REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 1-06-2005	**2. REPORT TYPE** Bi-annual Performance/Technical Report	**3. DATES COVERED** *(From - To)* 12/01/2004 - 05/31/2005

4. TITLE AND SUBTITLE

Bi-annual (12/2004--05/2005) Performance/Technical Report
for ONR YIP Award under Grant N00014-03-1-0466
Energy Efficient Wireless Sensor Networks Using Fuzzy Logic

5a. CONTRACT NUMBER

5b. GRANT NUMBER

N00014 – 03 –1 –0466

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Liang, Qilian

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Texas at Arlington
Office of Sponsored Projects
PO Box 19145
Arlington, TX 76019

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution is Unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

During the period of 12/1/2004 -- 5/31/2005, we have proposed different approaches on energy efficient wireless sensor networks. 1) We proposed an event forecasting methodology for wireless sensor networks using interval type-2 fuzzy logic system. We also studied the fundamental performance analysis of different event detection schemes. 2) We studied spectrum efficient coding scheme for correlated non-binary sources because there exists bandwidth constraint in wireless sensor networks. 3) We proposed to reduce the redundancy in wireless sensor networks using SVD-QR method. 4) A hybrid approach for Asynchronous Energy-Efficient MAC (ASCEMAC) Protocol was proposed for wireless sensor networks. 5) We studied energy-efficient query in sensor database systems with uncertainties. 6) We proposed a fuzzy sensor deployment scheme, and studied clustering in sensor networks with fuzzy cluster radius. 7) We proposed a cross-layer (physical layer, data-link layer and application layer) design scheme for mobile ad hoc networks. Eleven papers were produced during the past six months, and are attached to this report.

15. SUBJECT TERMS

Wireless Sensor Network, Energy Efficiency, Fuzzy Logic.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Qilian Liang
U	U	U	UU	83	**19b. TELEPHONE NUMBER** *(Include area code)* 817-272-1339

Bi-annual (12/2004–05/2005) Performance/Technical Report for ONR YIP Award under Grant N00014-03-1-0466 Energy Efficient Wireless Sensor Networks Using Fuzzy Logic

Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016 USA
Phone: 817-272-1339, Fax: 817-272-2253
E-mail: liang@uta.edu

Abstract

During the period of 12/1/2004 – 5/31/2005, we have proposed different approaches on energy efficient wireless sensor networks.

1. We proposed an event forecasting methodology for wireless sensor networks using interval type-2 fuzzy logic system, which consists of sensed signal strength forecasting and event detection. We also studied the fundamental performance analysis of different event detection schemes.
2. We studied spectrum efficient coding scheme for correlated non-binary sources because there exists bandwidth constraint in wireless sensor networks.
3. We proposed to reduce the redundancy in wireless sensor networks using SVD-QR method.
4. A hybrid approach for Asynchronous Energy-Efficient MAC (ASCEMAC) Protocol was proposed for wireless sensor networks.
5. We studied energy-efficient query in sensor database systems with uncertainties.
6. We proposed a fuzzy sensor deployment scheme, and studied clustering in sensor networks with fuzzy cluster radius.
7. We proposed a cross-layer (physical layer, data-link layer and application layer) design scheme for mobile ad hoc networks.

Eleven papers were produced during the past six months, and are attached to this report.

1 Event Forecasting for Wireless Sensor Networks Using Interval Type-2 Fuzzy Logic System

Wireless sensor networks (WSN) are often used to perform event detection, tracking, and classification. Therefore, compared to ad-hoc networks, WSN should be event-centric. In [1], we proposed an event forecasting scheme for wireless sensor networks using interval type-2 fuzzy logic system. Our event forecasting scheme consists of two steps: sensed signal strength forecasting and event detection. We demonstrated that real-world sensed acoustic signals are self-similar, which means they are forecastable. We showed that a type-2 fuzzy memebership function (MF), *i.e.*, a Gaussian MF with uncertain mean is appropriate to model the sensed signal strength of wireless sensors. Two fuzzy logic systems (FLS), a type-1 FLS and an interval type-2 FLS were designed for signal strength forecasting. Furthermore, we proposed a double sliding window scheme for event detection based on the forecasted signals. Simulation results show that the interval type-2 FLS outperforms the type-1 FLS in signal strength forcating and the performance of event detection based on the forecasted signal from type-2 FLS is much better than that based on type-1 FLS.

2 Event Detection Algorithm and Fundamental Performance Analysis in Wireless Sensor Networks

In [3], we presented two methods to do event detection, one is Double Sliding Window Detection, the other one is Fuzzy Logic approach. The accuracy of the results is established via sensor network testbed and simulations. In [5][6], we presented a fundamental performance analysis of event detection in wireless sensor networks. We compared double sliding window theoretically against the fixed threshold approach. In [5], Rayleigh and Rician distributions are validated for the sensed signals and used in the performance analysis; and in [6], Gaussian distribution with uniformly distributed mean values are assumed for the analysis. Measures of performance for these tasks are well defined, including detection of false alarms or misses, classification errors, and track quality.

3 Spectrum Efficient Coding Scheme for Correlated Non-Binary Sources in Wireless Sensor Networks

Energy-aware technique to reduce energy consumption in distributed sensor networks has become a prominent topic in sensor network research. Various sensor network applications have taken energy efficiency into consideration. In the case of correlated binary sources, distributed source coding has been literally studied in information theory. However, data sources from real sensor networks are normally non-binary. In [4], we proposed a spectrum efficient coding scheme for correlated non-binary sources in sensor networks. Our approach constructs the codeword cosets for the interested source, taking advantage of statistical characters of the distinct observations from sensor nodes. The coset leaders are then transmitted via the channel and decoding is performed with the available side information. Simulations were carried out over independent and identically distributed (i.i.d) Gaussian sources and data collected from Xbow wireless sensor network test bed. Simulation results show that the proposed scheme performs at 0.5 - 1.5 dB from the Wyner-Ziv distortion bound.

4 Redundancy Reduction in Wireless Sensor Networks Using SVD-QR

In densely deployed wireless sensor networks, not only does the data of one sensor node have self-similarity, but the data from adjacent sensor nodes also have cross-similarity. Therefore, it is clear that there exists highly redundancy in the collected data from sensor nodes in the neighborhood. Due to the intrinsic properties wireless sensor networks have, *e.g.*, energy constraint, bandwidth limitation, this kind of information redundancy will impact the whole networks in a negative way. In [2], we proposed to use Singular-Value-QR Decomposition (SVD-QR) to reduce the redundancy in wireless sensor networks.

5 A Hybrid Approach for Asynchronous Energy-Efficient MAC (ASCEMAC) Protocol for Wireless Sensor Networks

In [7], a novel asynchronous energy-efficient MAC protocol, ASCEMAC, was proposed for wireless sensor networks. We combined both contention-based and schedule-based MAC protocols' energy saving strategies in our algorithm. In ASCEMAC, by applying free-running method and fuzzy logic rescheduling scheme, time synchronization which is necessary in existing energy-efficient MAC protocols is not required any more. Moreover, we presented a traffic-intensity and network-density-based model to determine essential algorithm parameters, such as power on/off duration, interval of schedule broadcast and super-time-slot size and order. Simulation results showed that our algorithm ensures the average successful transmission rate, decreases the data packet average waiting time, and reduces the average energy consumption. Therefore, network performance is improved and network lifetime is extended by using our algorithm.

6 Energy-Efficient Query in Sensor Database Systems with Uncertainties

Query processing methods have been studied extensively in the context of database systems. But they are not directly applicable in sensor database systems due to the characteristics of sensor networks: the decentralized nature of sensor networks, the limited computational power and energy scarcity of individual sensor node, and imperfect information recorded. In [8], we proposed an energy-efficient query optimization algorithm (QOA) for imperfect information in sensor database systems. We employed an in-network query processing method, which tasks sensor networks through declarative queries. Given a query, our QOA generates an energy efficient query plan for in-network query processing. Moreover, our algorithm can explicitly exposes uncertainty and ambiguity of query results to database users. As we know, it is troublesome or even impossible to keep a large number of data in sensor database systems for network resource constraints. In our algorithm, we formulated the probability distribution functions (PDFs) of measurement uncertainties according to the knowledge on observation coverage and devices utilized, instead of estimating them from prior data. The simulation results demonstrated that our algorithm can vastly reduce resource usage and thus extend the lifetime of sensor database system.

7 Fuzzy Deployment for Wireless Sensor Networks

In [9], we developped a fuzzy deployment for wireless sensor networks. Traditional deployments often assume a homogeneous environment, which ignores the effect of terrain profile and obstacles such as buildings, trees and so on. Nevertheless, in many applications, some areas need to be more critically monitored. All these factors are combined together through Fuzzy Logic System in our proposed scheme. Simulation results show that the Fuzzy Deployment improves the worst-case coverage by around 5 dB.

8 Clustering in Sensor Networks with Fuzzy Cluster Radius

Previous research shows that restraining cluster size helps energy efficiency in sensor networks. However, it is often ignored that the distance estimation in sensor networks is inaccurate enough for fine-grained clustering decision. In [10], we were concerned with developing a fuzzy cluster size to handle the distance error and non-linearity. A fuzzy logic system was developed to make clustering decision based on the received signal strength. Simulation results showed that the proposed Fuzzy Cluster Size scheme can keep the performance near the optimal range when distance estimation is distorted by log-normal shadowing.

9 Bottom-up Cross-Layer Optimization for Mobile Ad Hoc Networks

In [11], we introduced a cross-layer design method for mobile ad hoc networks. We use fuzzy logic system (FLS) to coordinate physical layer, data-link layer and application layer for cross-layer design. Ground speed, average delay and packets successful transmission ratio are selected as antecedents for the FLS. The output of FLS provides adjusting factors for the AMC (Adaptive Modulation and Coding), transmission power, retransmission times and rate control decision. Simulation results show that our cross-layer design can reduce the average delay, increase the throughput and extend the network lifetime. The network performance parameters could also keep stable after the cross-layer optimization.

References

[1] Qilian Liang, Lingming Wang, "Event Forecasting for Wireless Sensor Networks Using Interval Type-2 Fuzzy Logic System," submitted to *IEEE Transactions on Fuzzy Systems.*

[2] Qilian Liang, Lingming Wang, "Redundancy Reduction in Wireless Sensor Networks Using SVD-QR," submitted to *MILCOM*, Atlantic City, NJ, 2005.

[3] Qilian Liang, Lingming Wang, "Event Detection in Wireless Sensor Networks Using Fuzzy Logic System," presented at *IEEE Computational Intelligence on Homeland Security and Personal Safety*, Orlando, FL, March 2005.

[4] Haining Shu, Qilian Liang, "Spectrum Efficient Coding Scheme for Correlated Non-Binary Sources in Wireless Sensor Networks," submitted to *MILCOM*, Atlantic City, NJ, 2005.

[5] Haining Shu, Qilian Liang, "Fundamental Performance Analysis of Event Detection in Wireless Sensor Networks," submitted to *Globecom*, St Louis, MO, 2005.

[6] Haining Shu, Qilian Liang, "Event Detection in Sensor Networks: Algorithms and Performance Analysis," submitted to *MILCOM*, Atlantic City, NJ, 2005.

[7] Qingchun Ren, Qilian Liang, "A Hybrid Approach for Asynchronous Energy-Efficient MAC Protocol for Wireless Sensor Networks," submitted to *Globecom*, St Louis, MO, 2005.

[8] Qingchun Ren, Qilian Liang, "Energy-Efficient Query in Sensor Database Systems with Uncertainties" submitted to *MILCOM*, Atlantic City, NJ, 2005.

[9] Liang Zhao, Qilian Liang, "Fuzzy Deployment for Wireless Sensor Networks," presented at *IEEE Computational Intelligence on Homeland Security and Personal Safety*, Orlando, FL, March 2005.

[10] Liang Zhao, Qilian Liang, "Clustering with Fuzzy Cluster Radius," submitted to *Globecom*, St Louis, MO, 2005.

[11] Xinsheng Xia, Qilian Liang, "Bottom-up Cross-Layer Optimization for Mobile Ad Hoc Networks," submitted to *Globecom*, St Louis, MO, 2005.

Event Forecasting for Wireless Sensor Networks Using Interval Type-2 Fuzzy Logic System

Qilian Liang and Lingming Wang

Department of Electrical Engineering

University of Texas at Arlington

Arlington, TX 76019-0016. USA

Email: liang@uta.edu, wang@wcn.uta.edu

Abstract

Wireless sensor networks (WSN) are often used to perform event detection, tracking, and classification. Therefore, compared to ad-hoc networks, WSN should be event-centric. In this paper, we propose an event forecasting scheme for wireless sensor networks using interval type-2 fuzzy logic system. Our event forecasting scheme consists of two steps: sensed signal strength forecasting and event detection. We demonstrate that real-world sensed acoustic signals are self-similar, which means they are forecastable. We showed that a type-2 fuzzy memebership function (MF), *i.e.*, a Gaussian MF with uncertain mean is appropriate to model the sensed signal strength of wireless sensors. Two fuzzy logic systems (FLS), a type-1 FLS and an interval type-2 FLS are designed for signal strength forecasting. Furthermore, we propose a double sliding window scheme for event detection based on the forecasted signals. Simulation results show that the interval type-2 FLS outperforms the type-1 FLS in signal strength forcating and the performance of event detection based on the forecasted signal from type-2 FLS is much better than that based on type-1 FLS.

Index Terms : Wireless sensor networks, fuzzy logic systems, interval type-2 membership function, self-similarity, forecasting, event detection.

1 Introduction

Wireless sensor networking is an emerging technology that promises unprecedented ability to monitor and manipulate the physical world via a network of densely distributed wireless sensor nodes. The nodes can sense the physical environment in a variety of modalities, including acoustic, seismic, thermal, and infrared. They are networked together in an ad hoc fashion, which involves peer-to-peer communication in a network with a dynamically changing topology. Wireless sensor networks do not rely on a preexisting fixed infrastructure, such as a wireline backbone network or a base station. They are self-organizing entities that are deployed on demand in support of various events such as security and surveillance, monitoring of wildlife habitats, smart sensor-instrumented environments, and condition-based maintenance of complex systems, etc.

Sensor nodes are typically powered by small batteries that are hard to replace or recharge. Hence, energy constraint is a unique character of WSN compared with traditional wireless adhoc networks. Energy comsuption occurs in three domains: sensing, data processing (including AD/DA and digital signal processing), and communications[8]. According to [1], the sensor, signal processing parts operate at low frequency and consume less than $1mW$. This is over an order of magnitude less than the energy consumption of the communication part. Therefore, we prefer less communication/data exchange between sensor nodes but more local processing implemented by one single sensor node so as to increase the lifetime of the WSN.

The main goal of WSN is to monitor physical world. Usually, people are more intereted in unexpected events. For example, in a scenario of battlefield, people are more interested in the appearance of enemies. If a WSN is to monitor forest-fire, unusual increasing of the temperature should be a necessary warning to people. Both the appearance of enemies and the unusual increasing of the temperature can be seen as events. Because of the energy constraint of WSN mentioned previously, the ideal state of WSN should be event-driven, so that we can power off the communication part at most of the time. Only when certain sensor nodes detect an event, they trigger the RF channel, and transmit the useful information to clusterhead or gateway. This power on/off management will be easier if each wireless sensor could forecast its future sensed signal strength and make event detection.

In this paper, we propose an event forecasting scheme for wireless sensor networks using interval type-2 fuzzy logic system. Our event forecasting scheme consists of two steps: sensed signal strength forecasting and event detection. We use Xbow wireless sensor network professional develper's kit MOTE-Kit[7] as our testbed to get data sets from different scenarios. First of all, we show that the sensed signal strength is self-similar and long-range dependent using *variance-time plotting* , a common statistical method which has been widely used to verify self-similarity of time-series. Since the sensed signal strength is self-similar, its characteristics can be captured. We apply a type-1 FLS and an interval type-2 FLS to sensed signal strength forecasting. Furthermore, we make event detection based on the forecasted signal.

The remainder of the paper is organized as follows. Section 2 studies the self-similarity of sensed signal strength. Section 3 gives an overview of type-2 fuzzy sets and interval type-2 FLSs. In Section 4, we demonstrate that sensed signal strength in WSN should be modeled as a type-2 MF, a Gaussian MF with uncertain mean. Hence, we apply this knowledge and design an interval type-2 FLS to forecast the sensed signal strength in WSN. A singleton type-1 FLS is also designed for performance comparison. In Section 5, we propose double sliding window scheme to make event detection based on the forecasted signal. Simulation results and discussions are presented in section 6. Section 7 concludes this paper.

2 Self-Similarity of Sensed Signal Strength in WSN

For a detailed discussion on self-similarity in time-series, see [17] [16]. Here we briefly present its definition [2]. Given a zero-mean, stationary time-series $X = (X_t; t = 1, 2, 3, \cdots)$, we define the m-aggregated series $X^{(m)} = (X_k^{(m)}; k = 1, 2, 3, \cdots)$ by summing the original series X over nonoverlapping blocks of size m. Then it's said that X is H-self-similar, if, for all positive m, $X^{(m)}$ has the same distribution as X rescaled by m^H. That is,

$$X_t \overset{\triangle}{=} m^{-H} \sum_{i=(t-1)m+1}^{tm} X_i \quad \forall m \in N. \tag{1}$$

If X is H-self-similar, it has the same autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)]/\sigma^2$ as the series $X^{(m)}$ for all m, which means that the series is distributionally self-similar: the distribution of the aggregated series is the same as that of the original.

Self-similar processes can show *long-range dependence.* A process with long-range dependence has an autocorrelation function $r(k) \sim k^{-\beta}$ as $k \rightarrow \infty$, where $0 < \beta < 1$. The degree of self-similarity can be expressed using *Hurst* parameter $H = 1 - \beta/2$. For self-similar series with long-range dependence, $1/2 < H < 1$. As $H \rightarrow 1$, the degree of both self-similarity and long-range depence increases.

One method that has been widely used to verify self-similarity is the *variance-time plot,* which relies on the slowly decaying variance of a self-similar series. The variance of $X^{(m)}$ is plotted against m on a log-log plot, and a straight line with slope $(-\beta)$ greater than -1 is indicative of self-similarity, and the parameter H is given by $H = 1 - \beta/2$. We use this method to verify the self-similarity of acoustic signal.

In our experiments, 8 sensors were deployed in a lab. The location of each sensor is plotted in Fig. 1. We designed two scenarios, one is with a fixed source, and the other is without a fixed source. In Fig. 2, we plot the variance of $X^{(m)}$ against m on a log-log plot for 8 sensor node data respectively in the first scenario and Fig. 3 is under the second scenario. From the two figures, it's very clear that the no matter under what kind of condition the sensor network data have self-similarity because their traces have slopes much greater than -1.

3 Introduction of Type-2 Fuzzy Set and Interval Type-2 Fuzzy Logic Systems

3.1 Introduction to Type-2 Fuzzy Set

The concept of type-2 fuzzy sets was introduced by Zadeh [18] as an extension of the concept of an ordinary fuzzy set, *i.e.*, a type-1 fuzzy set. Type-2 fuzzy sets have grades of membership that are themselves fuzzy [3]. A type-2 membership grade can be any subset in $[0, 1]$ – the *primary membership*; and, corresponding to each primary membership, there is a *secondary membership* (which can also be in $[0, 1]$) that defines the possibilities for the primary membership. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity. Type-2 fuzzy sets allow us to handle linguistic uncertainties, as typified by the adage "words can mean different things to different people." A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal, "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner [5]".

Figure 4 shows an example of a type-2 set. The domain of the membership grade corresponding to $x = 4$ is also shown. The membership grade for every point is a Gaussian type-1 set contained in $[0, 1]$, we call such a set a "Gaussian type-2 set". When the membership grade for every point is a crisp set, the domain of which is an interval contained in $[0, 1]$, we call such type-2 sets "interval type-2 sets" and their membership grades "interval type-1 sets". Interval type-2 sets are very useful when we have no other knowledge about secondary memberships. An interval type-2 MF is characterized by an upper and lower MF [10]. An upper MF and a lower MF are two type-1 MFs which are bounds for the footprint of uncertainty of an interval type-2 MF. The upper MF is a subset which has the maximum membership grade of the footprint of uncertainty; and, the lower MF is a subset which has the minimum membership grade of the footprint of uncertainty.

Example 1: Gaussian Primary MF with Uncertain Mean

Consider the case of a Gaussian primary MF having a fixed standard deviation, σ_k^l, and an uncertain mean that takes on values in $[m_{k1}^l, m_{k2}^l]$, i.e.,

$$\mu_k^l(x_k) = \exp\left[-\frac{1}{2}(\frac{x_k - m_k^l}{\sigma_k^l})^2\right], \quad m_k^l \in [m_{k1}^l, m_{k2}^l] \tag{2}$$

where: $k = 1, \ldots, p$; p is the number of antecedents; $l = 1, \ldots, M$; and, M is the number of rules. The upper MF, $\overline{\mu}_k^l(x_k)$, is (see Fig. 5)

$$\overline{\mu}_k^l(x_k) = \begin{cases} \mathcal{N}(m_{k1}^l, \sigma_k^l; x_k), & x_k < m_{k1}^l \\ 1, & m_{k1}^l \leq x_k \leq m_{k2}^l \\ \mathcal{N}(m_{k2}^l, \sigma_k^l; x_k), & x_k > m_{k2}^l \end{cases} \tag{3}$$

where, for example, $\mathcal{N}(m_{k1}^l, \sigma_k^l; x_k) \triangleq \exp\left(-\frac{1}{2}(\frac{x_k - m_{k1}^l}{\sigma_k^l})^2\right)$.

The lower MF, $\underline{\mu}_k^l(x_k)$, is (see Fig. 5)

$$\underline{\mu}_k^l(x_k) = \begin{cases} \mathcal{N}(m_{k2}^l, \sigma_k^l; x_k), & x_k \leq \frac{m_{k1}^l + m_{k2}^l}{2} \\ \mathcal{N}(m_{k1}^l, \sigma_k^l; x_k), & x_k > \frac{m_{k1}^l + m_{k2}^l}{2} \end{cases} \tag{4}$$

□

3.2 Introduction to Type-2 FLS

Figure 6 shows the structure of a type-2 FLS[14]. It is very similar to the structure of a type-1 FLS [11]. For a type-1 FLS, the *output processing* block only contains the defuzzifier. We assume that the reader is familiar with type-1 FLSs, so that here we focus only on the similarities and differences between the two FLSs.

