obtained using the Bayesian model.

Other possible mathematical data fusion methods include information theory approaches, methods based on fuzzy logic, neural networks, and heuristic methods. For instance, in [18] an information theory approach has been discussed. The authors have presented an entropy approach based on Shannon's entropy. Also, the authors show how to model and formulate an information fusion problem based on this method. Fuzzy logic methods are another important approach for combining and fusing of different sensor data. A discussion about these solutions can be found in [19].

Using artificial neural networks in data fusion is presented in [20]. In this paper the authors have attempted to use different functional expansions to increase the speed of convergence in the fusion of sensor readings for increasing their reliability and accuracy. The author of [21] presents a heuristic algorithm for data fusion. An adaptive weighted algorithm for the location estimation application is proposed in order to reduce the amount of computations required in traditional data fusion approaches for heterogeneous measurements.

While all of these abovementioned data fusion methods attempt to achieve more reliable data, none of them address the issue of reducing the number of nodes in multi-sensor networks, especially in large-scale wireless sensor networks. Consequently, the energy consumption in large-scale WSNs has been ignored thus far in this area of research. On the other hand, the authors of [22-24] discuss various mechanisms to achieve energy efficiency in generic WSNs without consideration for data fusion. For instance, in [22, 23], the authors introduce different MAC protocols, provide a classification for those and finally discuss some open issues in MAC protocols. In [24], the authors address the energy efficiency and QoS in routing schemes and introduce a multipath routing protocol that uses the residual energy, buffer size and SNR to select the next node for relaying the packets.

III. PROPOSED MODEL AND ALGORITHM

In this section, our model for using data fusion in reducing the number of sensor nodes in large-scale WSNs is presented. The primary assumptions of our model are shown below:

- The application we have used in this study is tracking a moving object utilizing a constant velocity model that includes noise.
- The centralized architecture for data fusion is considered. In this structure, each node sends its own observation as raw data to the data fusion center for processing.
- The nodes are all sensing the object's position at a constant rate. In other words, we assume that these sensor nodes are homogeneous.
- For simplicity, all of the sensor nodes are synchronous. That is, their measurements are performed at the same time and arrive at the data fusion center at the same time.
- The situation of sensor network is constant over the time. It means that the nodes are stationary and environmental conditions do not change during the observation time.
- For data fusion we use the Bayesian model. Furthermore, we assume that both state equation and observation

equation are linear functions of the state. Also, both process noise and observation are additive and have Gaussian distributions.

The general architecture of our model is illustrated in figure 1. In this structure, Sensor nodes measure the position of the object at each time step and send their results to the data fusion center. In the data fusion center all of this information is fused to obtain more accurate data. In the next stage, the data fusion center starts searching among all nodes' data in order to find similar nodes in terms of measurements. If one or more such nodes are found, then the data fusion center sends a command to this node set to tell them to switch to sleep mode. Thus, our model consist of two steps: the first step involves the data fusion while the second step is the reduced set determination phase. These two steps are discussed in the following subsections in more detail.

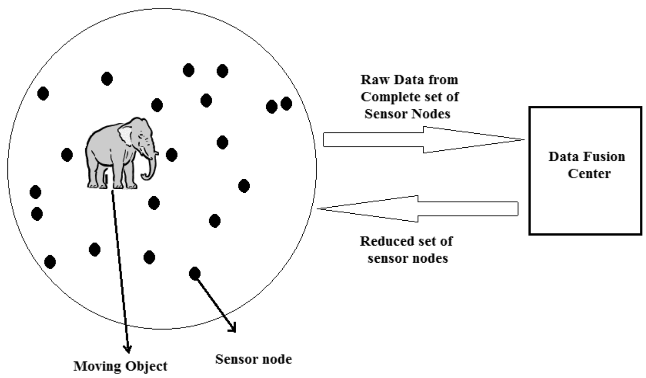

Fig. 1. The general structure of proposed model.