The fuzzifier maps the crisp input into a fuzzy set. This fuzzy set can, in general, be a type-2 set.

In the type-1 case, we generally have "IF-THEN" rules, where the lth rule has the form "R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l", where: x_is are inputs; F_i^ls are antecedent sets ($i = 1, \ldots, p$); y is the output; and G^ls are consequent sets. The distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important while forming rules; hence, the structure of the rules remains exactly the same in the type-2 case, the only difference being that now some or all of the sets involved are of type-2; so, the lth rule in a type-2 FLS has the form "R^l : IF x_1 is $\tilde{\mathrm{F}}_1^l$ and x_2 is $\tilde{\mathrm{F}}_2^l$ and $\cdots$ and x_p is $\tilde{\mathrm{F}}_p^l$, THEN y is $\tilde{\mathrm{G}}^l$".

In the type-2 case, the inference process is very similar to that in type-1. The inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this, one needs to find unions and intersections of type-2 sets, as well as compositions of type-2 relations.

In a type-1 FLS, the defuzzifier produces a crisp output from the fuzzy set that is the output of the inference engine, *i.e.*, a type-0 (crisp) output is obtained from a type-1 set. In the type-2 case, the output of the inference engine is a type-2 set; so, "extended versions" (using Zadeh's Extension Principle [18]) of type-1 defuzzification methods were proposed in[14]. The type-reduction gives a type-1 fuzzy set called "type-reduction set".

To obtain a crisp output from a type-2 FLS, we can defuzzify the type-reduced set. The most natural way of doing this seems to be by finding the centroid of the type-reduced set; however, there exist other possibilities like choosing the highest membership point in the type-reduced set.

General type-2 FLSs are computationally intensive, because type-reduction is very intensive. Things simplify a lot when secondary membership functions (MFs) are interval sets (in this case, the secondary memberships are either 0 or 1). When the secondary MFs are interval sets, the type-2 FLSs are called "interval type-2 FLSs". In [10], Liang and Mendel proposed the theory and design of interval type-2 FLSs. They proposed an efficient and simplified method to compute the input and antecedent operations for interval type-2 FLSs, one that is based on a general inference formula for them.

In an interval type-2 nonsingleton FLS with type-2 fuzzification and meet under minimum or product t-norm, the result of the input and antecedent operations, F^l, is an interval type-1 set, i.e., $F^l = [\underline{f}^l, \overline{f}^l]$, where $\underline{f}^l$ and $\overline{f}^l$ simplify to

$$\underline{f}^l = \underline{\mu}_{\tilde{\mathrm{F}}_1^l}(x_1) \star \ldots \star \underline{\mu}_{\tilde{\mathrm{F}}_p^l}(x_p) \tag{5}$$

and

$$\overline{f}^l = \overline{\mu}_{\tilde{\mathrm{F}}_1^l}(x_1) \star \ldots \star \overline{\mu}_{\tilde{\mathrm{F}}_p^l}(x_p) \tag{6}$$

where x_i $(i = 1, \ldots, p)$ denotes the location of the singleton. In this paper, we use center-of-sets type-reduction [10], which can be expressed as:

$$Y_{\cos}(Y^1, \cdots, Y^M, F^1, \cdots, F^M) = [y_l, y_r] = \int_{y^1} \cdots \int_{y^M} \int_{f^1} \cdots \int_{f^M} 1 \Big/ \frac{\sum_{i=1}^M f^i y^i}{\sum_{i=1}^M f^i} \tag{7}$$

where $Y_{\cos}$ is an interval set determined by two end points, y_l and y_r; $f^i \in F^i = [\underline{f}^i, \overline{f}^i]$; $y^i \in Y^i = [y_l^i, y_r^i]$, and Y^i is the centroid of the type-2 interval consequent set $\tilde{\mathrm{G}}^i$, and, $i = 1, \ldots, M$. We also use the training method proposed in [10] for designing an interval type-2 FLS in which its parameters are tuned

4 Sensed Singal Strength Forecasting Using Interval Type-2 FLS

Acoustic amplitude sensor node measures sound amplitude at its microphone. Assuming that the sound source is a point source and sound propagation is lossless and isotropic, a root-mean-squared (RMS) amplitude measurement z is related to the sound source position X as

$$z = \frac{a}{\|X - \varsigma\|} + w, \tag{8}$$

where a is the RMS amplitude of the sound source, ς is the location of the sensor, and w is RMS measurement noise [9]. According to [9], w is modelled as a Gaussian with zero mean and variance σ^2. The sound source amplitude a is also modelled as a random quantity, which is uniformly distributed in the interval $[a_{lo}, a_{hi}]$. Given the location of the sound source X and the sensor position ς, $\frac{a}{\|X-\varsigma\|}$ is uniformly distributed as a is. Therefore, z should be modelled as a Gaussian primary MF having a fixed standard deviation and an uncertain mean, as shown in Fig 5.

FLSs have been extensively used in time-series forecasting (e.g., [12], [10]). Since the sensed signal strength in WSN is self-similar as demonstrated in Section 2, its characteristics can be captured, which also means it can be forecasted. Here we apply an interval Type-2 FLS to do a multi-step forecasting, the step size is L. We use four antecedents, *i.e.*, $x(k-4\times L)$, $x(k-3\times L)$, $x(k-2\times L)$, and $x(k-1\times L)$, as inputs of the FLS to predict $x(k)$. Similarly, we use $x(k-4\times L+i)$, $x(k-3\times L+i)$, $x(k-2\times L+i)$, and $x(k-1\times L+i)$ to predict $x(k+i)$, $\forall i<L$. If antecedent has two fuzzy sets, the number of rules is $2^4=16$. The rules are set up as one example shown bellow:

R^l : IF $x(k-4\times L)$ is $\tilde{\mathrm{F}}_1^l$ and $x(k-3\times L)$ is $\tilde{\mathrm{F}}_2^l$ and $x(k-2\times L)$ is $\tilde{\mathrm{F}}_1^l$ and $x(k-1\times L)$ is $\tilde{\mathrm{F}}_2^l$, THEN $x(k)$ is $\tilde{G}^l$.

We use center-of-sets type reduction and steepest descent training algorithm [10] to design this interval type-2 FLS.

For comparison, we also design a type-1 FLS for signal strength forecasting. Antecedents are the same as in the interval type-2 FLS, however Gaussian MFs are chosen for this type-1 FLS. There are also 16 rules, since each of the antecedents has 2 fuzzy sub-set as well. The rule is designed as:

R^l : IF $x(k-4\times L)$ is F_1^l and $x(k-3\times L)$ is F_2^l and $x(k-2\times L)$ is F_1^l and $x(k-1\times L)$ is F_2^l, THEN $x(k)$ is G^l.

We use center-of-sets defuzzifier and steepest descent training algorithm to design this type-1 FLS.

Our event forecasting scheme consists of two steps: sensed signal strength forecasting and event detection. In this paper, we propose a new event detection algorithm, double sliding window event detection.

5 Double Sliding Window Event-Detection

In [15], the acoustic energy in a fixed period of time is integrated, when it exceeds a threshold, an event is claimed occurring, as:

$$E_s = \sum_{m=0}^{M-1} |z_{n-m}|^2, \tag{9}$$

$$E_s \geq E_{threshold}. \tag{10}$$

However, this simple method suffers from a significant drawback; namely, the value of the threshold depends on the sensed signal energy. When there is no event occuring in the sensing range, the sensed signal consists of only noise. The level of the noise power is generally unknown and can change when the environment changes or if unwanted interferers go on and off. Therefore, it is quite difficult to set a fixed threshold. We propose a double sliding window algorithm for event-detection so as to alleviate the threshold value selection problem.

The double sliding window event-detection algorithm calculates two consecutive sliding windows of the sensed signal energy. The basic principle is to form the decision variable as the ratio of the total energy contained inside the two windows. Figure 7 shows the windows A and B and the response of the ratio m_n to the start and end of a sensed event. It can be seen that when only noise is sensed the response is flat, since both windows contain ideally the same amount of noise energy.

The calculation of the window A and window B value is represented as

$$E_a = \sum_{m=0}^{M-1} |z_{n-m}|^2, \tag{11}$$

$$E_b = \sum_{l=0}^{M-1} |z_{n+l}|^2 . \tag{12}$$

Then the decision variable m_n is

$$m_n = \frac{E_a}{E_b}. \tag{13}$$

When m_n exceeds the threshold Th_1, an event is claimed occurring(see Fig. 7.(a)). The advantages of this approach are: 1st, the decision variable m_n does not depend on the sensed signal energy, but on the ratio of the energy of two consecutive windows; 2nd, we can predict not only the starting edge of the event, but also the ending edge, *i.e.* , when M_n below the threshold Th_2, the event is claim ending(see Fig. 7.(b)).

6 Simulations

Our simulations were based on $N = 480$ samples, $x(1)$, $x(2)$, ..., $x(480)$. The first 240 data, $x(1)$, $x(2)$, ..., $x(240)$, are for training, and the remaining 240 data, $x(241), x(242), \ldots, x(480)$ are for testing. In Fig. 8, we plot the sensed data that we used for training and testing, $x(1)$, $x(2)$, ..., $x(480)$. A standard $1kHz$ audio signal with different volume levels was used to simulate the events. Each sample has $1024ms$ duration.

We applied a type-1 FLS and an interval type-2 FLS for sensed signal forecasting. The initial locations of antecedent MFs were based on the mean, m_t, and std, σ_t, of the training data set. The parameters and number of parameters in the type-1 FLS and interval type-2 FLS are summarized in Table 1. The initial values we choose for the Guassian MFs are listed in Table 2. Then, we use steepest descent algorithm to train all the parameters based on the training data. After training, all the parameters and rules are fixed and we test the interval type-2 FLS based on the remaining 240 samples, $x(241)$, $x(242)$, ..., $x(480)$. We set the step size as $L = 5$ in both the type-1 FLS and the interval type-2 FLS. Meanwhile, the window size M equals to 5 in double sliding window event-detection as well. That makes the sensed signal forecating meaningful.

We compared the performance of the interval type-2 FLS with that of the type-1 FLS for sensed signal strength forecasting. For each FLS, we ran 100 Monte-Carlo realizations to eliminate the randomness of the consequences, and the two FLSs were tuned using a simple steepest-descent algorithm for 5 epochs. We used the testing data to see how each FLS performed by evaluating the root-mean-square-error (RMSE) between the defuzzified output of the FLS and the actual sensor data ($x(k+1)$), *i.e.*,

$$\mathrm{RMSE} = \sqrt{\frac{1}{240}\sum_{k=241}^{480}[x(k) - f(\mathbf{x}^k)]^2}, \tag{14}$$

where $\mathbf{x}^k = [x(k-4\times5), x(k-3\times5), x(k-2\times5), x(k-1\times5)]^T$, and T denotes transpose. The RMSE of all simulations are summarized in Figure 9. Observe Figure 9, the interval type-2 FLS outperforms the type-1 FLS in the sensed signal strength forecasting.

We are more interested in the system's capability of forecasting the events, especially the starting point of the events. We used the forecasted data sets to detect the starting point of the events, *i.e.*, the time stamp of event occurrence and then compared with the actual time stamp. We evaluated our double sliding window algorithm and compared it against the cumulated signal strength scheme[15]. We chose $Th_1 = mean + std$ for the double sliding window event detection. Since the threshold is hard to choose for cumulated signal strength scheme, we ran simulation for 3 different thresholds: *i.e.*, $mean$, $mean + std/2$ and $mean + std$. We also ran 100 Monte-Carlo simulations so as to get the average absolute error between the forecasted and actual time stamp, $\frac{1}{100}\sum_{i=1}^{100}|D_i - P_i|$, where D_i is the detected starting point (based on the forecasted signal) and P_i is the actual starting point. The results are summarized in Table 3.

Observe Table 3, the performance of event detection based on the forecasted signal from type-2 FLS is much better than that based on the forecasted signal from type-1 FLS. Meanwhile, our double sliding window is more effective than the existing cumulated signal strength scheme. Event forecasting helps us for power on/off management of the WSN, *i.e.* , we can power on the communication part of sensor nodes only when event has been forecasted. Since the sensor, signal processing parts consume less than 1/10 of the energy consumed by the communication part[1], this power on/off strategy can save the energy tremendously.

7 Conclusions

In this paper, we proposed an event forecasting scheme for wireless sensor networks using interval type-2 fuzzy logic system. Our event forecasting scheme consists of two steps: sensed signal strength forecasting and event detection. We demonstrated that real-world sensed acoustic signals are self-similar, which means they are forecastable. We showed that a type-2 fuzzy memebership function (MF), *i.e.*, a Gaussian MF with uncertain mean is appropriate to model the sensed signal strength of wireless sensors. We then applied an interval type-2 FLS to perform sensed signal forecasting. Furthermore, we proposed a double sliding window for event detection based on the forecasted signal, and compared it against the existing cumulated signal strength scheme. Simulation results show that FLSs can be used for sensed signal strength forecasting, and the interval type-2 FLS performs much better than the type-1 FLS in sensed signal forecasting. The sensed signal forecasting can further be used for event detection, and the average absolute error between the actual starting point and the point detected based on the sensed signal from the interval type-2 FLS is much smaller than the one based on the sensed signal from the type-2 FLS.

Acknowledgement

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466, "Energy Efficient Wireless Sensor Networks for Future Combat System Using Fuzzy Logic."

References

[1] Andrew Y. Wang, and Charles G. Sodini, " A Simple Energy Model for Wireless Microsensor Transceivers ," *IEEE Globecom 2004, Dallas, TX.*,

[2] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE Trans. on Networking*, vol. 5, no. 6, pp. 835-846, Dec 1997.

[3] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, USA, 1980.

[4] D. Estrin, and R. Govindan, " Next century chanlleges:scalable coordination in sensor networks ," *ACM Mobicom 1999, Seattle, WA.*, pp. 263-270, August 1999.

[5] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," *Int'l. J. Man-Machine Studies*, vol. 15, pp. 385-455, 1981.

[6] G. J. Pottie, and W. J. Kaiser, " Wireless integrated network sensors ," *Communications of the ACM*, vol. 43, no. 5, pp. 551-558, May 2000.

[7] Jason L. Hill, and David E. Culler, " Mica: A Wireless Platform for Deeply Embedded Networks ," *IEEE, Micro*, Vol. 22, No. 6, pp. 12-24, Nov.-Dec. 2002.

[8] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Communications*, Oct. 2000.

[9] L. E. Kinsler, A. R. Frey, A. B.Coppens and J. V. Sanders, *Fundamentals of Acoustic*, John Wiley and Sons, Inc., New York, USA, 1999.

[10] Q. Liang and J. M. Mendel, " Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. Fuzzy Systems.* vol. 8, no. 5, pp. 535-551, Oct 2000.

[11] J. M. Mendel, "Fuzzy Logic Systems for Engineering : A Tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.

[12] J. M. Mendel and G. Mouzouris, " Designing fuzzy logic systems ," *IEEE Trans. Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 44, no. 11, pp. 885-895, Nov 1997.

[13] E. H. Mamdani, " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 12, pp. 1182-1191, 1977.

[14] N. N. Karnik, J. M. Mendel and Q. Liang " Type-2 Fuzzy Logic Systems", *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 6, Dec. 1999.

[15] S. Meguerdichain, F. Koushanfar, G. Qu and M. Potkonjak, " Exposure in wireless ad-hoc sensor networks ," *Mobicom 2001*, Rome, Italy, July 2001.

[16] W. Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles*, Upper Saddle River, NJ, 1998.

[17] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level," *IEEE Trans. on Networking*, vol. 5, no. 1, pp. 71-86, Feb 1997.

[18] L. A. Zadeh, " The concept of a linguistic variable and its application to approximate reasoning - I," *Information Sciences*, vol. 8, pp. 199-249, 1975.

List of Tables

Table 1: The parameters and number of parameters in type-1 and interval type-2 FLSs.

FLS	type-1	interval type-2
Parameters in one antecedent	$m_{\mathrm{F}_k^i}$, $\sigma_{\mathrm{F}_k^i}$	$m_{\tilde{\mathrm{F}}_{k1}^i}$, $m_{\tilde{\mathrm{F}}_{k2}^i}$, $\sigma_{\tilde{\mathrm{F}}_k^i}$
Parameters in one consequent	$\overline{y}^i$	y_l^i, y_r^i
Total number of Parameters	144	224

Table 2: Initial values of the parameters in type-1 and interval type-2 FLSs. Each antecedent is described by two fuzzy sets.

	Type-1 FLS	Interval Type-2 FLS
mean	$m_t - 2\sigma_t$ or $m_t + 2\sigma_t$	$[m_t - 2.5\sigma_t, m_t - 1.5\sigma_t]$ or $[m_t + 1.5\sigma_t, m_t + 2.5\sigma_t]$
σ	$\sigma_{\mathrm{F}_k^i} = 2\sigma_t$	$\sigma_{\tilde{\mathrm{F}}_k^i} = 2\sigma_t$
consequent	$\overline{y}^i \in [min, max]$	$y_l^i = \overline{y}^i - \sigma_t$, $y_r^i = \overline{y}^i + \sigma_t$

Table 3: Average absolute error between the forecasted and actual time stamp of the starting edge of events in type-1 FLS and interval type-2 FLS. Here, 1 stands for one sample or $1024ms$, m and σ stands for the mean and the standard deviation of the cumulated signal strength of the training data respectively.

	type-1 FLS	interval type-2 FLS
double sliding window	7.8	1.4
SS with $th = m$	57.8	18.7
SS with $th = m + \sigma/2$	24.4	13.0
SS with $th = m + \sigma$	20.5	16.6

List of Figures

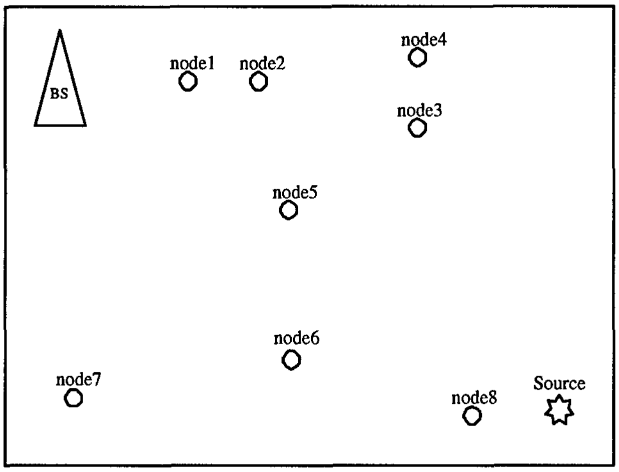


Figure 1: The deployment of eight sensor nodes in our experiments.

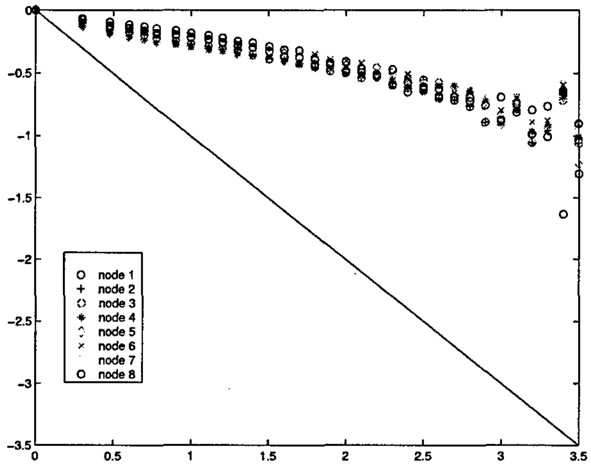


Figure 2: The *variance-time* plot for sensed signal strength with fixed source as background during 3 hours. The sample period is 1024ms.

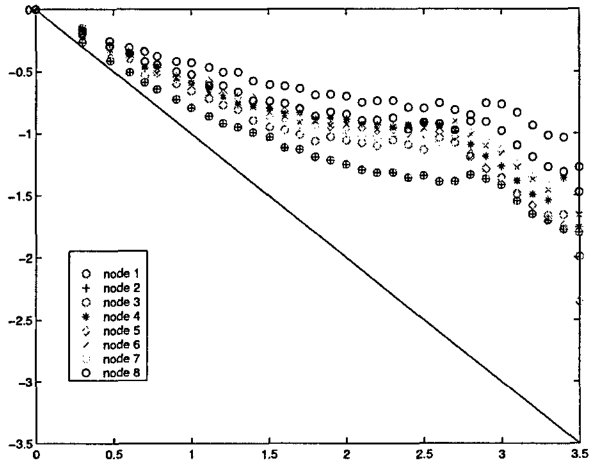


Figure 3: The *variance-time* plot for sensed signal strength without fixed source during 3 hours. The sample period is 1024ms.

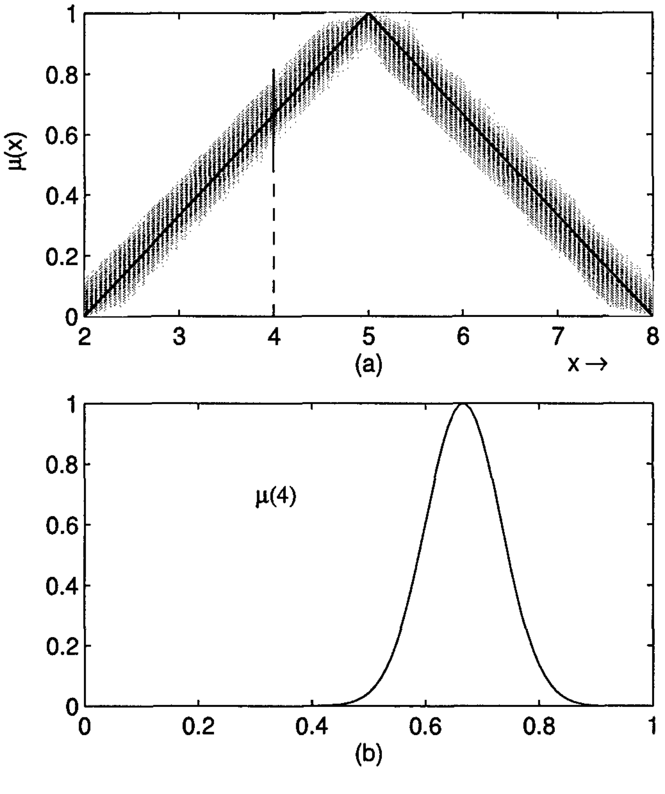


Figure 4: (a) Pictorial representation of a Gaussian type-2 set. The secondary memberships in this type-1 fuzzy set are shown in (b), and are Gaussian. Note that this set is called a Gaussian type-2 set because all its secondary membership functions are Gaussian. The "principal" membership function (the bold line), which is triangular in this case, can be of any shape.

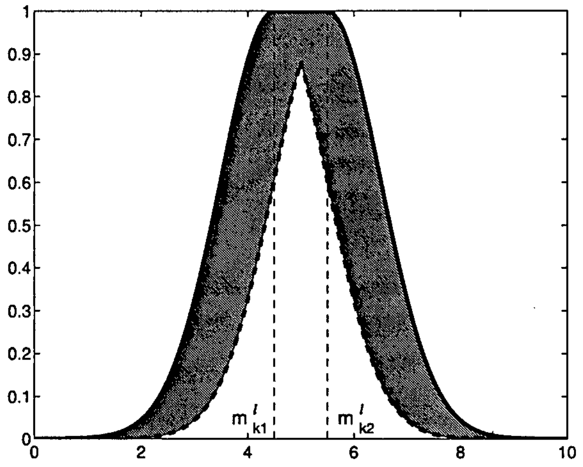


Figure 5: The interval type-2 MFs with fixed std and uncertain mean.

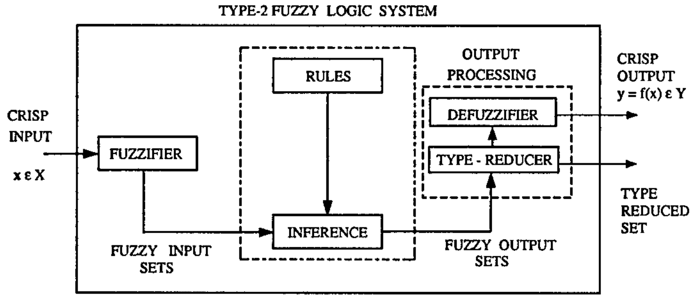


Figure 6: The structure of a type-2 FLS. In order to emphasize the importance of the type-reduced set, we have shown two outputs for the type-2 FLS, the type-reduced set and the crisp defuzzified value.

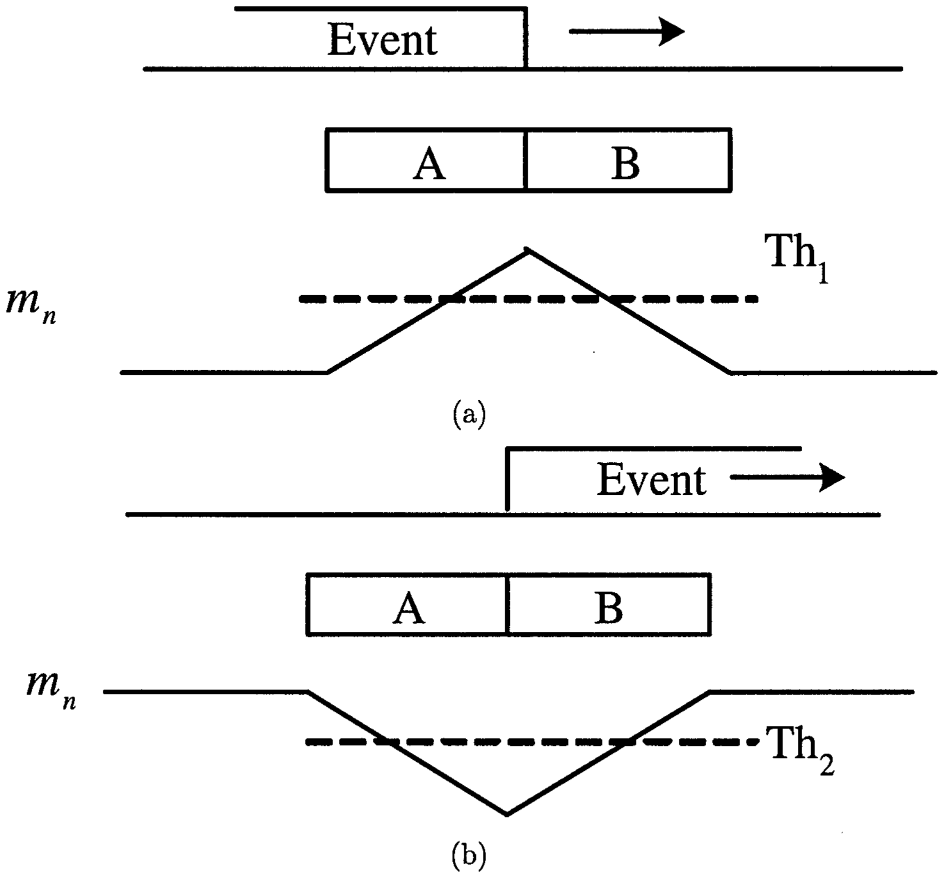


Figure 7: The response of the double sliding window event-detection algorithm. (a) starting edge of the event, and (b) ending edge of the event.

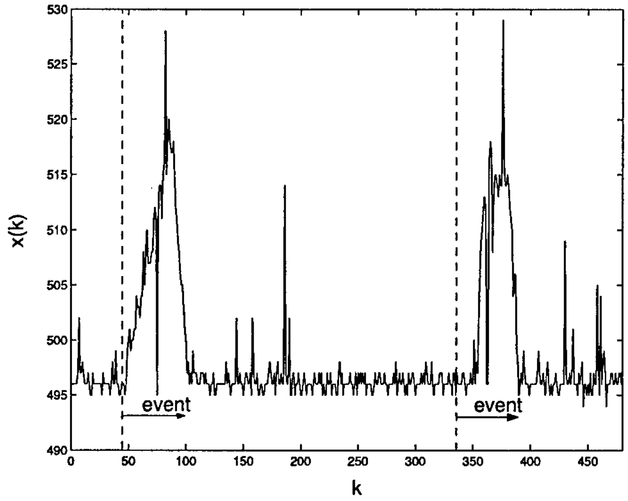


Figure 8: Sensed data for 1024 seconds. (The sample period is 1024ms).

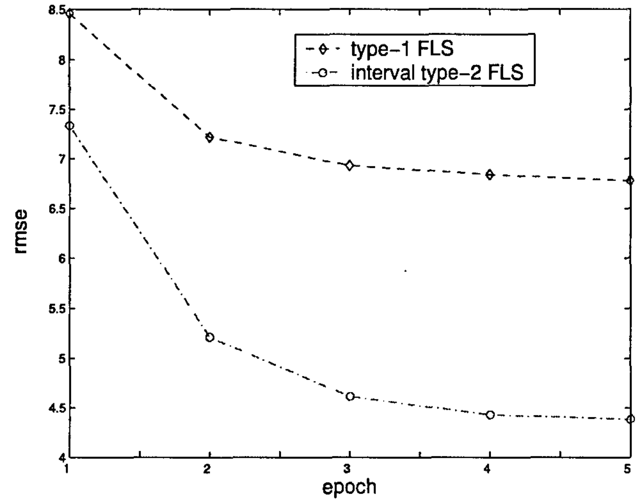


Figure 9: The RMSE of sensed signal strength forecasting of two FLSs averaged over 100 Monte-Carlo realizaions.

Redundancy Reduction in Wireless Sensor Networks Using SVD-QR

Qilian Liang and Lingming Wang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016. USA
E-mail:liang@uta.edu. wang@wcn.uta.edu

Abstract— **In densely deployed wireless sensor networks, not only does the data of one sensor node have self-similarity, but the data from adjacent sensor nodes also have cross-similarity. Therefore, it is clear that there exists highly redundancy in the collected data from sensor nodes in the neighborhood. Due to the intrinsic properties wireless sensor networks have, *e.g.*, energy constraint, bandwidth limitation, this kind of information redundancy will impact the whole networks in a negative way. In this paper, we propose to use Singular-Value-QR Decomposition (SVD-QR) to reduce the redundancy in wireless sensor networks.**

I. Introduction

Wireless sensor networking is an emerging technology that promises unprecedented ability to monitor and manipulate the physical world via a network of densely distributed wireless sensor nodes. The nodes can sense the physical environment in a variety of modalities, including acoustic, seismic, thermal, and infrared. They are networked together in an ad hoc fashion, which involves peer-to-peer communication in a network with a dynamically changing topology. Wireless sensor networks do not rely on a preexisting fixed infrastructure, such as a wireline backbone network or a base station. They are self-organizing entities that are deployed on demand in support of various events such as security and surveillance, monitoring of wildlife habitats, smart sensor-instrumented environments, and condition-based maintenance of complex systems, etc.

Sensor nodes are typically powered by small batteries that are hard to replace or recharge. Hence, how to efficiently use the sensor nodes, *e.g.* , not lose essential information but extend the lifetime of the nodes as long as possible, is an important issue.

Usually, in wireless sensor networks, sensor nodes are densely deployed, *e.g.*, tens of sensor nodes per square meters [6], therefore the information data collected from adjacent sensor nodes might be very similar with each other, that also means there exists redundancy among those information. Taking advantage of this property, we propose to reduce the redundancy so as to prolong the lifetime of the whole networks by using Singular-Value-QR Decomposition (SVD-QR).

In this paper, we use Xbow wireless sensor network professional developer's kit MOTE-Kit[5] as our testbed to get data sets from different scenarios. In the following sections, Section II studied the self-similarity of sensed data; the redundancy reduction for wireless sensor networks using SVD-QR is presented in Section III; and conclusions and future works are provided in Section IV.

II. Self-Similarity of Sensor Network Data

For a detailed discussion on self-similarity in time-series, see [8] [7]. Here we briefly present its definition [4].Given a zero-mean, stationary time-series $X = (X_t; t = 1, 2, 3, \cdots)$, we define the m-aggregated series $X^{(m)} = (X_k^{(m)}; k = 1, 2, 3, \cdots)$ by summing the original series X over nonoverlapping blocks of size m. Then it's said that X is H-self-similar, if, for all positive m, $X^{(m)}$ has the same distribution as X rescaled by m^H. That is,

$$X_t \triangleq m^{-H} \sum_{i=(t-1)m+1}^{tm} X_i \quad \forall m \in N \tag{1}$$

If X is H-self-similar, it has the same autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)]/\sigma^2$ as the series $X^{(m)}$ for all m, which means that the series is distributionally self-similar: the distribution of the aggregated series is the same as that of the original.

Self-similar processes can show *long-range dependence*. A process with long-range dependence has an autocorrelation function $r(k) \sim k^{-\beta}$ as $k \to \infty$, where $0 < \beta < 1$. The degree of self-similarity can be expressed using *Hurst* parameter $H = 1 - \beta/2$. For self-similar series with long-range dependence, $1/2 < H < 1$. As $H \to 1$, the degree of both self-similarity and long-range dependence increases.

One method that has been widely used to verify self-similarity is the *variance-time plot,* which relies on the slowly decaying variance of a self-similar series. The variance of $X^{(m)}$ is plotted against m on a log-log plot, and a straight line with slope $(-\beta)$ greater than -1 is indicative of self-similarity, and the parameter H is given by $H = 1 - \beta/2$. We use this method to verify the self-similarity of acoustic signal.

In our experiments, 8 sensors were deployed in a lab. The location of the sensors is showed in Fig. 1. We designed two scenarios, one is with a fixed source, and the other is without. In Fig. 2, we plot the variance of $X^{(m)}$ against m on a log-log plot for 8 sensor nodes respectively in the first scenario and Fig. 3 is under the second scenario. In order to prove that the data from all the sensor nodes have self-similarity as well, we mixed the data sets together to get a new time series as $Y = (X_t^1, X_t^2, \cdots, X_t^8; t = 1, 2, 3, \cdots)$. We test its self-similarity by plotting the variance-time curve in Fig 4 as well. From these three figures, it's very clear that the no matter under what kind of condition both the single sensor network data and the mixed sensor networks data have self-similarity because their traces have slopes much greater than -1.

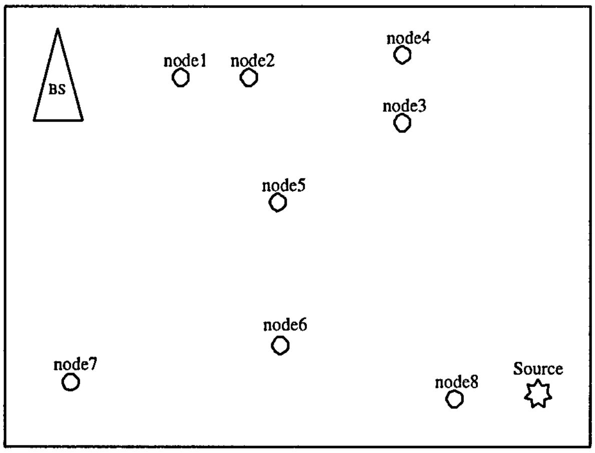


Fig. 1. The deployment of the eight sensor nodes in our experiments.

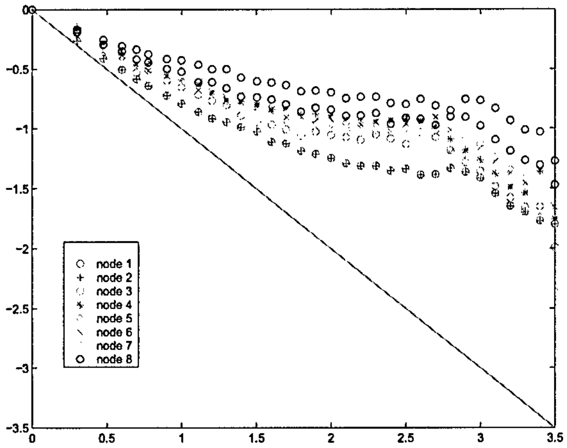


Fig. 2. The *variance-time* plot for sensed signal strength with fixed source as background during 3 hours. The sampling period is 1024ms.

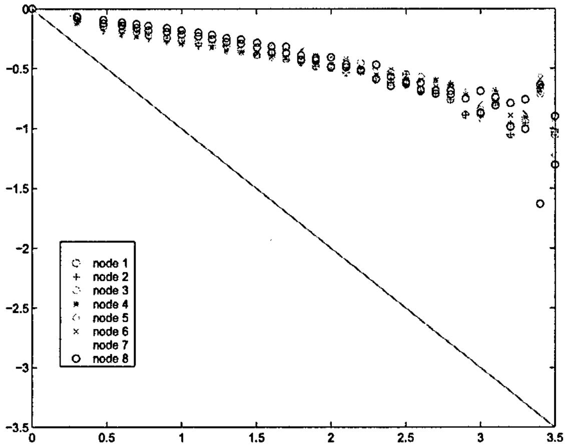


Fig. 3. The *variance-time* plot for sensed signal strength without fixed source during 3 hours. The sampling period is 1024ms.

III. Redundancy Reduction in Wireless Sensor Networks Using SVD-QR

In the previous section, we have proved that the data sets collected by adjacent sensor nodes are quite similar with each other. It is clear that there exists redundancy among the collected information. Therefore, two questions are popping up. Is such kind of redundancy profitable? Does more copies of the data set mean better estimates? The answers are both no. The goal of wireless sensor networks is to monitor the physical world, provide enough information in which users are interested so that users can perform further tasks, *e.g.*, events detection, targets estimating and tracking. Blair and Bar-Shalom have already demonstrated in [9] that more data from more sensor nodes doesn't mean better performance in terms of the maximum root-mean square errors(RMSE). Meanwhile, if we can get enough information from less sensor nodes, we can turn off the other sensor nodes so as to preserve energy and prolong the lifetime of the whole networks.

How to select the principal nodes to effectively represent the whole neighborhood? We view the data from all the adjacent sensor nodes as a matrix P, each column of P is the data from one sensor node, each row of P is the data collected at one epoch from all the sensor nodes. Therefore, the principal nodes picking problem can be simplified as subset selection.

Several subset selection methods exist [1], but a singular value decomposition (SVD) method is preferable in rank deficient problems [2]. Furthermore, the SVD provides a natural way to separate a space into dominant and subdominant subspaces. If we view the data matrix P as a span of the input subspace, then the SVD decomposes

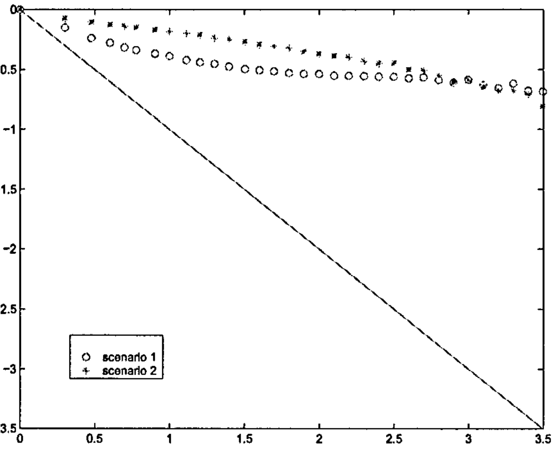


Fig. 4. The *variance-time* plot for mixed sensor data during 3 hours. The sampling period is 1024ms.

the span into an equivalent orthogonal span, from which we can identify the dominant and subdominant spans. In this way, we solve two problems simultaneously: (i) we estimate the data sets from *how many* sensor nodes are needed to effectively represent the neighborhood, and, (ii) we identify the data sets from *which* sensor nodes are the principal ones. The remainder can be discarded, and those sensor nodes can be turned off to conserve the energy.

A. Introduction of SVD-QR Algorithm

Here, we use the following SVD-QR algorithm that is similar to the one in [2] and [3] to select a set of independent data sets that minimize the residual error in a least-squares sense:

1) Given $P \in R^{N \times M}$, assume $N > M$, and $rank(P) = r \leq M$ denote the rank of P. Determine a numerical estimate r' of the rank of the data sets matrix P by calculating the singular value decomposition
$$P = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T, \tag{2}$$
where, U is an $N \times N$ matrix of orthonormalized eigenvectors of PP^T, V is an $M \times M$ matrix of orthonormalized eigenvectors of $P^T P$, and Σ is the diagonal matrix $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$, where σ_i denotes the i^{th} singular value of P, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. Select $\hat{r} \leq r'$.
2) Calculate a permutation matrix Π such that the columns of the matrix $\Gamma_1 \in R^{N \times \hat{r}}$ in
$$P\Pi = [\Gamma_1, \Gamma_2] \tag{3}$$
are independent. The permutation matrix Π is obtained from the QR decomposition of the submatrix comprised of the right singular vectors, which correspond to the $\hat{r}$ ordered *most-significant* singular values.

In short, we select the data sets as the following:

- Decomposes P, from the SVD of P, save V.
- Observe Σ. Select an appropriate $\hat{r}$.
- Partition
$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \tag{4}$$
where $V_{11} \in R^{\hat{r} \times \hat{r}}$, $V_{12} \in R^{\hat{r} \times (M-\hat{r})}$, $V_{21} \in R^{(M-\hat{r}) \times \hat{r}}$, and $V_{22} \in R^{(M-\hat{r}) \times (M-\hat{r})}$. In many practical cases, σ_1 is much larger than $\sigma_{r'}$; thus $\hat{r}$ can be chosen much smaller than the estimate r' of rank(P), even 1.
- Using QR decomposition with column pivoting, determine Π such that
$$Q^T [V_{11}^T, V_{21}^T] \Pi = [R_{11}, R_{12}], \tag{5}$$
where Q is a unitary matrix, and R_{11} and R_{12} form an upper triangular matrix; and Π is the permutation matrix, the column permutation Π is chosen so that $abs(diag(R))$ is decreasing. In short, Π corresponds to the $\hat{r}$ ordered *most-significant* sets.

B. An example of the SVD-QR decomposition in Redundancy Reduction

Here, we give an example of how to use SVD-QR decomposition to reduce the redundancy in wireless sensor networks, *i.e.*, determine *how many* sensors of data should be selected.

We use the data sets which also has been used in Section II, and get one clip, *i.e.*, 8 sensor nodes, each one has 100 samples of data, as the input of the following example.

Example 1

- *step 1.* SVD the input matrix P, get:
$diag(\Sigma) = (14160, 74, 20, 14, 13, 10, 9, 7)$;
Clearly, $\Sigma(1,1)$ is much larger than $\Sigma(2,2)$. That means we can only select one data set to represent all the eight sets of data, *i.e.*, $\hat{r} = 1$.
- *step 2.* Partition the V, and get V_{11} and V_{21}, which are needed in QR decomposition,
$V_{11} = -0.3565$, and
$$V_{21} = \begin{bmatrix} -0.3556 \\ -0.3535 \\ -0.3512 \\ -0.3546 \\ -0.3526 \\ -0.3540 \\ -0.3503 \end{bmatrix}.$$
- *step 3.* Using QR decomposition with column pivoting to determine the economy matrix Π. Since in this

example, $\hat{r} = 1$, we only care about the first column of Π,

$$\Pi(:,1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

That means the first column of the input matrix P, *i.e.*, the data collected from the first sensor node is the *most-significant* one, which can effectively represent all the eight sensor nodes in the neighborhood. $\sharp$

Example 2

What if we select more than one set of data? We have the following example to explain. We get another clip of data, still has 8 sensor nodes, each one has 100 samples of data, as the input.

- *step 1.* SVD the input matrix P, get: $diag(\Sigma) = (14759, 368, 275, 200, 186, 146, 97, 68)$. Oberserve Σ, the decreasing scope from $\Sigma(1,1)$ to $\Sigma(2,2)$ is not as large as it is in *Example 1* . So, we have $\hat{r} = 2$.
- *step 2.* Partition the V, and get V_{11} and V_{21},
$$V_{11} = \begin{bmatrix} -0.3503 & 0.1919 \\ -0.3582 & -0.1618 \end{bmatrix}, \text{ and}$$
$$V_{21} = \begin{bmatrix} -0.3528 & 0.3369 \\ -0.3570 & -0.8685 \\ -0.3580 & -0.1417 \\ -0.3525 & 0.0853 \\ -0.3585 & 0.1548 \\ -0.3406 & 0.1338 \end{bmatrix}.$$
- *step 3.* Using QR decomposition with column pivoting to determine the economy matrix Π. Since $\hat{r} = 2$, we only care about the two column of Π,
$$\Pi(:,1:2) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$
That means the forth and third columns of the input matrix P, *i.e.*, the data collected from the forth and third sensor nodes are the *most-significant* ones, which can effectively represent all the eight sensor nodes in the neighborhood. $\sharp$

From our plenty of simulations, the significant one(s) are changing, that depends on the change of the environment. However, we can define a *coherent time* , in this time, the environment is assumed to keep stable to a certain extend.