A. First Step – Data Fusion Process

For data fusion two important aspects should be determined: first, the state equation that indicates the evolution of the object state over time, and second the sensor model or observation model. Because we assume that the object has a constant velocity model we can utilize the DWNA model [25]. The state equation and observation equation can be expressed as follows (we do not consider control input in the state equation):

$$X(k+1) = F.X(k) + \Gamma.v(k) \tag{1}$$

and

$$Z_i(k) = H_i.X(k) + w_i(k) \tag{2}$$
$$i = 1,2,\dots,n_z$$

Where X is the state vector and includes two components: position and velocity. $v(k)$ denotes to the process noise, Z_i is the observation by sensor i, and $w_i(k)$ denotes to the observation noise corresponding to sensor i. Also, F is the transition matrix, Γ is the noise process matrix, H_i is the observation matrix for sensor i, and n_z denotes all the sensors participating in the measuring process. In the DWNA model the transition matrix F and noise process matrix Γ have the following relationship:

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3}$$

$$\Gamma = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \tag{4}$$

where T is called the revisit rate and indicates the sensing time interval of each sensor. Because we assume that all sensors obtain the position of the moving object we can express the observation matrix H_i as:

$$H_i = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{5}$$

Another important aspect is related to the distribution of both noise models, which we have listed in our assumptions as being Gaussian. The process noise and observation noise can thus be expressed using the following equations:

$$E[(\Gamma v(k) v(k)^T \Gamma^T)] = Q(k) \tag{6}$$

and

$$E[(w(k) w(k)^T)] = R(k). \tag{7}$$

These noises are stationary and assumed they are mutually independent. Hence, the covariance matrix for the process noise is as follows:

$$Q(k) = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix}. \tag{8}$$

It should be noted that both state equation and observation equation are discrete in time and are modeled as linear functions of state. Thus, by considering two components of X we can write (1) as follows [22]:

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} v(k). \tag{9}$$

In the first step, all the network's sensors start observing the position of the moving object. Based on these observations, we then employ Bayesian estimation or Kalman Filter estimation to obtain the state of the object for each time step using conditional mean given observation up to the current time. It is well known that a Kalman Filter estimates the state of a dynamic system in two stages: the prediction stage followed by the update stage based on the previously predicted values. Because this is a multi-sensor scenario, the implementation of the Kalman Filter is an important issue:

- It is possible that during the estimation in different iterations of the Kalman Filter the symmetry structure of the covariance matrix is lost.
- Because of rounding errors in computation, it is possible for some components of covariance matrix to become zero. In this case, the covariance matrix is not positive definiteness anymore. As a result it may be singular. This situation implies that for those components of the state filter assures completely the estimation is correct that it is not true.
- When the number of nodes is large the selection of the implementation method is critical since an inefficient solution can decrease the performance of the Kalman Filter.

For these reason, we have chosen the Information Filter approach for our work. The prediction of this method is the same as the standard Kalman Filter with the following prediction equations [13]:

$$\hat{X}(k+1\,|k) = F.\hat{X}(k|k) \tag{10}$$

and

$$P(k+1|k) = F.P(k|k).F^T + Q(k). \tag{11}$$

Equation (10) denotes the prediction state and (11) denotes the covariance state prediction. Before presenting the update equations, due to the presence of n_z sensors we consider the following two stack vectors:

$$Z(k) = \begin{bmatrix} Z_1^T(k) \\ \vdots \\ Z_{n_z}^T(k) \end{bmatrix} \tag{12}$$

and

$$H(k) = \begin{bmatrix} H_1^T(k) \\ \vdots \\ H_{n_z}^T(k) \end{bmatrix}. \tag{13}$$

Also, because the sensor observations are independent from each other we can write:

$$R(k) = diag(R_1(k), R_2(k), \dots, R_{n_z}(k)) \tag{14}$$

Equation (14) denotes to the observation covariance matrix for n_z sensors. Based on these definitions and the inverse matrix lemma [17] we can write the updating stage of the Information Filter as:

$$\hat{X}(k|k) = P(k|k)[P^{-1}(k|k-1)\hat{X}(k|k-1) + H^T(k)R^{-1}(k)Z(k)] \tag{15}$$

and

$$P(k|k)^{-1} = [P^{-1}(k|k-1) + H^T(k)R^{-1}(k)H(k)]. \tag{16}$$

The above equations (15) and (16) denote the update state estimation and covariance state estimation, respectively.

Note that (14) calculates the estimation of vector *X*, which includes position and velocity of the moving object, in each time step from the data obtained by all the sensors. After calculating of this estimation, the data fusion center starts the second phase in our model.

B. *Second Step – Selection Process*

This phase is used to identify similar sensor nodes that can be switched to sleep mode, with only the remaining active nodes participating in data gathering and reporting for data fusion. This set of remaining nodes we are referring to as the Reduced Set (RS).

In this step data fusion center select an appropriate set from among all sensors. This selecting process is conducted using two decision levels. The first level – the data association level – occurs when the raw data is received from the complete sensor set. Here, the data fusion center determines the measurements which are closely related to the state estimation. In other words, a data association algorithm is used to ignore unrelated measurements [17]. During the second step, the data from these nodes is used to find similar nodes. Once determined, the data fusion center sends these nodes a command to go to sleep mode. Observation in subsequent time steps is thus only carried out by the reduced node set. In the sections below we explain each of these steps in more detail.

1) Data Association Step

In this step, measurements closely related to the state estimation are determined by using the concept of a validation gate. Each observation from a specific sensor either falls into this gate region or outside, in which case they are removed from further processing. In this study, we have used a modified version of the Nearest Neighborhood Standard Filter (NNSF). The idea of this algorithm is to consider only those sensor measurements that are inside of the gate validation region. This gate is defined as follows [7]:

$$G(k) = \{Z_i(k) | (Z_i(k) - \hat{Z}_i(k|k-1)^T S_i(k)^{-1} (Z_i(k) - \hat{Z}_i(k|k-1)^T \leq \gamma\,\} \tag{17}$$

Where *G(k)* denotes to the probability of falling into the gate region, $\hat{Z}_i(k|k-1)$ and $S_i(k)$ are predicted measurement and predicted measurement covariance matrix corresponding to the sensor i at time *k*, respectively. This covariance matrix is given by the following equation:

$$S_i(k) = HP(k|k-1)H^T + R_i(k). \tag{18}$$

The inner term in (17) is considered as the squared distance between the observation generated by sensor i and the predicted measurement calculated by the filter. This squared distance graphically denotes an ellipsoid in the observation space. Thus, if the error of a sensor is less than the thresholdγ, then the correspondent measurement is considered to be related.

Because we assume that our measurements have Gaussian distribution, the distance between a sensor and its predicted measurement has a χ^2 distribution. As a result, the value of γ can be obtained using the χ^2 distribution table and a given probability.

2) Similar Node Determination Step

After obtaining the related measurements to the state estimation, these results are compared to each other in order to determine node similarity. If there are multiple related sensors, only one of them should observe the state in the next time steps whereas all others in this group can go to sleep. To do so, the data fusion center searches between all remaining sensors from the previous level to create different temporary sets that include sensors with observation values less than a specific threshold. Between multiple temporary sets, shared nodes should then be determined as this indicates that temporary sets can be merged. This process is illustrated in figure 2.

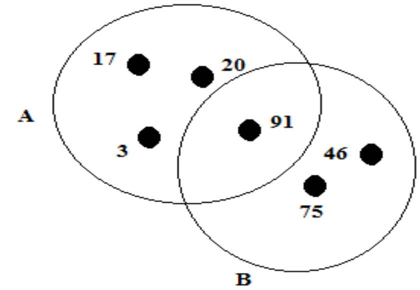

Fig. 2. Two Sets with similar node in each set.