IV. Conclusions and Future Works

In this paper, we used MOTE-Kit[5] testbed to collect the real data sets from different scenarios. First, we proved there exists not only self-similarity in the data from one sensor node, but cross-similarity among the data of all the adjacent sensor nodes also. That demonstrated that there exists redundancy in the collected data of the wireless sensor networks. Taking energy efficiency and better performance into consideration, we proposed to use SVD-QR to select the principal data sets from particular sensor nodes to represent the all the sensor nodes in the neighborhood effectively. We gave two examples to show how to do it.

The future work includes theoretical analysis on how much information loss after we reduce the redundancy? Does this loss affect the performance of the wireless sensor networks?

References

[1] Chen, S., S. A. Billings and W. Luo, "Orthogonal Least Squares Methods and their Application to Nonlinear System Identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873-1896, 1989.

[2] Golub, G. H., and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, MD 1983.

[3] Golub, G.H., "Numerical Methods for Solving Least Squares Problems," *Numer. Math*, no. 7, pp. 206-216, 1965.

[4] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE Trans. on Networking*, vol. 5, no. 6, pp. 835-846, Dec 1997.

[5] J. L. Hill, and D. E. Culler, " Mica: A Wireless Platform for Deeply Embedded Networks ," *IEEE, Micro*, Volume: 22, Issue: 6, pp. 12-24, Nov.-Dec. 2002.

[6] R. Min, M. Bhardwaj, S. Cho, N. Ickes, E. Shin, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric Enabling Technologies for Wireless Sensor Networks," *IEEE Wireless Coomun.*, vol. 9, pp. 28-39, Aug 2002.

[7] W. Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles*, Upper Saddle River, NJ, 1998.

[8] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level," *IEEE Trans. on Networking*, vol. 5, no. 1, pp. 71-86, Feb 1997.

[9] W. D. Blair and Y. Bar-Shalom, "Tracking Maneuvering Targets with Multiple Sensors: Does More Data Always Mean Better Estimates?," *IEEE Trans. on Aerospace and Elec. Systems*, vol. 32, no. 1, pp. 450-455, Jan 1996.

CIHSPS 2005 - IEEE International Conference on
Computational Intelligence for Homeland Security and Personal Safety
Orlando, FL, USA, 31 March - 1 April 2005

Event Detection in Wireless Sensor Networks Using Fuzzy Logic System

Qilian Liang, Lingming Wang

[1]Department of Electrical Engineering,
University of Texas at Arlington,
Arlington, TX 76019-0016. USA
Phone: +1 817 2723488, E-mail: liang@uta.edu.

Abstract – *Wireless Sensor Networks (WSN) are designed to monitor physical phenomena. The main task of WSN is to perform event detection, tracking, and classification. So, compared with traditional ad-hoc networks, WSN is event-centric. Therefore, an important question in WSN is to detect events. In this paper, we present two methods to do event detection, one is Double Sliding Window Detection, the other one is Fuzzy Logic approach. The accuracy of the results is established via sensor network testbed and simulations.*

Keywords – *Wireless sensor networks, fuzzy logic systems, event detection.*

I. INTRODUCTION

The infusion and maturation of the Micro Mechanical System(MEMS), computations, and wireless communication technologies has advanced the development of Wireless Sensor Networks (WSN). In WSN, a large amount of low cost sensor nodes are densely deployed to monitor the environment of interest. Due to the various applications [2] [3], WSN has generated flurry of research activity.

Sensor nodes are typically powered by small batteries that are hard to replace or recharge. Hence, energy constraint is a unique character of WSN compared with traditional wireless ad-hoc networks. Energy comsuption occurs in three domains: sensing, data processing (including AD/DA and digital signal processing), and communications[5]. According to[1], the sensor, signal processing parts operate at low frequency and consume less than $1mW$. This is over an order of magnitude less than the energy consumption of the communication part. Therefore, we prefer less communication/data exchange between sensor nodes but more local processing implemented by one single sensor node so as to increase the lifetime of the WSN.

The main goal of WSN is to monitor physical world. Usually, people are more intereted in unexpected events. For example, in a scenario of battlefield, people are more interested in the appearance of enemies. If a WSN is to monitor forest-fire, unusual increasing of the temperature should be a necessary warning to people. Both the appearance of enemies and the unusual increasing of the temperature can be seen as events. Because of the energy constraint of WSN mentioned previously, the ideal state of WSN should be event-driven, so that we can power off the communication part at most of the time. Only when certain sensor nodes detect an event, they trigger the RF channel, and transmit the useful information to basestation or headquarters. Therefore, event-detection is one of the key issues for WSN.

In this paper we present two approaches of event-detection for WSN, double sliding window and hybrid event-detection using fuzzy logic system. We use Berkely MICA2 motes[4] as our testbed and evaluate the event-detection approaches based on the acoustic data collected by the testbed in different experiments.

The remainder of the paper is organized as follows. the sensor model is given in Section II. The double sliding window and hybrid event-detection based on fuzzy logic sytem approaches are presented in Section III and Section IV respectively. Simulation results and discussions are presented in section V. Section VI concludes this paper.

II. ACOUSTIC SENSOR MODEL

Acoustic amplitude sensor node measures sound amplitude at the microphone. Assuming that the sound source is a point source and sound propagation is lossless and isotropic, a root-mean-squared (RMS) amplitude measurement z is related to the sound source position X as

$$z = \frac{a}{\|X - \varsigma\|} + w, \qquad (1)$$

where a is the RMS amplitude of the sound source, ς is the location of the sensor, and w is RMS measurement noise [6]. In this paper, we use Xbow wireless sensor network professional developer's kit MOTE-Kit for data collection.

III. DOUBLE SLIDING WINDOW EVENT-DETECTION

In [9],the acoustic energy in a fixed period of time is integrated, when it exceeds a threshold, the authors claim a detection of event occurred, as:

$$E_s = \sum_{m=0}^{M-1} |z_{n-m}|^2 , \tag{2}$$

$$E_s \geq E_{threshold}. \tag{3}$$

However, this simple method suffers from a significant drawback; namely, the value of the threshold depends on the sensed signal energy. When there is no event occuring in the sensing range, the sensed signal consists of only noise. The level of the noise power is generally unknown and can change when the environment changes or if unwanted interferers go on and off. Therefore, it is quite difficult to set a fixed threshold. We design a double sliding window algorithm for event-detection so as to alleviate the threshold value selection problem.

The double sliding window event-detection algorithm calculates two consecutive sliding windows of the sensed signal energy. The basic principle is to form the decision variable as the ratio of the total energy contained inside the two windows. Figure 1 shows the windows A and B and the response of the ratio m_n to a sensed event. It can be seen that when only noise is sensed the response is flat, since both windows contain ideally the same amount of noise energy.

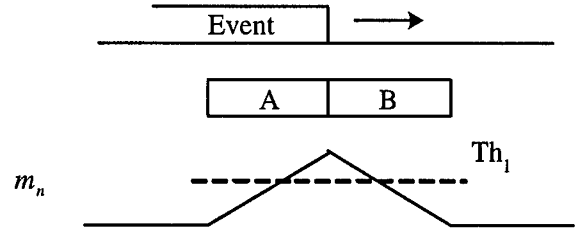


Fig. 1. The response of the double sliding window event-detection algorithm.

The calculation of the window A and window B value is shown as

$$E_a = \sum_{m=0}^{M-1} |z_{n-m}|^2 , \tag{4}$$

$$E_b = \sum_{m=1}^{M} |z_{n+m}|^2 . \tag{5}$$

Then the decision variable m_n is

$$m_n = \frac{E_a}{E_b}. \tag{6}$$

The advantage of this approach is the decision variable m_n does not depend on the sensed signal energy, but on the ratio of the energy of two consecutive windows.

IV. HYBRID EVENT-DETECTION BASED ON FUZZY LOGIC SYSTEM

Using the double sliding window algorith to do event-detection is a good approach. However, if an event continuously appears in the sensing range of a node, the ratio m_n will still be flat. The probability of detection will decrease accordingly. In order to solve this problem, we present a hybrid event-detection algorithm based on fuzzy logic system.

A. Overview of Fuzzy Logic Systems

Figure 2 shows the structure of a fuzzy logic system (FLS) [7]. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets. Consider a p-input 1-output FLS, using singleton fuzzification, *height* defuzzification [7] and "IF-THEN" rules of the form [8]

R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l.

Assuming singleton fuzzification, when an input $\mathbf{x}' = \{x_1', \ldots, x_p'\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{F_1^l}\left(x_1^{'}\right) \star \mu_{F_2^l}\left(x_2^{'}\right) \star \cdots \star \mu_{F_p^l}\left(x_p^{'}\right) = \mathcal{T}_{i=1}^{p} \mu_{F_i^l}\left(x_i^{'}\right) \tag{7}$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm. There are many kinds of defuzzifiers. In this paper, we focus, for illustrative purposes, on the height defuzzifier [7]. It computes a crisp output for the FLS by first obtaining the height, $\overline{y}^l$, of every consequent set G^l, and, then computing a weighted average of these heights. The weight corresponding to the lth rule consequent height is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p} \mu_{F_i^l}\left(x_i^{'}\right)$, so that

$$y_h(\mathbf{x}') = \frac{\sum_{l=1}^{M} \overline{y}^l \mathcal{T}_{i=1}^{p} \mu_{F_i^l}\left(x_i^{'}\right)}{\sum_{l=1}^{M} \mathcal{T}_{i=1}^{p} \mu_{F_i^l}\left(x_i^{'}\right)} \tag{8}$$

where M is the number of rules in the FLS. In this paper, we design a FLS for event-detection of WSN.

B. Hybrid event-detection algorithm

We have two inputs for the FLS:*the accumulated signal energy E_s in a fixed period of time* and *the ratio of the accumulated signal energy in two consecutive sliding windows m_n*. The linguistic variables used to represent them were divided into three levels: *low*, *moderate*, and *high*. The consequent – the possibility that an event occurs – was divided into five

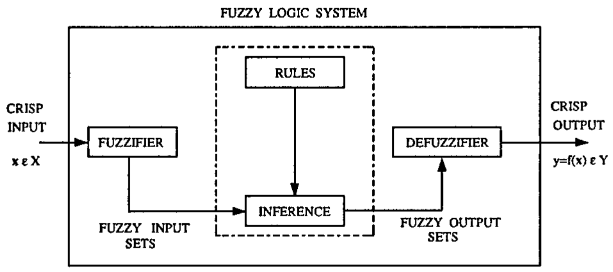


Fig. 2. The structure of a fuzzy logic system.

levels,*very strong*, *strong* , *medium* , *weak* and *very weak*. We used trapezoidal membership functions (MFs) to represent *low, high, very strong, very weak*; and triangle MFs to represent *moderate, medium , strong, weak*. We show these MFs in Figure 3(a) and 3(b).

Based on the fact that when event occurrs, E_s or m_n should be high. We design a fuzzy logic system using rules such as:

R^l : IF E_s is F_1^l and m_n is F_2^l, THEN the possibility that there is event (y) is G^l.

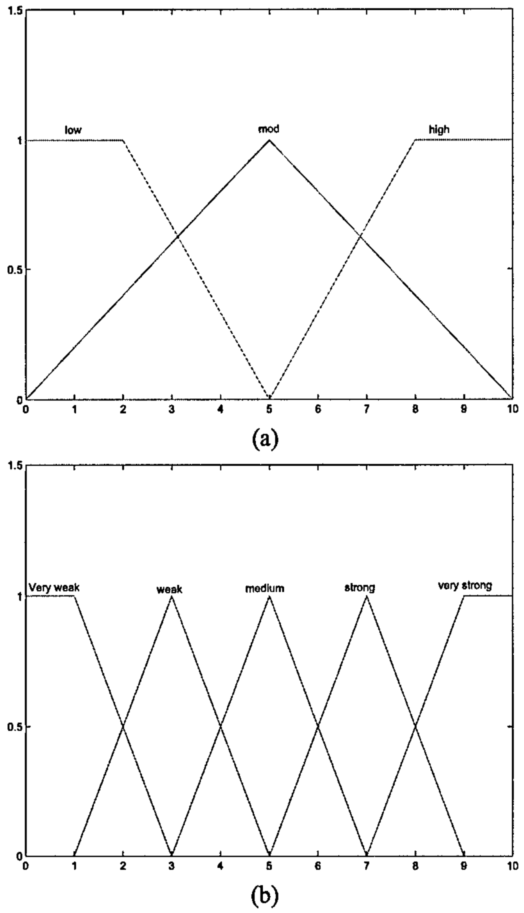
where $l = 1, \cdots, 9$. We summarize all the rules in Table 1.

Table 1. Rules for event-detection. Antecedent 1 is E_s, Antecedent 2 is m_n.

Rule	Antecedent 1	Antecedent 2	Consequent
1	low	low	very weak
2	low	mod	weak
3	low	high	mod
4	mod	low	weak
5	mod	mod	mod
6	mod	high	strong
7	high	low	mod
8	high	mod	strong
9	high	high	very strong

V. SIMULATIONS

Figure 4 shows the basic data set, which was colletecd from Berkely MICA2 motes, we used in our simulations. In order to get the probability of detection $P - d$, and probability of false alarm $P - f$, white Gaussian Noise is added, SNR is $10dB$. We ran 100,000 Monte-Carlo simulations. The results of each algorithms are summarized in Table 2. Obviously, in terms of both P_d and P_f, the performances of both Double Sliding Window scheme and hybrid event-detection algorithm based on FLS are much better than that of signal strength event-detection algorithm.



(a)

(b)

Fig. 3. MFs used to represent the linguistic labels. (a) MFs for antecedent, and (b) MFs for consequent.

VI. CONCLUSIONS

In this paper, we proposed two event-detection algorithms in Wireless Sensor Networks, Double Sliding Window scheme and hybrid approach based on Fuzzy Logic System. We use the basic data set collected by MOTE-Kit[4] testbed and white Gaussian Noise is added. Simulation results show that both the Double Sliding Window and the hybrid scheme based on FLS outperform the existing Signal Strength event-detection algorithm in terms of both the probability of detection and the probalibity of false alarm.

VII. ACKNOWLEDGEMENT

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

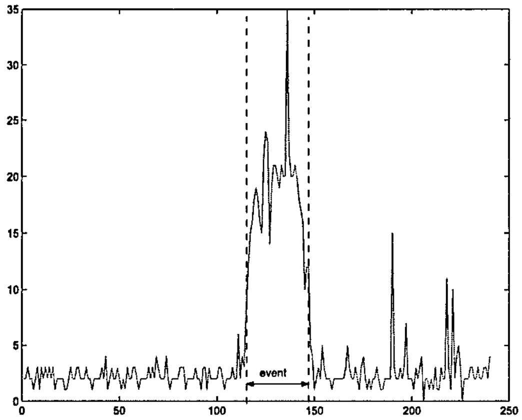


Fig. 4. 240 sensed data set, the sample period is 1024ms)

Table 2. Probabilities of detection and false alarms .

	P_d	P_f
Signal Strength event-detection	69.75%	0.08%
Double Sliding Window event-detection	91.499%	0.02%
Hybrid event-detection based on FLS	99.97%	0.05%

REFERENCES

[1] Andrew Y. Wang, and Charles G. Sodini, " A Simple Energy Model for Wireless Microsensor Transceivers ," *Globecom 2004, Dallas, TX.*,

[2] D. Estrin, and R. Govindan, " Next century chanlleges:scalable coordination in sensor networks ," *Mobicom 1999, Seattle, WA.*, pp. 263-270, August 1999.

[3] G. J. Pottie, and W. J. Kaiser, " Wireless integrated network sensors ," *Communications of the ACM*, vol. 43, no. 5, pp. 551-558, May 2000.

[4] Jason L. Hill, and David E. Culler, " Mica: A Wireless Platform for Deeply Embedded Networks ," *Micro, IEEE,*, Volume: 22, Issue: 6, pp. 12-24, Nov.-Dec. 2002.

[5] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie *Protocols for Self-Organization of a Wireless Sensor Network*, IEEE Personal Communications, October 2000.

[6] L. E. Kinsler, A. R. Frey, A. B.Coppens and J. V. Sanders, *Fundamentals of Acoustic*, John Wiley and Sons, Inc., New York, USA, 1999.

[7] J. M. Mendel, "Fuzzy Logic Systems for Engineering : A Tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.

[8] E. H. Mamdani, " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 12, pp. 1182-1191, 1977.

[9] S. Meguerdichain, F. Koushanfar, G. Qu and M. Potkonjak, " Exposure in wireless ad-hoc sensor networks ," *Mobicom 2001*, Rome, Italy, July 2001.

Spectrum Efficient Coding Scheme for Correlated Non-Binary Sources in Wireless Sensor Networks

Haining Shu and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016 USA
E-mail: shu@wcn.uta.edu, liang@uta.edu

Abstract— **Energy-aware technique to reduce energy consumption in distributed sensor networks has become a prominent topic in sensor network research. Various sensor network applications have taken energy efficiency into consideration. In the case of correlated binary sources, distributed source coding has been literally studied in information theory. However, data sources from real sensor networks are normally non-binary. In this paper, we proposed a spectrum efficient coding scheme for correlated non-binary sources in sensor networks. Our approach constructs the codeword cosets for the interested source, taking advantage of statistical characters of the distinct observations from sensor nodes. The coset leaders are then transmitted via the channel and decoding is performed with the available side information. Simulations are carried out over independent and identically distributed (i.i.d) Gaussian sources and data collected from Xbow wireless sensor network test bed. Simulation results show that the proposed scheme performs at 0.5 - 1.5 dB from the Wyner-Ziv distortion bound.**

I. Introduction

Wireless sensor network consists of certain amount of small and energy constrained nodes. Such networks are normally deployed for data collection where human intervention after deployment, to recharge or replace node batteries may not be feasible, resulting in limited network lifetime. Failure of an amount of sensors due to energy depletion has a significant impact on the functioning of the entire wireless sensor networks.

Various research has been done to alleviate the energy consumption in wireless sensor networks, from hardware design of individual sensor to routing and topology construction of the whole network. Among which, one distinct technology for energy-efficient wireless sensor networks is distributed source coding (DSC) [1], [2]. DSC was proposed to encode the correlated sensor readings separately, i.e. sensors encoding the readings do not communicate with each other. After the distributed encoding, the compressed data is sent to a central hub node for joint decoding. Further research on this topic demonstrated that convolutional codes [3], Turbo and LDPC codes [4], [5] performed well in distributed compression for sensor networks. All these approaches are based on binary distributed sources with refined correlation to each other. However, in a practical sensor network or even in a lab test bed of wireless sensor network, the distributed deployed sensors have very rough readings which can hardly be fitted into the above binary compressing schemes.

In this paper, we address the spectrum efficient coding scheme for correlated non-binary sources in wireless sensor networks. Our approach attempts to provide a solution to Chief Executive Officer (CEO) problem. The goal of the CEO problem is to recover as much information as possible about the actual event from the noisy observations, while minimizing the total information rate. We propose to exploit the statistical characters of real sensor readings before constructing codeword cosets. From the approximate Gaussian readings, Lloyd-Max quantization is applied to minimize the mean square distortion. To save communication spectrum, a coset encoder is designed to reduce the transmitted bits based on the probability distribution of quantized values. We show that source encoding can be completed in a fully distributed way. Each sensor encodes its own readings without knowing what the other sensors have measured. Our work differs from previous ones not only in the non-binary sources but in proposing a practical coset encoding scheme for real senor readings. Simulations are carried out over independent and identically distributed (i.i.d) Gaussian sources and data collected from Xbow wireless sensor network test bed. Simulation results show that the proposed scheme performs at 0.5 - 1.5 dB from the Wyner-Ziv distortion bound.

This paper is organized as follows. In section II, we briefly review the basic concept of distributed source coding for correlated information. Section III discusses the intuition behind our approach. Section IV details the coset construction based on the statistical knowledge of sensor readings. Simulation results are presented in Section V. Section VI concludes with a summary.

II. Prelimiaries

In this section, we review the basic concepts of distributed source coding for correlated information and introduce Slepian-Wolf coding for lossless source coding and Wyner-Ziv coding for the lossy case.

Consider a distributed wireless sensor network consisting of individual sensors that monitor the sensor field. These sensors transmit their highly correlated data to a central hub node to reconstruct the observations. Transmission of redundant information can be easily avoided if the sensors communicate with each other but such inter-node cooperation requires higher bandwidth and consumes more energy in communication. Slepian and Wolf in [6] proved that if no communication among the sensors, theoretically there was no loss in performance under certain conditions. After [6] the Slepian-Wolf theorem has been extended to the lossy coding of continuous-valued sources by Wyner and Ziv [7].

A. Slepian-Wolf Coding

Let X and Y be two correlated independent and identically distributed ($i.i.d$) binary sources. For lossless compression with $X' = X$ and $Y' = Y$ after decompression, we know from Shannon's source coding theory [8] that a rate given by the joint entropy $H(X, Y)$ of X and Y is sufficient if we are encoding them together.

Fig. 1 gives an example of joint encoding and distributed encoding of two binary sources. In Fig. 1 (a), encoder X compress X into $H(X)$ bits per sample and based on the complete knowledge of X at both encoder and decoder, Y is then compressed into $H(Y|X)$ bits per sample, while in Fig. 1 (b), encoder X and Y do not communicate and perform separate encoding.

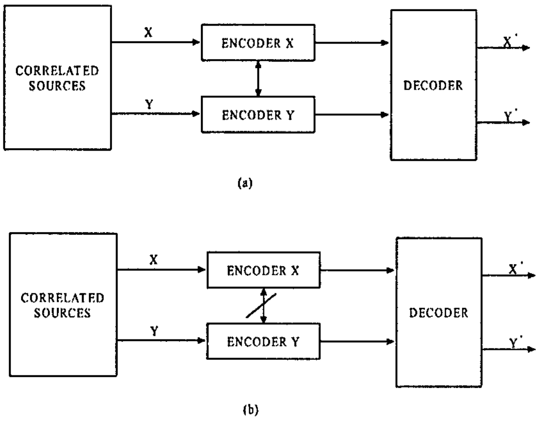
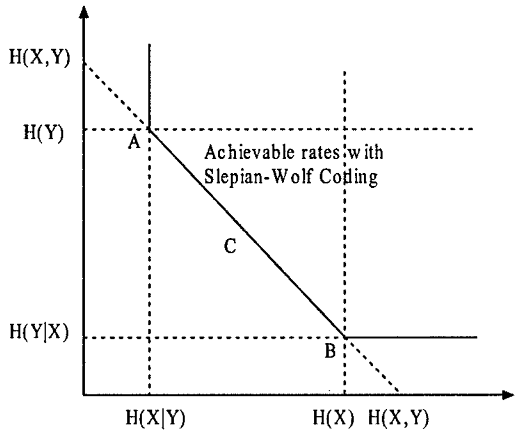


Fig. 1. Correlated source coding configuration. (a) Joint encoding of X and Y. The encoders communicate with each other and a rate $H(X,Y)$ is sufficient. (b) Distributed encoding of X and Y. The encoders do not communicate. Slepian-Wolf theorem proved that $H(X,Y)$ is also sufficient.