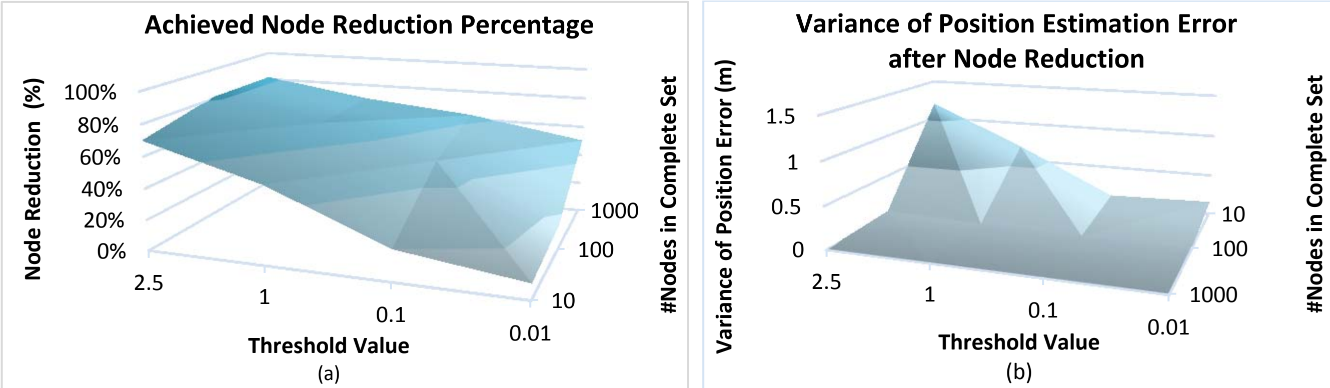

Fig. 3. (a) Achieved Node Set Size Reduction (b) Variance of Position Estimation Error in Reduced Set for various complete set sizes and threshold values

Suppose that A and B are sets having similar sensors in terms of their observation value. For example, nodes 91, 46, and 75 have observations whose values are mutually close to each other. Based on this similar set formation, we should determine which node is selected from each set. If for example node 17 from set A and node 46 from set B are randomly selected while all other nodes are switched to sleep mode, this will achieve an inefficient selection process outcome. Because node 91 is similar to the nodes in both sets of A and B, it is reasonable to select this node on behalf of the others in both sets. So, it is essential to consider these common nodes as intersections among the different sets for efficiently set size reduction.

It is also important to note that if the intersection of different sets includes more than one node, we further consider other criteria for choosing between them. However, for our simulations we randomly select one of the intersecting set nodes.

By following this method, we can determine the reduced set and the fusion process can continue with this reduced set of nodes reporting their measurements instead of the entire Complete Set, whereas the reliability of the fused data is not impacted.

IV. Simulation Results

In this section, we present the results of simulations using the proposed method for node reduction in large-scale WSNs. For each scenario, we have employed large number of Monte Carlo simulation Monte Carlo simulation runs. The scenarios consider different values for the threshold parameter in the node similarity determination process as well as different complete set sizes. Furthermore, the magnitude of observation noise for each sensor is assumed to be distributed uniformly in the interval 0 to 5. In figure 3a, the different scenarios and the achieved node set reduction are illustrated. We considered scenarios for four different threshold values, ranging from 0.01 to 2.5, as well as different complete set sizes ranging from 10 nodes to 1000 nodes.

As it can be observed in figure 3a, when the threshold value increases, a larger number of similar nodes are determined and removed and thus a higher set size reduction percentage is achieved.

The variance of the position error for the different scenarios is illustrated in figure 3b. From these results, we can see that when the threshold value increases, the amount of estimation accuracy achieved by the reduced node set is decreased slightly, which leads to the corresponding increase in variance. In figures 4 and 5, the error in position estimation incurred by reducing the node set size in two sample estimations is shown. Figure 4 illustrates a case where the threshold is chosen to be 0.1. The node set was reduced from 1000 nodes to 372, or by over 62%. We can see that the maximum incurred error is only around 3 mm! Similarly, figure 5 shows the incurred estimation error for the object position with a threshold of 1.0 in a scenario with 1000 nodes, reduced down to 302 active sensors. As it can be seen, the error increased slightly, to a maximum value of 5 mm, whereas most of the errors remained in a range of less than 2 mm. This represents a negligible error in most applications.