The Slepian-Wolf theorem [6] states that if X and Y are correlated according to some arbitrary probability distribution $p(x, y)$, then X can be compressed separately (without access to Y) without losing performance comparing to the condition in Fig. 1 (a). It says that the achievable region of DSC for discrete sources X and Y is given by $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ and $R_X + R_Y \geq H(X, Y)$, which is shown in Fig. 2.

For practical Slepian-Wolf coding, the first attemp is to approach the corner point A in the Slepian-Wolf rate region of Fig. 2 with $R_1 + R_2 = H(X|Y) + H(Y) = H(X, Y)$. This is actually a problem of source coding of X with side information Y at the decoder as shown in Fig. 3. Similarly the other corner point B of the Slepian-Wolf rate region can be approached by exchanging the roles of X and Y and all points between the two corner points can be realized by time-sharing.



Point A: compression of X with side information Y at the joint decoder

Fig. 2. The Slepian-Wolf region for two binary sources

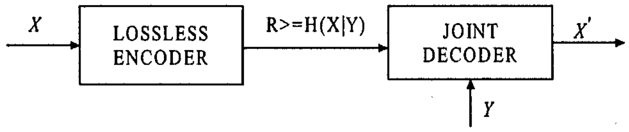


Fig. 3. One example of Slepian-Wolf coding: Lossless source coding with side information at the decoder

B. Wyner-Ziv Coding

Slepian-Wolf scheme focused on lossless source coding of discrete sources with side information at the decoder. However most sensor network applications deal with continuous sources, the rate distortion with side information at the decoder thus becomes a big concern. The problem to solve in the lossy source coding is how many bits are needed to encode X under the constraint that the average distortion between X and X' is $E[d(X, X')] \leq D$, assuming the side information Y is available only at the decoder.

Wyner and Ziv [7] first considered this problem ȧnd gave the rate-distortion function $R^*_{WZ}(D)$ for both discrete and continuous cases and general distortion metrics $d(.)$. Fig. 4 is an illustration of Wyner-Ziv coding. In general, Wyner-Ziv coding set up the Slepian-Wolf coding in that coding of X is with respect to a fidelity criterion rather than lossless.

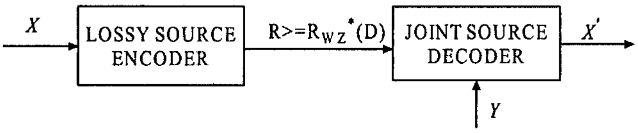


Fig. 4. Wyner-Ziv coding or lossy source coding with side information

But the important thing about Wyner-Ziv coding is that it normally suffers rate loss when compared to lossy coding of X

as the side information Y is available at both the encoder and decoder. One exception is when X and Y are jointly Gaussian which is of special interest in practice since many image and video sources can be modeled as jointly Gaussian.

Since we are introducing distortion to the source with Wyner-Ziv coding, quantization is needed in source coding. Usually there is still certain correlation in the quantized version of X and the side information Y, thus Slepian-Wolf coding could be employed to reduce the rate. In this case, the side information Y is used in jointly decoding and estimating X' at the decoder to help reduce the distortion $d(X, X')$ for non-binary sources. Fig 5 depicts the block diagram of a generic Wyner-Ziv coder.

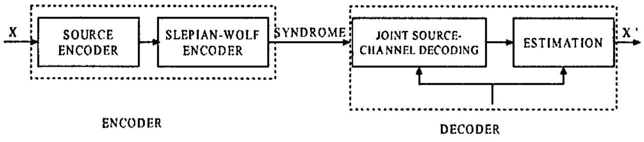


Fig. 5. Block diagram of a generic Wyner-Ziv coder

III. Intuition behind approch

In the above section, we discussed lossless (Slepian-Wolf) and lossy (Wyner-Ziv) source coding with side information available only at the decoder. Most of the work in DSC so far has been focusing on the two problems. In wireless sensor network, employing current DSC schemes requires the sensor nodes transmitting correlated information to cooperate in a small group so that one node provides side information and others compress the information down to the Slepian-Wolf or the Wyner-Ziv limit.

Tha major concern for practical application of DSC is the correlation model. Theoretically, two correlated non-binary sources can be constructed easily. An example with uniform distribution is shown as follows:

- Let $X = X_0X_1...$ and $Y = Y_0Y_1...$ be two correlated non-binary sequences taking values in $[L, R]$.
- Generate the $i.i.d$ sequence X using the probability distribution $P(X_k = i) = 1/(R - L)$ where $i \in [L, R]$.
- Define the sequence Y from the sequence X using the conditional probability distribution $P(Y_k = j|X_k = i) = p_{ij}$, where $i, j \in [L, R]$. The joint probability distribution between sources will be denoted by $P(X_k = i, Y_i = j) = p_{ij}/(R - L)$.

Although significant efforts have been put in DSC design for various correlation models, in real sensor network there still exist many situations that is hard to come up with certain joint probability functions. For instance, the correlation statistics of the video surveillance networks can be mainly a function of the sensors' location. Fig 6 is another example of the noisy versions of the acoustic signal strength collected using the Xbow wireless sensor network professional developer's kit MOTE-Kit.

In this paper, we address the issue of lossy coding for correlated non-binary sources in the Xbow wireless sensor networks. We are interested in the measurement noise in wireless sensor network specifically in the Chief Executive Officer (CEO) problem [9]. In this particular application, for example, the CEO of a company employs a number of agents to observe an event and each of the agents provides the CEO with his/her noise version the event. The agent are not allowed to convene, and the goal of the CEO is to recover as much information as possible about the actual event from the noisy observations received from the agents, while minimizing the total information rate from the agents. The CEO problem can then illuminate the measurement noise at the sensor node.

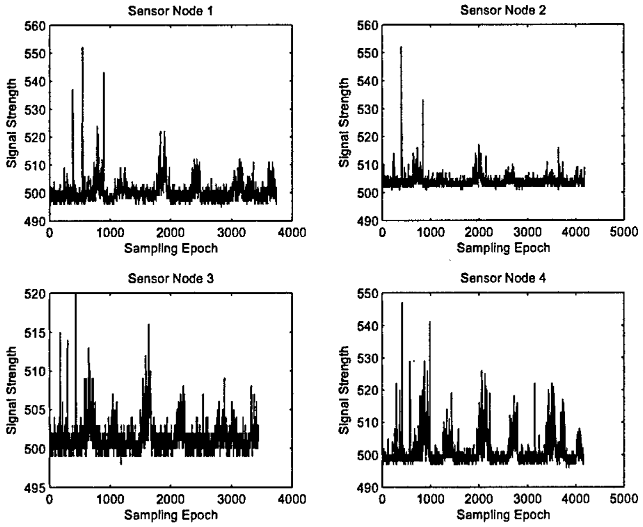


Fig. 6. Noisy observations of acoustic signal strength from four distributed sensors. The four sensors are not in equal distance to the acoustic source in network.

Preliminary practical code constructions for the CEO problem appeared in [10], [11], based on the Wyner-Ziv coding approaches, but they are only limited to special cases. Fig 7 is a CEO example in wireless sensor network where the central hub node is responsible to recover the information from the noisy measurements.

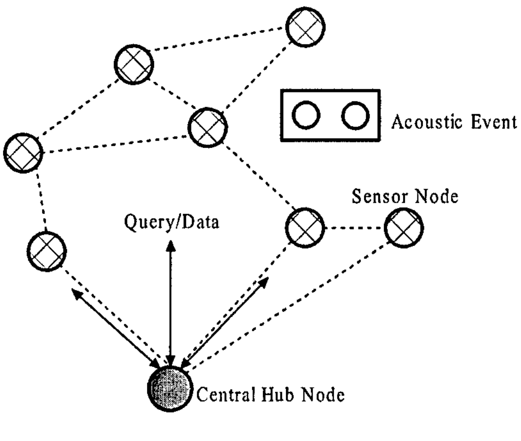


Fig. 7. A CEO example of sensor network. The central hub node broadcasts the queries and collects the noisy observations from the sensors.

IV. Construct Codeword Cosets

For correlated binary sources X and Y, Y is a noise corrupted version of X as $Y = X + N$, where N is an additive Gaussian noise. The correlation between the interested output X and the side information Y can be modeled with a "virtual" correlation channel, then a good channel code over this channel can provide us with a good Slepian-Wolf codes. In a sense, the seemingly source coding problem of Slepian-Wolf coding can be considered as a channel coding problem.

In this section, we detail our spectrum efficient coding scheme for correlated non-binary sources in wireless sensor networks. For interested information X, the encoder side consists of two parts: source encoder and coset encoder. We apply Lloyd-Max quantization in souce encoder which conducts the design of the initial codebook. The non-binary sources are then represented by the binary codewords according to the quantization levels. A coset encoder is constructed to save transmitting bits over channels. A n-bit codeword is transmitted by a m-bit $(m < n)$ coset leader to achieve a compression ratio of $n : m$ after the coset encoder. Side information Y will be transmitted at full rate, i.e. not through the coset encoder. The block diagram of our coding scheme is illustrated in Fig 8.

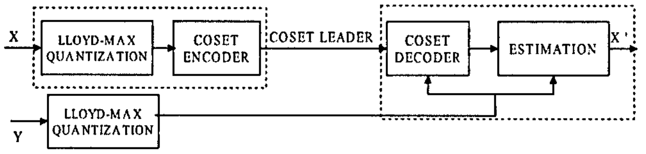


Fig. 8. Block diagram of the asymmetric coding scheme for correlated non-binary sources

We next give an example of constructing the coset encoder.

Example 1: Construct Codeword Cosets with Hamming Distance $d_H = 3$

For 8-level Lloyd-Max quantization, the input to the coset encoder is a 3-bit binary codeword $X_Q \in [000, 001, 011, 010, 110, 111, 101, 100]$. Assuming the Hamming distance between X_Q and the quantized binary side information Y_Q is $d_H(X_Q, Y_Q) \leq 1$, the cosets for X_Q can be constructed using the parity-check matrix H

$$H = \begin{bmatrix} 1\,1\,0 \\ 1\,0\,1 \end{bmatrix} \tag{1}$$

Four coset sets are constructed as $C1 = [000, 111]$, $C2 = [001, 110]$, $C3 = [010, 101]$ and $C4 = [011, 100]$. The transmitted coset leader X_C is associated with the syndrome $s = X_Q H^T$. Sending the 2-bit coset leader instead of the original 3-bit X_Q achieves a compression ratio of 3 : 2.

□

Now let us consider the noisy observation from sensor node 1 (see Fig 6 (a)) as the interested information X. Results from 8-level Lloyd-Max quantization are presented in Table I.

TABLE I

Results from 8-level Lloyd-Max Quantization

Codebook	Occurring Probability	Binary Codebook
498.09	0.4923	000
500.37	0.2809	001
503.06	0.1590	011
507.3	0.0457	010
511.31	0.0136	110
515.26	0.0051	111
523	0.0027	101
544	0.0008	100

From Table I, the first codeword after quantization 498.09 occurs at a dominant probability of 49.23%. The probability of occurence decreases dramatically along the initial codebook. We assign the binary codewords such that along the probability decreasing, every adjacent codeword differs in only 1 bit.

Suppose sensor node 3 (see Fig 6 (c)) is transmitting the side information Y for decoding. Data from sensor node 3 is quantized separately using Lloyd-Max quantizer. Now we have X_Q and Y_Q at 3-bit correlated binary codewords. Perfect coset encoder [1] requires that X_Q and Y_Q are correlated in the way that the Hamming distance between X_Q and Y_Q is no more than one. Then the cosets for X_Q are constructed that the elements within each coset have maximal Hamming distance $d_H = 3$ as depicted in example 1.

In our work, the correlation between X_Q and Y_Q is unknown or can hardly reach the perfect correlation. But with the knowledge of the codewords probability distribution, the coset construction could be done in a different way.

We propose to design the coset sets minimizing the overall *cross ratio*. We define the *cross ratio* as the ratio that within one coset, the codeword with less occuring probability will cross the other. We intend to decrease the decoding failure by reducing the *cross ratio* while keeping the Hamming distance within each coset as large as possible. Table II gives the *cross ratio* of two different coset sets.

TABLE II

Collision Ratio of Two Cosets Sets

Coset Set 1	Cross Ratio	Coset Set 2	Cross Ratio
(000, 111)	0.01	(000, 110)	0.027
(001, 110)	0.046	(001, 111)	0.018
(011, 100)	0.01	(011, 101)	0.017
(010, 101)	0.056	(010, 100)	0.017
Overall	0.0305	Overall	0.01975

From Table II, we see that coset set 2 has less *cross ratio* even though the Hamming distance within each coset is $d_H = 2$ but not 3.

The parity-check matrix to construct the codeword coset 2 with Hamming distance $d_H = 2$ is shown as bellow:

$$H = \begin{bmatrix} 0\,0\,1 \\ 1\,1\,0 \end{bmatrix} \tag{2}$$

At the decoder, we use the side information Y_Q to look for the most-likely codeword from the coset represented by the transmitted coset leader. The decoder then get the optimal estimation of X using all received information.

V. Simulation Results

Our simulations are performed over the acoustic noisy observations from the Xbow wireless sensor network professional developer's kit MOTE-Kit. We collected 8 sets of acoustic noisy version from 8 distributed deployed sensors in a lab. Information from the sensor closest to the acoustic source is set as side information for decoding. All others are encoded separately and reconstructed at the decoder with the side information. The correlation-in-dB between the interested information and the side information is presented in Table III.

TABLE III
Correlation-in-dB between X and Y

Sensor Node	Correlation-in-dB
1	5.7988
2	5.0864
3	6.3903
4	4.2262
5	4.2343
6	4.1522
7	5.5238

Due to the packet loss in data collecting at the central hub node, the correlation between the interested information X and the side information Y from sensor node 8 is pretty low. We choose two sensor nodes (node 1 and node3) with the highest correlation to side information Y for our simulation.

For comparison, we generate two ideal $i.i.d$ Gaussian sequences X and Y correlated by $Y = X + N$, where X has zero mean and unit variance and N is the zero mean Guassian noise with variance σ_N. Y, the corrupted version of X is the side information used for joint decoding.

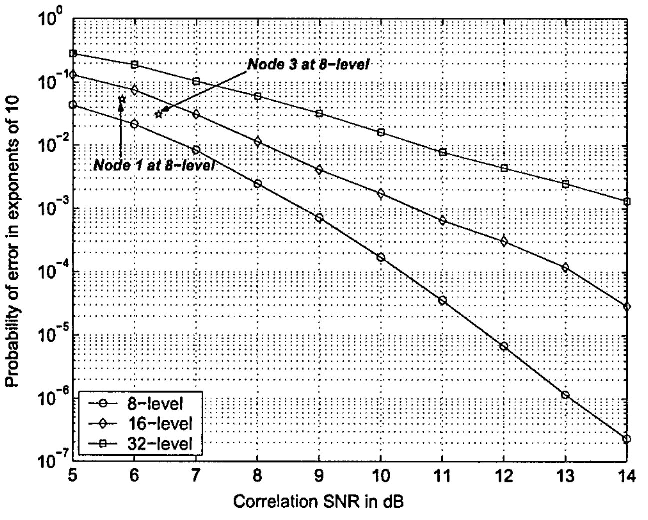


Fig. 9. Probability of Error for R=2bits/sample, Lloyd-Max quantization and coset encoder.

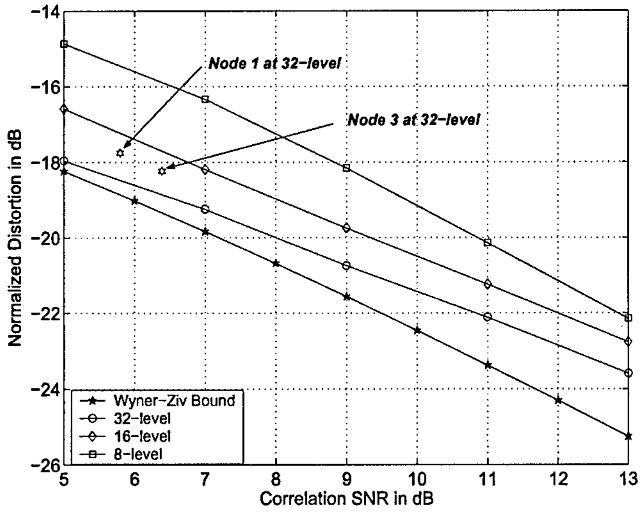


Fig. 10. Normalized Distortion for R=2bits/sample, Lloyd-Max quantization and coset encoder.

We employ 8-level, 16-level and 32-level Lloyd-Max quantization. Each is partitioned into two cosets, where each coset set contains 2, 4 and 8 codewords respectively. The number of samples used for the Monte Carlo simulations is 10^{7}. Fig. 9 shows the probability of decoding error for the above three schemes and normalized distortion with correct decoding only is plotted versus correlation SNR for the same schemes in Fig. 10. Observe that for a given correlation SNR, as the number of quantization levels increases, the normalized distortion decreases and the probability of decoding error increases. Ideally for a given transmission rate, we want to quantize with a large number of levels to cut down distortion, but the tradeoff between the distortion and probability of decoding errors put a constraint in this. As can be noted from Fig. 9 and 10, at 8-level quantization, performance of sensor readings from node 1 and node 3 are approximate 0.5 dB from the one of ideal $i.i.d$ Gaussian sources.

In coset encoding, we compare different coset construction methods. Fig. 11 gives the result of coset set 1 and coset set 2 at 8-level quantization. The performance of coset set 2 is slightly better than the one of coset set 1.

Last we employ our coding scheme to all 7 sensor nodes and compute the actual transmitted data bits. We process the information observed in the common epoch and discard the incomplete observations. Under the scheme of two coset sets, the real transmitted data bits remain the same for 8-,16- and 32-level cases. Results are presented in Table IV.

TABLE IV
Compression Ratio of real transmitted data

Levels	Original bits	Transmitted bits	Compression Ratio
8-level	78339	31815	2.46
16-level	104532	31815	3.29
32-level	130665	31815	4.11

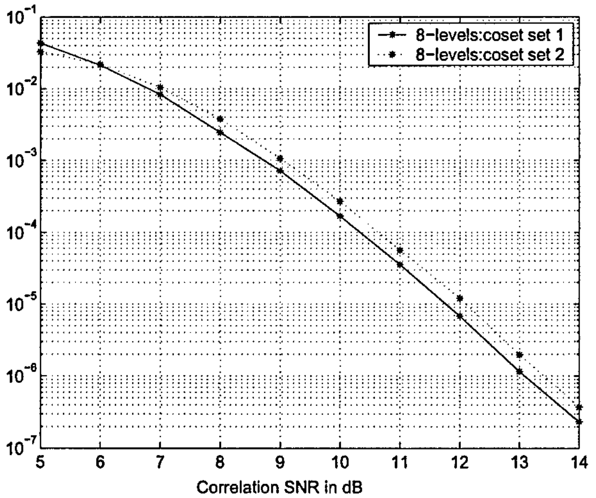


Fig. 11. Probability of Error for R=2bits/sample, Lloyd-Max quantization and coset encoder.

VI. Conclusions

In this paper, we have proposed a spectrum efficient coding scheme for correlated non-binary sources in sensor networks. Instead of using theoreticaly ideal data, our scheme is based on the statistic characters of the correlated non-binary sources from real sensor network. The coset construction introduced in this paper leverages the inherent correlations between sensor observations, but more importantly by minimizing the cross ratio, decreases the probability of decoding error. The proposed scheme performs at 0.5 - 1.5 dB from the Wyner-Ziv distortion bound. We believe our approach provides a practical solution to distributedly compress the acoustic sensor observations and can be extended to the CEO problem. Our future work will concentrate on spectrum efficient coding for distributed sources with memory which is rarely studies so far.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

References

[1] S. S. Pradhan and K. Ramchandran "Distributed Source Coding Using Syndromes(DISCUS):Design and Construction" *IEEE Trans on Information Theory,* vol.49, no.3,Mar.2003.

[2] S. S. Pradhan and K. Ramchandran "Distributed Source Coding:Symmetric rates and Applications to Sensor Networks" *Proc. DCC'00,* Snowbird, UT, Mar.2000

[3] A. D. Liveris, Z. Xiong and C. N. Georghiades "Distributed Compression of Binary Sources Using Conventional Parallel and Serial Concatenated Convolutional Codes" *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies.,* vol. 2, pp. 1293-1303, 2003.

[4] Z. Xiong, A. D. Liveris and S. Cheng "Distributed Source Coding for Sensor Networks" *IEEE International Conference on Wireless Communications and Networking,* vol. 3, pp. 1597 - 1602, 2003.

[5] A. Aaron and B. Girod "Compression with Side Information Using Turbo Codes" *IEEE International Conference on Robotics and Automation,* vol. 1, pp. 165 - 171,2004

[6] D. Slepian and J. K. Wolf "Noiseless Coding of Correlated Information Sources" *Proceedings of the IEEE,* vol. 83, no. 3, pp. 345 - 377,1995

[7] A. D. Wyner and J. Ziv "The rate-distortion function for source coding with side information at the decoder" *IEEE Trans on Information Theory,* vol. 83, no. 3, pp. 345 - 377,1995

[8] T. Cover and J. Thomas "Elements of Information Theory", *New York,* Wiley, 1997

[9] T. Berger, Z. Zhang and H. Viswanathan "The CEO problem [multiterminal source coding]" *IEEE Trans. Inform. Theory,* vol.42,pp.887-902,May 1996

[10] S. Pradhan and K. Ramchandran "Generalized coset codes for symmetric distributed source coding" *submitted to IEEE Trans. Inform. Theory,* Feb,2003

[11] Y. Yang, V. Stankovic, Z. Xiong and W. Zhao "Asymmetric code design for remote multiterminal source coding" *Proc. DCC'04,* Snowbird, UT, Mar.2004

Fundamental Performance Analysis of Event Detection in Wireless Sensor Networks

Haining Shu and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016 USA
E-mail: shu@wcn.uta.edu, liang@uta.edu

Abstract— **A wireless sensor network (WSN) is designed to perform various information processing tasks such as event detection, target tracking and data classification. Comparing with traditional centralized networks, networked sensing offers unique advantage in improved robustness and scalability. Measures of performance for these tasks are well defined, including detection of false alarms or misses, classification errors, and track quality. In this paper, we present a fundamental performance analysis of event detection in wireless sensor networks. Our performance analysis is based on a new detection scheme - double sliding window (DSW) even detection. We compare it theoretically against the fixed threshold approach.**

I. Introduction

Research on sensor networks was originally motivated by military applications. Starting around 1980, networked microsensors technology has been widely used in military applications. One example of such applications is the Cooperative Engagement Capability (CEC) developed by the U.S.Navy. This network-centric warfare consists of multiple radars collecting data on air targets [1]. Other military sensor networks include acoustic sensor arrays for antisubmarine warfare such as the Fixed Distributed System (FDS) and the Advanced Deployable System (ADS), and unattended ground sensors (UGS) such as the Remote Battlefield Sensor System (REMBASS) and the Tactical Remote Sensor System (TRSS).

Nowadays small and inexpensive sensors based upon microelectromechanical system (MEMS) [2] technology, wireless networking, and inexpensive low-power processors allow the deployment of wireless sensor networks for various non-military applications, from environment and habitat monitoring, to industrial process control, to infrastructure security [3] and automation in the transportation.

A wireless sensor network (WSN) consists of certain amount of small and energy constrained nodes. Basic components of sensor node include a single or multiple sensor modules, a wireless transmitter-receiver module, a computational module and a power supply module. Such networks are normally deployed for data collection where human intervention after deployment, to recharge or replace node batteries may not be feasible. Therefore, energy constraint becomes a unique character of WSN comparing to traditional wireless ad-hoc networks. According to [4], energy comsuption occurs in three domains: sensing, data processing (including AD/DA and digital signal processing), and communications. [5] discovered that the sensor, signal processing parts operate at low frequency and consume less than $1mW$. This is over an order of magnitude less than the energy consumption of the communication part. Therefore, we prefer less communication/data exchange between sensor nodes but more local processing implemented by one single sensor node so as to increase the lifetime of the WSN.

The main goal of wireless sensor networks is to monitor physical world. In most of the time, no event happens in the sensed field or surveillance zone. So the sensed data are not necessarily to be stored for a long time or be transmitted to the gateway. Usually, people are more interested in unexpected events. For example, in a scenario of battlefield, people are more interested in the appearance of enemies. If a wireless sensor network is to monitor forest-fire, unusual increasing of the temperature should be a necessary warning to people. Both the appearance of enemies and the unusual increasing of the temperature can be seen as events. Because of the energy, storage, and memory constraints of wireless sensor networks, the ideal state of wireless sensor networks should be event-driven, so that the RF communication circuits can power off at most of the time. Only when certain sensor nodes detect an event, they trigger the RF channel, and transmit the useful information to gateway or headquarters. Therefore, event-detection is one of the key issues for wireless sensor networks, and it's a very efficient way of self-managing, which helps to release the memory and storage constraint and energy constraint.

Performance of wireless sensor network applications is measured in several ways including detection of false alarms or misses, classification errors, and track quality. In this paper, we present a fundamental performance analysis of event detection in wireless sensor networks. We introduce a new scheme of event detection for WSN - double sliding window (DSW) event detection and analyze the fundamental performance: the probability of detection and the probability of false alarm over this new detection scheme.

The rest of this paper is organized as follows. Section II introduce a common type of sensors for tracking: acoustic amplitude sensor model. Double sliding window event detection is described in Section III. In Section IV we detail the fundamental performance analysis over the proposed detection scheme. Section V concludes this paper.

II. Acoustic Amplitude Sensor Model

Localizing and tracking moving objects is an essential capability for a sensor network in many practical applications. While another class of sensor network applications concerns with the problem of sensing/detecting a field. Although they may seem quite different from each other, both require collaborative processing among sensor nodes along the temporal dimension as well as in the spatial domain [6]. In the field sensing case, the collaboration among sensors primarily occurs in the spatial domain and occasionally along the temporal dimension when the field evolves over time. In our study, we focus on on the field sensing/detecting problem.

A. Notation and Assumptions

We use the following notation in our formulation of the sensing/detecting problem in a sensor network:

- Superscript t denotes time. We consider discrete times t that are nonnegative integers.
- Subscript $i \in [1, ..., K]$ denotes the sensor index; K is the total number of sensors in the network.
- Subscript $j \in [1, ..., N]$ denotes the target index; N is the total number of targets being observed.
- The target state at time t is denoted as $\mathbf{x}^t$. For a multi-target sensing/detecting problem, this is a concatenation of individual target states $\mathbf{x}_j^t$.
- The measurement of sensor i at time t is denoted as $\mathbf{z}_i^t$.
- The measurement history up to time t is denoted as $\overline{\mathbf{z}^{(t)}} = \{\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, ..., \mathbf{z}^{(t)}\}$. The measurements may originate from a single sensor or a set of sensors.
- The collection of all sensor measurements at time t are denoted as $\underline{\mathbf{z}^{(t)}} = \left\{\mathbf{z}_1^{(t)}, \mathbf{z}_2^{(t)}, ..., \mathbf{z}_K^{(t)}\right\}$.

In this paper, we consider a single sound source as the target ($N = 1$) and the target state $\mathbf{x}^t$ is the location of the target in a two-dimensinal plane. Each sensor measures the received signal strength reflected from the target. We make the assumption that the sensor characteristics are time-invariant and the target locates in a fixed position.

B. Sensing Model

The time-dependent measurement $\mathbf{z}_i^{(t)}$ of sensor i with characteristics $\lambda_i^{(t)}$ is related to the target state $\mathbf{x}^{(t)}$ through the following observation model,

$$\mathbf{z}_i^{(t)} = \mathbf{h}\left(\mathbf{x}^{(t)}, \lambda_i^{(t)}\right) \qquad (1)$$

where $\mathbf{h}$ is a function depending on $\mathbf{x}^{(t)}$ and parameterized by $\lambda_i^{(t)}$, which represents our knowledge about sensor i. In our study, we consider the sensing model for a single target with $\mathbf{x}$ representing the location of the target. Typical characteristics $\lambda_i^{(t)}$ about sensor i include sensing modality (e.g. what kind of sensor i is), sensor position ζ_i and other parameters, such as the noise model of sensor i. Normally, the sensor characteristics are relatively stable comparing with the more dynamic measurements.

Eq (1) is a general form of the observation model that accounts for possibly nonlinear relations between the sensor type, sensor position, noise model etc. A special case of (1) would be

$$\mathbf{h}\left(\mathbf{x}^{(t)}, \lambda_i^{(t)}\right) = \mathbf{f}_i\left(\mathbf{x}^{(t)}, \lambda_i^{(t)}\right) + \mathbf{w}_i^t \qquad (2)$$

where $\mathbf{f}_i$ is a observation function, and $\mathbf{w}_i$ is additive, zero mean noise with known covariance.

In order to illustrate the idea, we consider the problem of stationary target localization with time-invariant sensor characteristics. In this paper, we assume that all sensors are acoustic sensors measuring only the amplitude of the received sound signal so that the state parameter $\mathbf{x}$ is the unknown target position. Note that under our assumption, there is no longer a time dependence for $\mathbf{x}$ and λ_i. Assuming that acoustic signals propagate isotropically, the parameters are related to the measurements by

$$z_i = \frac{a_i}{\|\mathbf{x} - \zeta_i\|^{\frac{\alpha}{2}}} + w_i \qquad (3)$$

where a_i is a given random variable representing the amplitude of the signal at the target, α is a known attenuation coefficient, and $\|\cdot\|$ is the Euclidean norm. The term w_i is a zero mean Gaussian random variable with variance σ_i^2.

C. Acoustic Amplitude Sensor

There are two common types of sensors for detecting and tracking: acoustic amplitude sensors and direction-of-arrival (DOA) sensors. In this section, we detail the characteristics of the acoustic amplitude sensors.

An acoustic amplitude sensor node measures sound amplitude at the microphone and estimates the distance to the target based on the physics of sound attenuation. Generally, range sensors estimate distance based on received signal strength or time difference of arrival (TDOA).

Assuming that the sound source is a point source and sound propagation is lossless and isotropic, a root-mean-squared (RMS) amplitude measurement z is related to the sound source position $\mathbf{x}$ as

$$z = \frac{a}{\|\mathbf{x} - \zeta\|} + w \qquad (4)$$

where a is the RMS amplitude of the sound source, ζ is the location of the sensor, and w is RMS measurement noise [7]. This is a special case of (3). For simplicity, we model w as a Gaussian with zero mean and variance σ^2.

III. Double Slide Window Event Detection

The ability of a sensor receiver to detect a weak echo signal is limited by the noise that occupies the same part of the frequency spectrum as the signal. Detection of an acoustic signal is based on establishing a threshold at the output of the receiver. If the receiver output exceeds the threshold, a target is said to be present. This is called *threshold detection.* Fig. 1 represents the output of an acoustic receiver as a function of time. The fluctuating appearance of the output is due to the random nature of receiver noise.

A threshold level in Fig. 1 is shown by the long dashed line. If the signal is large enough, as at A and B, a target is reported to be present, but C is a missed detection at the given threshold. The signal at C would have been detected if the threshold were lower. But too low threshold increases the likelihood that noise alone will exceed the threshold and causes false alarm.

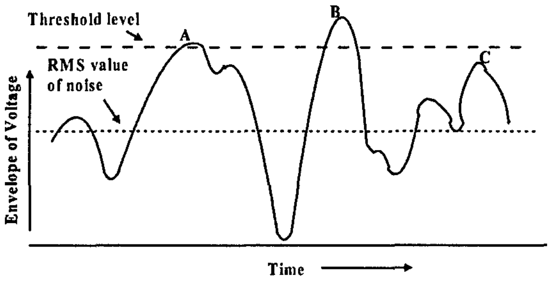


Fig. 1. Envelope of the radar receiver output as a function of time. A, B and C represent signal plus noise. A and B would be valid detections, but C is a missed detection

In [8], the received signal strength $\overline{S}$ from acoustic sensors in a fixed period of time is integrated, when it exceeds a threshold, the authors claim a detection of event occurred as:

$$\overline{S} = \sum_{l=0}^{M-1} |z_{n-l}|^2 \tag{5}$$

$$\overline{S} \geq \overline{S}_{threshold} \tag{6}$$

where z denotes the measurement of received signal strength at each sampling point. M is the length of observing/sampling window.

However, this simple method suffers from a significant drawback; namely, the value of the threshold depends on the sensed signal energy. When there is no event occuring in the sensing range, the sensed signal consists of only noise. The level of the noise power is generally unknown and can change when the environment changes or if unwanted interferers go on and off. Therefore, it is quite difficult to set a fixed threshold. We design a double sliding window algorithm for event-detection so as to alleviate the threshold value selection problem.

The double sliding window event-detection algorithm calculates two consecutive sliding windows of the sensed signal energy. The basic principle is to form the decision variable as the ratio of the total energy contained inside the two windows. Fig. 2 shows two consecutive windows A and B (note that window B arrives after window A) and the response of the ratio R_s to a sensed event. It can be seen that when only noise is sensed the response is nearly flat, since both windows contain ideally the same amount of noise energy.

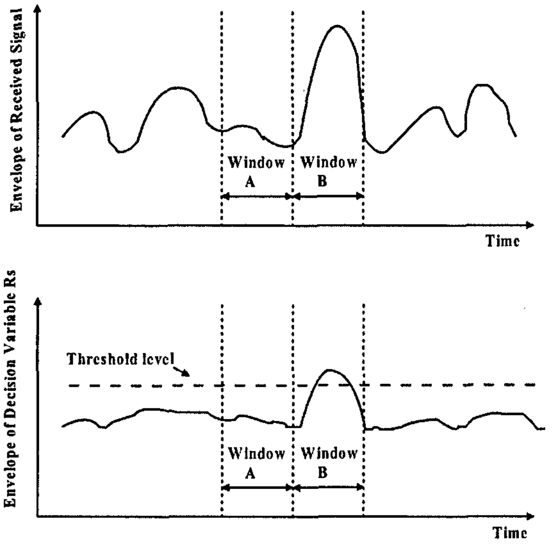


Fig. 2. Illustration of double sliding window event detection, A and B are two continuous sampling windows with same length and B arrives after A.

The calculation of the window A and window B value is shown as

$$\overline{S}_a = \sum_{l=0}^{M-1} |z_{n-l}|^2 , \tag{7}$$

$$\overline{S}_b = \sum_{l=0}^{M-1} |z_{n+M-l}|^2 . \tag{8}$$

Then the decision variable R_s is

$$R_s = \frac{\overline{S}_b}{\overline{S}_a} . \tag{9}$$

The advantage of this approach is the decision variable R_s does not depend on the sensed signal energy, but on the ratio of the energy of two consecutive windows.

IV. Fundamental Performance Analysis

A. Fixed Threshold Event Detection

In the acoustic sensor, the sensed signal needs to pass an IF filter after the A/D converter. If there is no event (only noise exists) in the observed sampling window, the noise to the sensor at the input to the IF filter can be described by Gaussian probability density function (pdf) with mean value of zero and variance ψ_0. Rice [9] has shown that when Gaussian noise is passed through the IF filter, the pdf of the noise envelop R follows Rayleigh distribution:

$$p(R) = \frac{R}{\psi_0} \exp\left(-\frac{R^2}{2\psi_0}\right) \tag{10}$$

The probability that the envelop of the noise will exceed the fixed threshold V_D is,

$$P_{fa} = \int_{V_D}^{\infty} \frac{R}{\psi_0} \exp\left(-\frac{R^2}{2\psi_0}\right) dR = \exp\left(-\frac{V_D^2}{2\psi_0}\right) \quad (11)$$

which is the probability of false alarm rate.

If there is event occurring, the pdf of the sensed signal to the sensor at the input of IF filter is Gaussian pdf with mean value of m and variance ψ_0. The pdf of the envelop R of the sensed signal passing the IF filter has a Rice distribution [9]:

$$P_s(R) = \frac{R}{\psi_0} \exp\left(-\frac{R^2+m^2}{2\psi_0}\right) I_0\left(\frac{Rm}{\psi_0}\right) \quad (12)$$

where $I_0(z)$ is the zero-order modified Bessel function. The probability of detection P_d is the probability that the envelope R will exceed the threshold V_D:

$$P_d = \int_{V_D}^{\infty} p_s(R) dR \quad (13)$$

which is the probability of detection.

We are interested to know the optimal value of threshold V_D. To get V_D, we use maximum a posterior (MAP) detection. The decision boundary is

$$\frac{f(R|H_2)}{f(R|H_1)} = \frac{p(H_1)}{p(H_2)} \quad (14)$$

where $p(H_1)$ is the probability of no events and $p(H_2)$ is the probablity of events happening in one observation. Assume we have the knowledge of $p(H_1)$ and $p(H_2)$, optimal threshold V_D can be derived from (14).

Let $\beta \triangleq p(H_1)/p(H_2)$, applying (10) and (12) we get,

$$\frac{R}{\psi_0}\exp(-\frac{R^2}{2\psi_0}) = \beta\frac{R}{\psi_0}\exp(-\frac{R^2+m^2}{2\psi_0})I_0(\frac{Rm}{\psi_0}) \quad (15)$$

Optimal threshold V_D is the solution of (15) and V_D can be written as:

$$V_D = \frac{\psi_0}{m} I_0^{-1}\left(\frac{1}{\beta}\exp(\frac{m^2}{2\psi_0})\right) \quad (16)$$

B. Double Sliding Window Event Detection

In double sliding window (DSW) detection, decision is made over two consecutive sampling windows. According to the example in Section III, an event or false alarm is reported when the decision variable R_s exceeds a given threshold S_D (note S_D is different from signal strength threshold V_D in Section IV-A). In the case of two consecutive windwos A and B (note that window B arrives after window A), detecting an event and false alarm occur respectively in the following two conditions:

- Detecting an event - Window A represents background noise and window B represents the occurring events.
- False alarm - Window A and B both represent backgroud noise but the decision variable R_s exceeds the threshold.

We then analyze the foundamental performance - the probability of detection and the probability of false alarm in the DSW detection scheme.

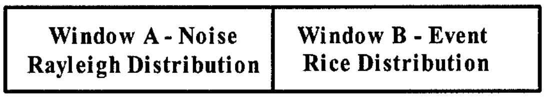


Fig. 3. Case of Detecting an Event (window B arrives after window A)

1) Probability of Detection: From Fig. 3, in the observing window A, a zero mean and variance ψ_0 Gaussian noise passes through the IF filter, the pdf of the envelope of signal strength follows Rayleigh distribution as in (10). In window B, since there is an event occurring, the pdf of the envelope of the sensed signal strength passing the IF filter has a Rice distribution as in (12).

Let $X = \overline{S}_b$, $Y = \overline{S}_a$ and $Z = R_s$ (referring to Section III). We get the pdf of decision variable $Z = X/Y$ in the following.

Since random variable X and Y are identically independent, we have

$$f_Z(z) = \int_{y=0}^{\infty} y f_{x=yz}(x = yz) f_y(y) dy \quad (17)$$

Using the pdf of observing window A and B, we get,

$$f_Z(z) = \int_{y=0}^{\infty} y \cdot \frac{yz}{\psi_0} \exp\left(-\frac{y^2z^2+m^2}{2\psi_0}\right) I_0\left(\frac{yzm}{\psi_0}\right) \frac{y}{\psi_0} \exp\left(-\frac{y^2}{2\psi_0}\right) dy \quad (18)$$

The pdf of decision variable $Z = R_s$ can be get by simplifying (18),

$$f_Z(z) = \int_{y=0}^{\infty} \frac{y^3 z}{\psi_0^2} \exp\left[-\frac{y^2(1+z^2)+m^2}{2\psi_0}\right] I_0(\frac{yzm}{\psi_0}) dy \quad (19)$$

The probability of detection in DSW detection scheme, i.e., the probability that the envelop of the decision variable $Z = R_s$ will exceed the given threshold S_D is,

$$P_d = \int_{S_D}^{\infty} dz \int_{y=0}^{\infty} \frac{y^3 z}{\psi_0^2} \exp\left[-\frac{y^2(1+z^2)+m^2}{2\psi_0}\right] I_0(\frac{yzm}{\psi_0}) dy \quad (20)$$

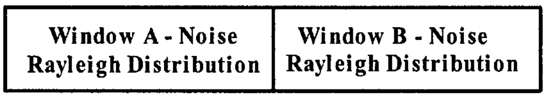


Fig. 4. Case of False Alarm (window B arrives after window A)

2) Probability of False Alarm: When window A and B both represent backgroud noise as shown in Fig. 4 but the decision variable R_s exceeds the threshold S_D, a false alarm is reported. In this case, the pdf of decision variable $Z = R_s$ is derived similarly starting from (17).

$$f_Z(z) = \int_{y=0}^{\infty} y \cdot \frac{yz}{\psi_0} \exp\left(-\frac{y^2 z^2}{2\psi_0}\right) \cdot \frac{y}{\psi_0} \exp\left(-\frac{y^2}{2\psi_0}\right) dy \quad (21)$$

It follows that:

$$\begin{aligned} f_Z(z) &= \frac{z}{\psi_0^2} \int_{y=0}^{\infty} y^3 \exp\left(-\frac{1+z^2}{2\psi_0} y^2\right) dy \\ &= \frac{z}{\psi_0^2} \int_{y=0}^{\infty} \frac{1}{2} y^2 \exp\left(-\frac{1+z^2}{2\psi_0} y^2\right) dy^2 \end{aligned} \quad (22)$$

Replacing y^2 with s, (22) becomes:

$$\begin{aligned} f_Z(z) &= \frac{z}{\psi_0^2} \int_{s=0}^{\infty} s \cdot \exp\left(-\frac{1+z^2}{2\psi_0} s\right) ds \\ &= -\frac{z}{1+z^2} \int_{s=0}^{\infty} s \cdot d\left[\exp\left(-\frac{1+z^2}{2\psi_0} s\right)\right] \end{aligned} \quad (23)$$

Let

$$U(z) = \int_{s=0}^{\infty} s \cdot d\left[\exp\left(-\frac{1+z^2}{2\psi_0} s\right)\right] \quad (24)$$

We next simplify $U(z)$ in (24) using the method of definite integral:

$$\int_a^b f(x) dg(x) = f(x)g(x)|_a^b - \int_a^b g(x) f'(x) dx \quad (25)$$

Comparing (24) and (25), we get $f(s) = s$ and $g(s) = \exp(-(1+z^2)s/(2\psi_0)$. $U(z)$ can be solved as follows:

$$U(z) = s \cdot \left[\exp(-\frac{1+z^2}{2\psi_0} s)\right] |_{s=0}^{\infty} - \int_0^{\infty} \exp(-\frac{1+z^2}{2\psi_0} s) ds$$

$$U(z) = -\int_0^{\infty} \exp(-\frac{1+z^2}{2\psi_0} s) ds = -\frac{2\psi_0}{1+z^2} \quad (26)$$

Substituting $U(z)$ in (23) with (26), we get the pdf of decicion variable $Z = R_s$ as:

$$f_Z(z) = \frac{2\psi_0 z}{(1+z^2)^2} \quad (27)$$

The probability of false alarm in DSW detection scheme, i.e., the probability that the envelop of the decision variable $Z = R_s$will exceed the given threshold S_D is,

$$P_{fa} = \int_{S_D}^{\infty} \frac{2\psi_0 z}{(1+z^2)^2} dz = \int_{S_D}^{\infty} \frac{\psi_0}{(1+z^2)^2} dz^2 \quad (28)$$

Let $z^2 = q$, (28) can be written as:

$$P_{fa} = \int_{S_D^2}^{\infty} \frac{\psi_0}{(1+q)^2} dq = \frac{\psi_0}{1+S_D^2} \quad (29)$$

Observe that the probability of false alarm depends only on the noise variance and threshold level which is reasonable since in this case, no signal but noise gets involved.

Similarly, we use MAP detection to get optimal threshold S_D in DSW event detection.