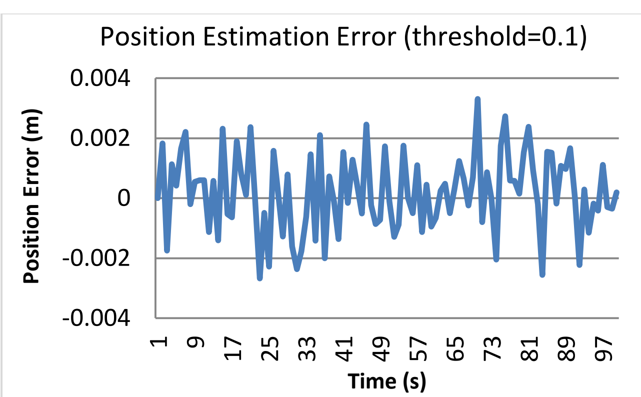

Fig. 4. The Error between complete set and reduced set in Position Estimation for threshold=0.1.

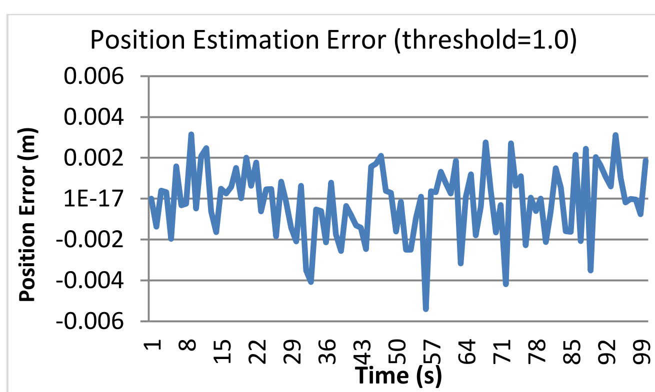

Fig. 5. The Error between complete set and reduced set in Position Estimation for threshold=1.

In table I, the percentage of node set reduction is shown for varying complete set sizes and threshold values.

TABLE I. THE PERCENTAGE OF NODE REDUCTION

	Threshold Values			
# Nodes in Complete Set	0.01	0.1	1	2.5
10	10%	20%	50%	70%
100	16%	51%	69%	81%
1000	50.3%	62.8%	69.8%	81.4%

V. CONCLUSION

In this paper, we presented our approach of data fusion for resource efficiency in very large wireless sensor networks. In such networks, the high node count causes high data volume, and large resource consumptions. In our approach, we utilized data fusion to determine a reduced node set to be active in the network, resulting in reduction of network resource consumptions. This node reduction is in addition to the traditional benefit of data fusion for improving data reliability and accuracy. In our scheme, only the nodes from the reduced set resulted from data fusion approach are needed in reporting events, whereas all others are allowed to be switched to sleep mode. This node decrease results in reducing the network load and conserving significant energy and other network resources.

Our method for determining the reduced set of sensors is based on the data association of related measurements and selecting similar nodes. The simulation results for our presented data fusion approach show that fusing data using measurements from only the reduced set of sensors maintains a high level of accuracy and a reduced node count of up to 82%! This represents a significant saving in terms of energy and other network resources.

VI. ACKNOWLEDGEMENTS

This study is being conducted at the University of Nebraska-Lincoln by the research faculty and students at the Advanced Telecommunications Engineering Laboratory (www.TEL.unl.edu). This project is supported by the Mid-America Transportation Center.

REFERENCES

[1] H. S. Carvalho, W. B. Heinzelman, A. L. Murphy, C. J. N. Coelho, "A general data fusion architecture," IEEE International Conference of Information Fusion, 2003, pp 1465 – 1472.

[2] J. R. Raol, "Multi-Sensor Data Fusion with MATLAB," CRC Press, Inc., 2010.