Assume $\beta \triangleq p(H_1)/p(H_2)$ (same as in fixed threshold detection), applying (19) and (27) we get:

$$\frac{2\psi_0 z}{(1+z^2)^2} = \beta \int_{y=0}^{\infty} \frac{y^3 z}{\psi_0^2} \exp\left[-\frac{y^2(1+z^2)+m^2}{2\psi_0}\right] I_0(\frac{yzm}{\psi_0}) dy \quad (30)$$

which is equivalent to,

$$(1+z^2)^2 \int_{y=0}^{\infty} y^3 \exp\left[-\frac{y^2(1+z^2)+m^2}{2\psi_0}\right] I_0(\frac{yzm}{\psi_0}) dy = \frac{2\psi_0^3}{\beta} \quad (31)$$

Solving (31) gives the optimal threshold S_D in DSW event detection.

V. Conclusions

Measures of performance for wireless sensor network applications are .defined in various ways in which, detection probability and false alarm probability, classification errors and track quality have been widely used. In this paper, we studied the performance of event detection in WSN. We introduced a detection scheme - double sliding window (DSW) event detection and analyzed the fundamental performace - the probability of detection and the probability of false alarm over this new detection scheme. We believe that our DSW detection will practically approach or exceed the fixed threshold detection. Simulations over Xbow WSN professional developer's kit will be provided in the later version.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

References

[1] C. Y. Chong, S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proc. IEEE* , vol 91, no. 8 , pp. 1247 -1256, Aug 2003.

[2] J. W. Gardner, V. K. Varadan and O. O. Awadelkarim, *Microsensors, MEMS and Smart Devices*, New York, Wiley, 2001

[3] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer and T. W. Kenny, "Two-tired wireless sensor network architecture for structural health monitoring," *Proc. SPIE,* San Diego, CA, Mar 2003.

[4] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie 'Protocols for Self-Organization of a Wireless Sensor Network,", *IEEE Personal Communications,* vol 7, issue 5, pp. 16-27, Oct.2000.

[5] A. Y. Wang and C. G. Sodini, "A Simple Energy Model for Wireless Microsensor Transceivers," *Globecom 2004, Dallas, TX.,*

[6] F. Zhao and L. Guibas, *Wireless Sensor Networks,* San Francisco, CA, Morgan Kaufmann, 2004

[7] L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders, *Fundamentals of Acoustics*, New York, John Wiley & Sons, Inc., 1999

[8] S. Meguerdichain, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks", *Mobicom 2001,* Rome, Italy, July 2001.

[9] S. O. Rice, "Mathematical analysis of random noise", *Bell System Tech J.*, vol. 23, pp. 282-332, 1944 and vol. 24, pp. 46-156, 1945

Event Detection in Sensor Networks: Algorithms and Performance Analysis

Haining Shu and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016 USA
E-mail: shu@wcn.uta.edu, liang@uta.edu

***Abstract*— A wireless sensor network (WSN) is designed to perform various information processing tasks such as event detection, target tracking and data classification. Comparing with traditional centralized networks, networked sensing offers unique advantage in improved robustness and scalability. Measures of performance for these tasks are well defined, including detection of false alarms or misses, classification errors, and track quality. In this paper, we present a new algorithm of event detection in wireless sensor networks. Our performance analysis is based on the new detection scheme - double sliding window (DSW) event detection. We compare it theoretically against the fixed threshold approach in terms of probability of detection and false alarm.**

I. Introduction

Research on sensor networks was originally motivated by military applications. Starting around 1980, networked microsensors technology has been widely used in military applications. One example of such applications is the Co-operative Engagement Capability (CEC) developed by the U.S.Navy. This network-centric warfare consists of multiple radars collecting data on air targets [1]. Other military sensor networks include acoustic sensor arrays for antisubmarine warfare such as the Fixed Distributed System (FDS) and the Advanced Deployable System (ADS), and unattended ground sensors (UGS) such as the Remote Battlefield Sensor System (REMBASS) and the Tactical Remote Sensor System (TRSS).

Nowadays small and inexpensive sensors based upon microelectromechanical system (MEMS) [2] technology, wireless networking, and inexpensive low-power processors allow the deployment of wireless sensor networks for various non-military applications, from environment and habitat monitoring, to industrial process control, to infrastructure security [3] and automation in the transportation.

A wireless sensor network (WSN) consists of certain amount of small and energy constrained nodes. Basic components of sensor node include a single or multiple sensor modules, a wireless transmitter-receiver module, a computational module and a power supply module. Such networks are normally deployed for data collection where human intervention after deployment, to recharge or replace node batteries may not be feasible. Therefore, energy constraint becomes a unique character of WSN comparing to traditional wireless ad-hoc networks. According to [4], energy comsuption occurs in three domains: sensing, data processing (including AD/DA and digital signal processing), and communications. [5] discovered that the sensor, signal processing parts operate at low frequency and consume less than $1mW$. This is over an order of magnitude less than the energy consumption of the communication part. Therefore, we prefer less communication/data exchange between sensor nodes but more local processing implemented by one single sensor node so as to increase the lifetime of the WSN.

The main goal of wireless sensor networks is to monitor physical world. In most of the time, no event happens in the sensed field or surveillance zone. So the sensed data are not necessarily to be stored for a long time or be transmitted to the gateway. Usually, people are more interested in unexpected events. For example, in a scenario of battlefield, people are more interested in the appearance of enemies. If a wireless sensor network is to monitor forest-fire, unusual increasing of the temperature should be a necessary warning to people. Both the appearance of enemies and the unusual increasing of the temperature can be seen as events. Because of the energy, storage, and memory constraints of wireless sensor networks, the ideal state of wireless sensor networks should be event-driven, so that the RF communication circuits can power off at most of the time. Only when certain sensor nodes detect an event, they trigger the RF channel, and transmit the useful information to gateway or headquarters. Therefore, event-detection is one of the key issues for wireless sensor networks, and it's a very efficient way of self-managing, which helps to release the memory and storage constraint and energy constraint.

Performance of wireless sensor network applications is measured in several ways including detection of false alarms or misses, classification errors, and track quality. In this paper, we present a fundamental performance analysis of event detection in wireless sensor networks. We introduce a new scheme of event detection for WSN - double sliding window (DSW) event detection and analyze the fundamental performance: the probability of detection and the probability of false alarm over this new detection scheme.

The rest of this paper is organized as follows. Section II introduce a common type of sensors for tracking: acoustic amplitude sensor model. Double sliding window event detection is described in Section III. In Section IV we detail the fundamental performance analysis over the proposed detection scheme. Section V concludes this paper.

II. Acoustic Amplitude Sensor Model

Localizing and tracking moving objects is an essential capability for a sensor network in many practical applications. While another class of sensor network applications concerns with the problem of sensing/detecting a field. Although they may seem quite different from each other, both require collaborative processing among sensor nodes along the temporal dimension as well as in the spatial domain [6]. In the field sensing case, the collaboration among sensors primarily occurs in the spatial domain and occasionally along the temporal dimension when the field evolves over time. In our study, we focus on on the field sensing/detecting problem.

A. Notation and Assumptions

We use the following notation in our formulation of the sensing/detecting problem in a sensor network:

- Superscript t denotes time. We consider discrete times t that are nonnegative integers.
- Subscript $i \in [1, ..., K]$ denotes the sensor index; K is the total number of sensors in the network.
- Subscript $j \in [1, ..., N]$ denotes the target index; N is the total number of targets being observed.
- The target state at time t is denoted as $\mathbf{x}^t$. For a multi-target sensing/detecting problem, this is a concatenation of individual target states $\mathbf{x}_j^t$.
- The measurement of sensor i at time t is denoted as $\mathbf{z}_i^t$.
- The measurement history up to time t is denoted as $\overline{\mathbf{z}^{(t)}} = \left\{\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, ..., \mathbf{z}^{(t)}\right\}$. The measurements may originate from a single sensor or a set of sensors.
- The collection of all sensor measurements at time t are denoted as $\underline{\mathbf{z}^{(t)}} = \left\{\mathbf{z}_1^{(t)}, \mathbf{z}_2^{(t)}, ..., \mathbf{z}_K^{(t)}\right\}$.

In this paper, we consider a single sound source as the target ($N = 1$) and the target state $\mathbf{x}^t$ is the location of the target in a two-dimensinal plane. Each sensor measures the received signal strength reflected from the target. We make the assumption that the sensor characteristics are time-invariant and the target locates in a fixed position.

B. Sensing Model

The time-dependent measurement $\mathbf{z}_i^{(t)}$ of sensor i with characteristics $\lambda_i^{(t)}$ is related to the target state $\mathbf{x}^{(t)}$ through the following observation model,

$$\mathbf{z}_i^{(t)} = \mathbf{h}\left(\mathbf{x}^{(t)}, \lambda_i^{(t)}\right) \tag{1}$$

where $\mathbf{h}$ is a function depending on $\mathbf{x}^{(t)}$ and parameterized by $\lambda_i^{(t)}$, which represents our knowledge about sensor i. In our study, we consider the sensing model for a single target with $\mathbf{x}$ representing the location of the target. Typical characteristics $\lambda_i^{(t)}$ about sensor i include sensing modality (e.g. what kind of sensor i is), sensor position ζ_i and other parameters, such as the noise model of sensor i. Normally, the sensor characteristics are relatively stable comparing with the more dynamic measurements.

Eq (1) is a general form of the observation model that accounts for possibly nonlinear relations between the sensor type, sensor position, noise model etc. A special case of (1) would be

$$\mathbf{h}\left(\mathbf{x}^{(t)}, \lambda_i^{(t)}\right) = \mathbf{f}_i\left(\mathbf{x}^{(t)}, \lambda_i^{(t)}\right) + \mathbf{w}_i^t \tag{2}$$

where $\mathbf{f}_i$ is a observation function, and $\mathbf{w}_i$ is additive, zero mean noise with known covariance.

In order to illustrate the idea, we consider the problem of stationary target localization with time-invariant sensor characteristics. In this paper, we assume that all sensors are acoustic sensors measuring only the amplitude of the received sound signal so that the state parameter $\mathbf{x}$ is the unknown target position. Note that under our assumption, there is no longer a time dependence for $\mathbf{x}$ and λ_i. Assuming that acoustic signals propagate isotropically, the parameters are related to the measurements by

$$z_i = \frac{a_i}{\|\mathbf{x} - \zeta_i\|^{\frac{\alpha}{2}}} + w_i \tag{3}$$

where a_i is a given random variable representing the amplitude of the signal at the target, α is a known attenuation coefficient, and $\|\cdot\|$ is the Euclidean norm. The term w_i is a zero mean Gaussian random variable with variance σ_i^2.

C. Acoustic Amplitude Sensor

There are two common types of sensors for detecting and tracking: acoustic amplitude sensors and direction-of-arrival (DOA) sensors. In this section, we detail the characteristics of the acoustic amplitude sensors.

An acoustic amplitude sensor node measures sound amplitude at the microphone and estimates the distance to the target based on the physics of sound attenuation. Generally, range sensors estimate distance based on received signal strength or time difference of arrival (TDOA).

Assuming that the sound source is a point source and sound propagation is lossless and isotropic, a root-mean-squared (RMS) amplitude measurement z is related to the sound source position $\mathbf{x}$ as

$$z = \frac{a}{\|\mathbf{x} - \zeta\|} + w \tag{4}$$

where a is the RMS amplitude of the sound source, ζ is the location of the sensor, and w is RMS measurement noise [7]. This is a special case of (3). w is Gaussian with zero mean and variance σ^2.

III. Double Slide Window Event Detection

The ability of a sensor receiver to detect a weak echo signal is limited by the noise that occupies the same part of the frequency spectrum as the signal. Detection of an acoustic signal is based on establishing a threshold at the output of the receiver. If the receiver output exceeds the threshold, a target is said to be present. This is called *threshold detection.* Fig. 1 represents the output of an acoustic receiver as a function of time. The fluctuating appearance of the output is due to the random nature of receiver noise.

A threshold level in Fig. 1 is shown by the long dashed line. If the signal is large enough, as at A and B, a target is reported to be present, but C is a missed detection at the given threshold. The signal at C would have been detected if the threshold were lower. But too low threshold increases the likelihood that noise alone will exceed the threshold and causes false alarm.

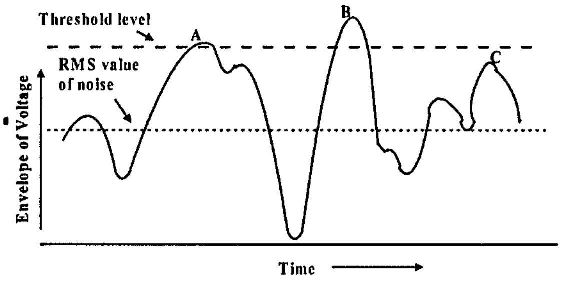


Fig. 1. Envelope of the radar receiver output as a function of time. A, B and C represent signal plus noise. A and B would be valid detections, but C is a missed detection

In [8], the received signal strength $\overline{S}$ from acoustic sensors in a fixed period of time is integrated, when it exceeds a threshold, the authors claim a detection of event occurred as:

$$\overline{S} = \sum_{l=0}^{M-1} |z_{n-l}|^2 \tag{5}$$

$$\overline{S} \geq \overline{S}_{threshold} \tag{6}$$

where z denotes the measurement of received signal strength at each sampling point. M is the length of observing/sampling window.

However, this simple method suffers from a significant drawback; namely, the value of the threshold depends on the sensed signal energy. When there is no event occuring in the sensing range, the sensed signal consists of only noise. The level of the noise power is generally unknown and can change when the environment changes or if unwanted interferers go on and off. Therefore, it is quite difficult to set a fixed threshold. We design a double sliding window algorithm for event-detection so as to alleviate the threshold value selection problem.

The double sliding window event-detection algorithm calculates two consecutive sliding windows of the sensed signal energy. The basic principle is to form the decision variable as the ratio of the total energy contained inside the two windows. Fig. 2 shows two consecutive windows A and B (note that window B arrives after window A) and the response of the ratio R_s to a sensed event. It can be seen that when only noise is sensed the response is nearly flat, since both windows contain ideally the same amount of noise energy.

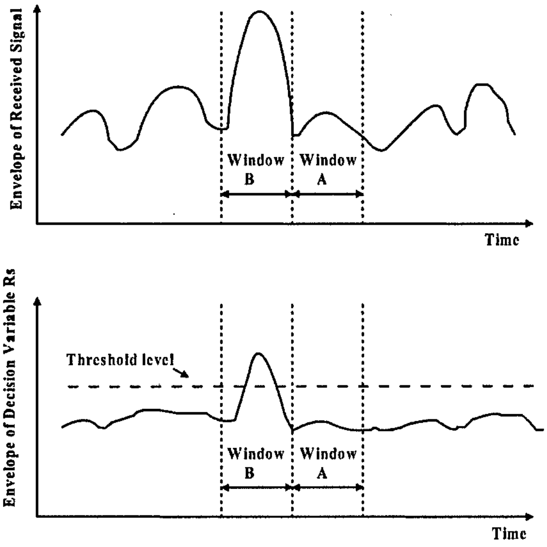


Fig. 2. Illustration of double sliding window event detection, A and B are two continuous sampling windows with same length and B arrives after A.

The calculation of the window A and window B value is shown as

$$\overline{S}_a = \sum_{l=0}^{M-1} |z_{n-l}|^2 , \tag{7}$$

$$\overline{S}_b = \sum_{l=0}^{M-1} |z_{n+M-l}|^2 . \tag{8}$$

Then the decision variable R_s is

$$R_s = \frac{\overline{S}_b}{\overline{S}_a}. \tag{9}$$

The advantage of this approach is the decision variable R_s does not depend on the sensed signal energy, but on the ratio of the energy of two consecutive windows.

IV. Fundamental Performance Analysis

A. Fixed Threshold Event Detection

In a wireless sensor network consisting of acoustic sensors, the received signal at the sensor nodes can be described by Gaussian probability density function (*pdf*).

If there is no event (only noise exists) in the observed sampling window, the received noise at the sensor follows Gaussian distribution with mean value of zero and variance ψ_0.

$$p(R) = \frac{1}{\sqrt{2\pi\psi_0}} \exp(-\frac{R^2}{2\psi_0}) \tag{10}$$

The probability of false alarm which is the probability that the envelop of the noise will exceed the fixed threshold V_D can be determined using Q-function.

$$P_{fa} = \int_{V_D}^{\infty} \frac{1}{\sqrt{2\pi\psi_0}} \exp(-\frac{R^2}{2\psi_0})dR = Q\left(\frac{V_D}{\sqrt{\psi_0}}\right) \quad (11)$$

where Q-function is defined as:

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_{z}^{\infty} \exp(-\frac{x^2}{2})dx = \frac{1}{2}\left[1 - erf(\frac{z}{\sqrt{2}})\right] \quad (12)$$

If there is an event occurring, the *pdf* of the sensed signal to the sensor is Gaussian pdf with mean value of m and variance ψ_0.

$$p(R) = \frac{1}{\sqrt{2\pi\psi_0}} \exp\left[-\frac{(R-m)^2}{2\psi_0}\right] \quad (13)$$

The probability of detection P_d which is the probability that the envelope R will exceed the threshold V_D can also be determined using Q-function.

$$P_d = \int_{V_D}^{\infty} \frac{1}{\sqrt{2\pi\psi_0}} \exp\left[-\frac{(R-m)^2}{2\psi_0}\right] dR = Q\left(\frac{V_D - m}{\sqrt{\psi_0}}\right) \quad (14)$$

We are interested to know the optimal value of threshold $V_D^{'}$. To get $V_D^{'}$, we use maximum a posterior (MAP) detection. The decision boundary is

$$\frac{f(R|H_2)}{f(R|H_1)} = \frac{p(H_1)}{p(H_2)} \quad (15)$$

where H_1 denotes the case of no events while H_2 denotes the case with events. $f(R|H_1)$ and $f(R|H_2)$ therefore represent the *pdfs* of the two cases respectively. In one observation, the probability of no events equals to $p(H_1)$ and the probability of events equals to $p(H_2)$.

Let $\beta \triangleq p(H_1)/p(H_2)$, applying (10) and (13) we get,

$$\frac{1}{\sqrt{2\pi\psi_0}} \exp\left[-\frac{(R-m)^2}{2\psi_0}\right] = \beta\frac{1}{\sqrt{2\pi\psi_0}} \exp(-\frac{R^2}{2\psi_0}) \quad (16)$$

Optimal threshold $V_D^{'}$ is the solution of (16) and $V_D^{'}$ can be written as:

$$V_D^{'} = \frac{\psi_0}{m}\ln\beta + \frac{m}{2} \quad (17)$$

B. Double Sliding Window Event Detection

In double sliding window (DSW) detection, decision is made over two consecutive sampling windows. According to the example in Section III, an event or false alarm is reported when the decision variable R_s exceeds a given threshold S_D (note S_D is different from signal strength threshold V_D in Section IV-A).

In the case of two consecutive windows A and B (note that window B arrives after window A), detecting an event and false alarm occur respectively in the following two conditions:

- Detecting an event - Window A represents background noise and window B represents the occurring events as shown in Fig. 3.
- False alarm - Window A and B both represent backgroud noise but the decision variable R_s exceeds the threshold as shown in Fig. 4.

We then analyze the foundamental performance - the probability of detection and the probability of false alarm in the DSW detection scheme.

Window B - Event
Window A - Noise

Fig. 3. Case of Detecting an Event (window B arrives after window A)

1) Probability of Detection: In Fig. 3, observation of window A is Gaussian noise with zero mean and variance ψ_0 and window B represents event *pdf* which is also Gaussian but with mean value of m and variance ψ_0.

Let $X = \overline{S}_b$, $Y = \overline{S}_a$ and $Z = R_s = \overline{S}_b/\overline{S}_a$ (referring to Section III). We get the *pdf* of decision variable $Z = X/Y$ in the following.

Since random variable X and Y are identically independent, we have

$$f_Z(z) = \int_{y=0}^{\infty} y f_{x=yz}(x = yz) f_y(y)dy \quad (18)$$

Using the *pdf* of observing window A and B, we get,

$$f_Z(z) = \int_{y=0}^{\infty} y \frac{1}{\sqrt{2\pi\psi_0}} \exp\left[-\frac{(yz-m)^2}{2\psi_0}\right] \frac{1}{\sqrt{2\pi\psi_0}} \exp(-\frac{y^2}{2\psi_0}) \quad (19)$$

The *pdf* of decision variable $Z = R_s$ can be get by simplifying (19),

$$f_Z(z) = \int_{y=0}^{\infty} \frac{y}{2\pi\psi_0} \exp\left[-\frac{y^2(1+z^2) - 2mzy + m^2}{2\psi_0}\right] dy \quad (20)$$

The probability of detection in DSW detection scheme, i.e., the probability that the envelop of the decision variable $Z = R_s$ will exceed the given threshold S_D is given as below.

$$P_d = \int_{S_D}^{\infty} dz \int_{y=0}^{\infty} \frac{y}{2\pi\psi_0} \exp\left[-\frac{y^2(1+z^2) - 2mzy + m^2}{2\psi_0}\right] dy \quad (21)$$

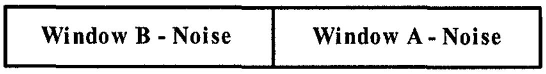


Fig. 4. Case of False Alarm (window B arrives after window A)

2) Probability of False Alarm: When window A and B both represent backgroud noise as shown in Fig. 4 but the decision variable R_s exceeds the threshold S_D, a false alarm is reported. In this case, the *pdf* of decision variable $Z = R_s$ is derived similarly starting from (18).

$$f_Z(z) = \int_{y=0}^{\infty} \frac{y}{\sqrt{2\pi\psi_0}} \exp\left(-\frac{(yz)^2}{2\psi_0}\right) \frac{1}{\sqrt{2\pi\psi_0}} \exp(-\frac{y^2}{2\psi_0})dy \quad (22)$$

It follows that:

$$\begin{aligned} f_Z(z) &= \frac{1}{2\pi\psi_0} \int_{y=0}^{\infty} y \exp\left(-\frac{1+z^2}{2\psi_0} y^2\right) dy \\ &= \frac{1}{4\pi\psi_0} \int_{y=0}^{\infty} \exp\left(-\frac{1+z^2}{2\psi_0} y^2\right) dy^2 \end{aligned} \quad (23)$$

Replacing y^2 with s, (23) becomes:

$$f_Z(z) = \frac{1}{4\pi\psi_0} \int_{s=0}^{\infty} \exp\left(-\frac{1+z^2}{2\psi_0} s\right) ds \quad (24)$$

$f_Z(z)$ can be solved as:

$$f_Z(z) = \frac{1}{2\pi(1+z^2)} \quad (25)$$

The probability of false alarm in DSW detection scheme, i.e., the probability that the envelop of the decision variable $Z = R_s$will exceed the given threshold S_D is,

$$P_{fa} = \int_{S_D}^{\infty} \frac{1}{2\pi(1+z^2)} dz \quad (26)$$

Using indefinite integration formula:

$$\int \frac{1}{a^2+x^2} dx = \frac{1}{a} \arctan\left(\frac{x}{a}\right) \quad (27)$$

The probability of false alarm can be solved as:

$$P_{fa} = \frac{1}{4} - \frac{1}{2\pi} \arctan(S_D) \quad (28)$$

Observe that in DSW event detection scheme, the probability of false alarm does not depend on the noise variance but only on the decision variable threshold S_D.