[3] M. S. Grewal, A. P. Andrews, "Kalman Filtering: Theory and Practice Using MATLAB," John Wiley & Sons, Inc., 2001.

[4] X. R. Li, V. P. Jilkov, "A Survey of Maneuvering Target Tracking: Dynamic Models," IEEE Conference on Signal and Data Processing of Small Targets, pp 1333 – 1364, 2000.

[5] Y. Zhu, et al., "Networked multisensor decision and estimation fusion: based on advanced mathematical methods," CRC Press, 2013.

[6] H. Xinhan, W. Min, "Multi-sensor Data Fusion Structures in Autonomous Systems: A Review," IEEE International Symposium on Intelligent Control, pp 817 - 821, 2003.

[7] H. Durrant-Whyte, "Multi Sensor Data Fusion," Australian Centre for Field Robotics, University of Sydney, NSW, 2006.

[8] W. Koch, "On Bayesian Tracking and Data Fusion: A Tutorial Introduction with Examples," IEEE Aerospace and Electronic Systems Magazine, pp 29 – 52, 2010.

[9] O. Punska, "Bayesian Approaches to Multi-Sensor Data Fusion, A dissertation for the degree of Master of Philosophy", University of Cambridge, 1999.

[10] S. Y. Chen, "Kalman Filter for Robot Vision: A Survey," IEEE Transactions on Industrial Electronics, vol. 59, no. 11, Nov 2012.

[11] B. Sadeghi, B. Moshiri, "Second-order EKF and Unscented Kalman Filter Fusion for Tracking Maneuvering Targets," IEEE International Conference on Information Reuse and Integration, 2007.

[12] H. Durrant-Whyte, "Introduction to Estimation and the Kalman Filter," Australian Centre for Field Robotics, University of Sydney, NSW, 2006.

[13] D. Simon, "Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches," John Wiley & Sons, Inc., 2006.

[14] S. Challa, D. Koks, "Bayesian and Dempster–Shafer Fusion," Information and Communications Group, Faculty of Engineering, University of Technology, Sydney, pp. 145–176, 2004.

[15] R. R. Murphy, "Dempster–Shafer Theory for Sensor Fusion in Autonomous Mobile Robots, "IEEE Transactions on Robotics and Automation, vol. 14, no. 2, 1998.

[16] A. Dallil, M. Oussalah, "Sensor Fusion and Target Tracking Using Evidential Data Association," IEEE Sensors Journal, vol. 13, no. 1, 2013.

[17] Y. Bar-Shalom and T.E. Fortmann, "Tracking and Data Association," Academic Press, 1988.

[18] B. Fassinut-Mombot, J.-B. Choquel, "A new probabilistic and entropy fusion approach for management of information sources," Information Fusion, pp 35–47, 2004.

[19] W.-T. Sung, S.-J. Hsiao, C.-C. Tsai, "Multi-Sensors Data Fusion for Precise Computing based on Fuzzy Approach," IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), 2011.

[20] J. Zhang, K. Wang, Q. Yue, "Data Fusion Algorithm based on Functional Link Artificial Neural Networks," IEEE Congress on Intelligent Control and Automation, 2006.

[21] Y.-S. Chiou, F. Tsai, S.-C. Yeh, "A Low-Complexity Data-Fusion Algorithm Based on Adaptive Weighting for Location Estimation", IEEE 2012 International Conference on Information Security and Intelligence Control (ISIC), 2012.

[22] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, "Estimation with applications to tracking and navigation," John Wiley & Sons, Inc., 2001.

[23] J. Ben-Othman, B. Yahya, " Energy efficient and QoS based routing protocol for wireless sensor networks.," J. Parallel Distrib. Comput. 70(8): 849-857, 2010.

[24] Bashir Yahya, Jalel Ben-Othman, "Towards a classification of energy aware MAC protocols for wireless sensor networks," Wireless Communications and Mobile Computing 9(12): 1572-1607, 2009.

[25] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, "Estimation with applications to tracking and navigation," John Wiley & Sons, Inc., 2001.