Similarly, we use MAP detection to get optimal threshold S_D in DSW event detection.

Assume $\beta \triangleq p(H_1)/p(H_2)$ (same as in fixed threshold detection), applying (20) and (25) we get:

$$\int_{y=0}^{\infty} \frac{y}{2\pi\psi_0} \exp\left[-\frac{(1+z^2)y^2 - 2mzy + m^2}{2\psi_0}\right] dy = \frac{1}{2\pi(1+z^2)}\beta \quad (29)$$

Solving (29) gives the optimal threshold S_D in DSW event detection.

V. Conclusions

Measures of performance for wireless sensor network applications are defined in various ways in which, detection probability and false alarm probability, classification errors and track quality have been widely used.

In this paper, we studied the performance of event detection in wireless sensor network. We introduced a new detection algorithm - double sliding window (DSW) event detection where detection decision is made over two consecutive sampling windows. We analyzed the fundamental performace - the probability of detection and the probability of false alarm over this new detection scheme and compared it theoretically against the fixed threshold algorithm. We believe that our DSW detection will practically approach or exceed the fixed threshold detection. Simulations over Xbow wireless sensor network professional developer's kit will be provided in the later version.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

References

[1] C. Y. Chong, S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proc. IEEE* , vol 91, no. 8 , pp. 1247 -1256, Aug 2003.

[2] J. W. Gardner, V. K. Varadan and O. O. Awadelkarim, *Microsensors, MEMS and Smart Devices*, New York, Wiley, 2001

[3] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer and T. W. Kenny, "Two-tired wireless sensor network architecture for structural health monitoring," *Proc. SPIE,* San Diego, CA, Mar 2003.

[4] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie 'Protocols for Self-Organization of a Wireless Sensor Network,", *IEEE Personal Communications,* vol 7, issue 5, pp. 16-27, Oct.2000.

[5] A. Y. Wang and C. G. Sodini, "A Simple Energy Model for Wireless Microsensor Transceivers," *Globecom 2004, Dallas, TX.*,

[6] F. Zhao and L. Guibas, *Wireless Sensor Networks,* San Francisco, CA, Morgan Kaufmann, 2004

[7] L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders, *Fundamentals of Acoustics*, New York, John Wiley & Sons, Inc., 1999

[8] S. Meguerdichain, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks", *Mobicom 2001,* Rome, Italy, July 2001.

[9] S. O. Rice, "Mathematical analysis of random noise", *Bell System Tech J.*, vol. 23, pp. 282-332, 1944 and vol. 24, pp. 46-156, 1945

A Hybrid Approach for Asynchronous Energy-Efficient MAC Protocol for Wireless Sensor Networks

Qingchun Ren and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016
Email: ren@wcn.uta.edu, liang@uta.edu

Abstract—**In this paper, a novel asynchronous energy-efficient MAC protocol, ASCEMAC, is proposed for wireless sensor networks. We combine both contention-based and schedule-based MAC protocols' energy saving strategies in our algorithm. In ASCEMAC, by applying free-running method and fuzzy logic rescheduling scheme, time synchronization which is necessary in existing energy-efficient MAC protocols is not required any more. Moreover, we present a traffic intensity and network density-based model to determine essential algorithm parameters, such as power on/off duration, interval of schedule broadcast and super-time-slot size and order. Simulation results show that our algorithm ensures the average successful transmission rate, decreases the data packet average waiting time, and reduces the average energy consumption. Therefore, network performance is improved and network lifetime is extended by using our algorithm.**

I. Introduction

For wireless sensor networks (WSNs), energy saving is becoming more and more important, due to nodes' limited energy resource. Some solutions for saving energy at MAC layer for WSNs are put forward. They can be classified into two main categories, according to their channel access strategies: contention-based MAC protocols and schedule-based MAC protocols.

In schedule-based energy-efficient MAC protocols: a new standard named IEEE 802.15.4 [5] has been developed. It concentrates on providing a physical-layer and MAC-layer standard with ultra-low complexity, cost, and power for low-data-rate wireless connectivity among cheap fixed devices; Traffic-Adaptive Medium Access (TRAMA) [2] employs a traffic adaptive and distributed election scheme to allocate the system time for different sensor nodes; in EMACS [12], only active nodes monitor new communication requests from passive nodes. Notice that, through appointing transmission time for different sensor nodes, these schedule-based MAC protocols reduce the energy consumption on collision and idle. But, how to allocate time-slots efficiently and fairly is one of the biggest challenges for them.

In contention-based energy-efficient MAC protocols: S-MAC[3], divides the system time into frames. During the sleeping part, a node powers off its radio to save energy, and it performances communications during the active part; T-MAC is proposed in [4]. This protocol enables each node to dynamically and locally adjust the communication duration based on each node's traffic. We can see that, these contention-based MAC protocols implement energy saving through adjusting all nodes' communications into a certain period of time.

In all previously mentioned energy-efficient MAC protocols, even though the mechanisms of managing power on/off period are distinguished, accurate time synchronization method [7] is the common premise to ensure saving energy and communicating successfully among nodes.

As we know, the quality of each sensor node's clock usually boils down to its frequency stability and frequency accuracy [7]. In general, as frequency stability and accuracy increase, so do their power requirements, size and cost, which are all troublesome for general sensor nodes. Hence, clock drifts are unavoidable in most WSNs, which are introduced by unstable and inaccurate frequency standards. In this case, there must be some unsuccessful communications caused by uncoincidently switching back-and-forth between power on/off states (we call these mismatch operations), without a correct global clock established by time synchronization for previously mentioned energy-efficient MAC protocols. In the algorithm description part, we will discuss further how clock drift results in unsuccessful communications.

Moreover, in all previously mentioned energy-efficient MAC protocols, how to determine the durations of power on/off phases are seldom discussed. But, these two durations are closely related to system performances, such as energy efficiency and throughput. In WSNs, the traffic in general has a heterogeneous nature [6], i.e., the traffic arrival rate for different nodes or even for the same node at different time is fluctuating considerably during the network lifetime.

In this paper, we present an asynchronous energy-efficient MAC protocol: ASCEMAC, which not only outperforms the existing energy-efficient MAC protocols, but also removes the tight dependency on time synchronization. We combine both contention-based and schedule-based MAC protocols' energy saving strategies in our algorithm. In ASCEMAC, by applying free-running method and fuzzy logic [10] rescheduling scheme to set up phase-switching schedules and compensate clock drifts among nodes. Moreove, we present a traffic intensity and network density-based model to determine the essential algorithm parameters, such as power on/off duration, interval of schedule broadcast, super-time-slot length and order.

The remainder of this paper is organized as follows: our ASCEMAC design is described in Section II; simulation results are given in Section III; Section IV concludes this paper.

II. ASCEMAC Protocol Description and Discussion

We use Energy-Efficient Self-Organization (ESO) [9] algorithm to form clusters. Each cluster has only one cluster head. The radius of. a cluster is the communication range of the cluster head. Nodes in one cluster can talk to their neighbors directly. The wireless media (or the common channel) access scheme within a cluster is specified by our ASCEMAC.

ASCEMAC divides system time into four phases: *T*RFR-*P*hase, Schedule-*B*roadcast-*P*hase, *On*-*P*hase and *Off*-*P*hase. An on/off rotation consists of two adjacent *On*-*P*hase and *Off*-*P*hase. It is a fixed-schedule stage between two adjacent *S*chedule-*B*roadcast-*P*hases. The fixed-schedule stage consists of several on/off rotations. Fig.1 presents the system time scheme structure. The function for each phase is:

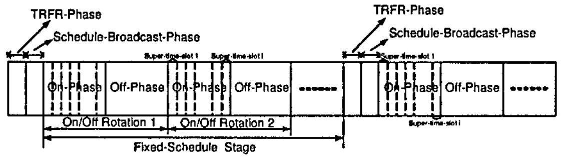


Fig. 1. System Time Scheme Structure

- *T*RFR-*P*hase is preserved for normal nodes to send Traffic-Rate & Failure-Rate (TRFR) messages to their cluster head;
- *S*chedule-*B*roadcast-*P*hase is preserved for cluster head to locally broadcast phase-switching schedules within their control range;
- *Off*-*P*hase is preserved for all nodes to power off their radios. In this phase, there is no communication, but data storing and sensing may happen;
- *On*-*P*hase is preserved for all nodes to power on their radios to make communication. In this phase, the system time is further divided into super-time-slots, which are composed of several normal time-slots. Each super-time-slot is continuously used by one source-destination pair. One normal time-slot is a period of time (T_d) to complete one data transmission from source to destination.

In ASCEMAC, each node informs the cluster head its traffic intensity, failure transmission and buffer overflow through TRFR message (see Section A). Based on that information, the cluster head determines the power on/off duration (see Section B), the interval of schedule broadcast (see Section C), as well as the length and the order of super-time-slot (see Section D). After receiving the schedule broadcast message from the cluster head, each node sets up its own phase-switching schedule. Since then, each node starts to power on its radio to make communication and to power off its radio to save energy according to its phase-switching schedule. We will describe our ACEMAC in detail in the following sections.

A. TRFR Message Design

TRFR message (see Fig.2) is sent by normal node at *T*RFR-*P*hase.

Type	Source	Data Arrival Rate	Failure Rate	Overflowing Rate

Fig. 2. TRFR Message Format

- "Data Arrival Rate" is the number of data packets coming from node's sensing component per second;
- "Failure Rate" is the rate of unsuccessfully transmitted data packets, caused by mismatch operations, to total transmitted data packets;
- "Overflowing Rate" is the rate of overflowing data packets, caused by improper power off duration, to total data packets coming from node's sensing component.

In our algorithm, we add an ACK message as the acknowledgment for successfully receiving. A transmission is defined as unsuccessful when the transmitter does not receive ACK after certain period of time.

During each on/off rotation, each node independently estimates its data traffic arrival rate, unsuccessful transmission rate and overflowing rate. But those rates sent to its cluster head are the average values on all on/off rotations.

During *T*RFR-*P*hase, each node randomly chooses a time to send TRFR message, and this random transmission process complies with an uniform distribution. The working process is almost similar to CSMA [1]. Notice that, transmission time's randomness and carrier sense reduce the collision possibility and increase the successful transmission possibility of TRFR message. In the simulation section (Section III), the experiment on TRFR message will show that normal nodes within a cluster have a very high probability to send TRFR messages to their cluster head successfully.

B. Power on/off Duration (T_n/T_f) Design

Designing T_f, we consider the following factors:

- During $Off\text{-}Phase$ and other nodes' transmission time, a node stops communicating, but there are still data packets arriving from sensing component;
- For each node, the buffer space is limited. When buffer is used up (or overflowed), the following incoming data packets must be discarded;
- There is a lifetime for each data packet. So the traffic over the network is sensitive to waiting time.

Based on the traffic arrival rate, buffer space and traffic lifetime, we design *Off*-*P*hase duration (T_f) to avoid buffer overflow and keep information up to date at most degree. If we know the maximum waiting time W_{max}, the buffer size k_i and the traffic arrival rate λ_i for node i, T_f can be calculated using

$$T_f = \min\left\{(2W_{max} - T_n), \min_i(\frac{k_i}{\lambda_i} - T_n)\right\} \tag{1}$$

In general WSNs, each node has similar capability. Therefore, we can let k_i=K (i=1,2,...). Then (1) is changed to

$$T_f = \min\left\{(2W_{max} - T_n), \min_i(\frac{K}{\lambda_i} - T_n)\right\} \tag{2}$$

It is obvious that the longer the *Off*-*P*hase is, the more the energy is saved. However, the average waiting time of data packets will increase as the duration of *Off*-*P*hase increasing. So there is a trade-off between saving energy and reducing waiting time.

During *On*-*P*hase, nodes start to send/receive data packets. In this phase, system time is divided into slots. Certain number of time slots are continuously occupied by a source-destination pair. There is no competition and carrier sense at *On*-*P*hase. Knowing average traffic arrival rate λ_i for node i, *Off*-*P*hase duration (T_f) and totally N nodes in this cluster, T_n can be calculated as

$$T_n = \frac{T_d T_f \sum_{i=1}^{N} \lambda_i}{1 - T_d \sum_{i=1}^{N} \lambda_i} \tag{3}$$

Combining (2) and (3), we obtain the final equations for T_f and T_n. There are two cases:

1) when $2W_{max} < \min_i(\frac{K}{\lambda_i})$

$$T_f = 2W_{max}(1 - T_d\phi) \tag{4}$$

$$T_n = 2W_{max}T_d\phi \tag{5}$$

2) when $2W_{max} \geq \min_i(\frac{K}{\lambda_i})$

$$T_f = \min_i(\frac{K}{\lambda_i})(1 - T_d\phi) \tag{6}$$

$$T_n = \phi T_d \min_i(\frac{K}{\lambda_i}) \tag{7}$$

where ϕ is the sum of N nodes' traffic arrive rate, defined as $\phi = \sum_{i=1}^{N} \lambda_i$.

Notice that, in our *On*-*P*hase and *Off*-*P*hase durations designing, we try to extend the power off time to save more energy, and also adjust data packets' waiting time to an acceptable value.

C. Phase-Switching Schedule Establishment and Interval of Schedule Broadcast Design

Free-running is a timing method which allows each node to run on its own clock. ACEMAC use free-running method to save energy and spectrum resources because free-running method does not maintain a global clock within a cluster. Furthermore, we design a schedule broadcast message (see Fig. 3). Cluster head generates this message and broadcasts within this cluster. The function for each field of schedule broadcast message is:

Type	SRC	Off-Duration	On-Duration
SRC_1	DEST_1	Defer-Duration_1	Slot-Duration_1
SRC_2	DEST_2	Defer-Duration_2	Slot-Duration_2

SRC_i	DEST_i	Defer-Duration_i	Slot-Duration_i

Fig. 3. Schedule Broadcast Packet Format

- "On-Duration" specifies when all nodes should switch to *Off-Phase*;
- "Off-Duration" field regulates how long all nodes should stay at one On/Off rotation;
- "Slot-Duration_i" field regulates the length of ith super-time-slot;
- "Defer-Duration_i" is designed to inform nodes after how long the ith super-time-slot starts for an *On-Phase*;
- "SRC_i" and "DEST_i" fields regulate the source and destination of ith super-time-slot.

If clock drifts do not exist, coincident phase-switching schedule is supposed to be set up at each node, based on each node's own local clock and this schedule broadcast message. These phase-switching schedules ensure the match operations among nodes.

But, as we mentioned earlier, mismatch operations among nodes are unavoidable because there are always some clock drifts caused by unstable and inaccurate frequency standards.

The following example illustrates how clock drift results in a mismatch operation, and how our ASCEMAC removes this mismatch to ensure successful communication. There is a source-destination pair, nodes A and B. If the frequency standard for A is faster than that of B and super-time-slot_1 is the time-slot of A and B, A will run into super-time-slot_1 preceding B for an unneglectable time (Δt_1) after a period of time. During Δt_1, data transmissions cannot be done successfully between them, because B's radio is still off. See Fig. 4.

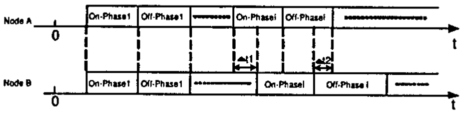


Fig. 4. Mismatch Operation Due to Clock Drift

Schedule broadcast is responsible for removing mismatch, in addition to informing nodes about phase-switching schedules. From Fig. 5, we see that Δt_1 between nodes A and B is successfully removed after receiving a new schedule broadcast message.

From the above discussion, we can see that ensuring nodes against mismatch operations can avoid unsuccessful transmissions, which are caused by clock drifts.

However, it is unnecessary to offer match operations at all time and for all nodes. For instance, two nodes, which have little information to exchange, do not need to switch phases coincidently, since their mismatch operations have little effect on information transmission. In that case, some nodes could be allowed to go out of coincidence, and be rescheduled only if necessary.

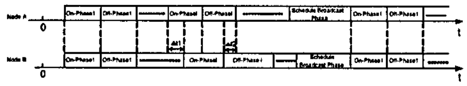


Fig. 5. Mismatch Operation Removed by Re-schedule

There is an another function for schedule broadcast, besides removing mismatch and informing phase-switching schedules. That is, the cluster head can acquire more suitable durations for power on/off phase according to current traffic conditions. For WSNs, the traffic is heterogeneous. With the vibration of traffic arrival rate, previously chosen T_f and T_n may not optimum any more. For example, when the traffic arrival rate increases, more data packets arrive during *Off-Phase* and *On-Phase*, so the possibility for buffer overflowing will increase. In another case, when the traffic arrival rate decreases, less data packets arrive during *Off-Phase* and *On-Phase*, so some energy is wasted by idle at *On-Phase*.

We adopt an adaptive adjustment method to determine the interval of schedule broadcast. This method can save energy through avoiding unnecessary schedule broadcasts and idle, as well as ensure an acceptable data successful transmission rate.

We use

$$T_i = \xi_i \times T_{i-1} \tag{8}$$

as the interval adjusting function, where T_i is the ith interval of schedule broadcast, ξ_i is the ith adjustment factor and is a positive numeral.

We design a rescheduling-FLS to determine the value of ξ_i, which reflects the influence degree of clock drifts and traffic intensity changes on communications.

In our rescheduling-FLS. there are three antecedents:

- the ratio of nodes with overflowed buffer (R_{of});
- the ratio of nodes with high failing transmission rate (R_{hf});
- the ratio of nodes experiencing unsuccessful transmission (R_{sr}).

The consequent is the adjustment factor for the interval of schedule broadcast(ξ_i). The linguistic variables used to represent R_{of}, R_{hf} and R_{sr} are divided into three levels: Low, $Moderate$ and $High$. ξ_i is divided into 5 levels, $Highly\ Decrease$, $Decrease$, $Unchange$, $Increase$ and $Highly\ Increase$. We show these MFs in Fig. 6 and Fig. 7.

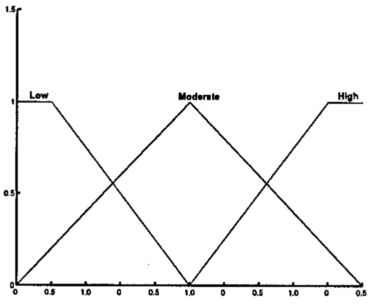


Fig. 6. Antecedent Membership Function

We design our rescheduling-FLS using rules with one example shown below:

R^l:IF $the\ ratio\ of\ nodes\ (x_1)\ with\ overflow\ buffer$ is $High$, $the\ ratio\ of\ nodes\ (x_2)\ with\ high\ failure\ rate$ is $High$ and $the\ ratio\ of\ nodes\ (x_3)\ experiencing\ unsuccessful\ transmission$ is $High$, THEN $the\ adjustment\ factor\ for\ the\ interval\ of\ schedule\ broadcast\ (\xi)$ should be $Highly\ Decrease$.

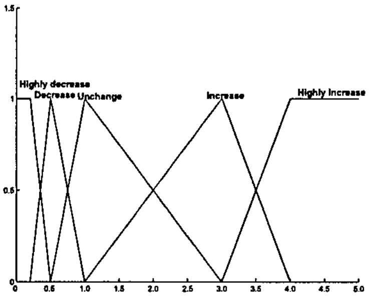


Fig. 7. Consequent Membership Function

TABLE I

THE RULES FOR ADJUSTING THE INTERVAL OF SCHEDULE BROADCAST. ANTE1 IS THE RATIO OF NODES HAVING OVERFLOWED BUFFER. ANTE2 IS THE RATIO OF NODES WITH HIGH FAILURE TRANSMISSION RATE. ANTE3 IS THE RATIO OF NODES OWNING UNSUCCESSFUL TRANSMISSION. AND CONSEQUENT IS THE ADJUSTMENT FACTOR FOR THE INTERVAL OF SCHEDULE BROADCAST.

Rule	*Ante1*	*Ante2*	*Ante3*	*Consequent*
1	*Low*	*Low*	*Low*	*HighlyIncrease*
2	*Low*	*Low*	*Moderate*	*Increase*
3	*Low*	*Moderate*	*Moderate*	*Decrease*
4	*Low*	*Moderate*	*High*	*Decrease*
5	*Moderate*	*Low*	*Moderate*	*Increase*
6	*Moderate*	*Low*	*High*	*Unchange*
7	*Moderate*	*Moderate*	*Moderate*	*Decrease*
8	*Moderate*	*Moderate*	*High*	*HighlyDecrease*
9	*Low*	*High*	*High*	*Decrease*
10	*Moderate*	*High*	*High*	*HighlyDecrease*
11	*High*	*Low*	*Moderate*	*Increase*
12	*High*	*Low*	*High*	*Unchange*
13	*High*	*Moderate*	*Moderate*	*Decrease*
14	*High*	*Moderate*	*High*	*Decrease*
15	*High*	*High*	*High*	*HighlyDecrease*

We summarize all meaningful rules in Table I.

For every input (x_1,x_2,x_3), the output is defuzzified [8] using

$$\xi(x_1, x_2, x_3) = \frac{\sum_{l=1}^{15} \bar{\xi}^l \mu_{\mathrm{F}_1^l}(x_1)\mu_{\mathrm{F}_2^l}(x_2)\mu_{\mathrm{F}_3^l}(x_3)}{\sum_{l=1}^{15} \mu_{\mathrm{F}_1^l}(x_1)\mu_{\mathrm{F}_2^l}(x_2)\mu_{\mathrm{F}_3^l}(x_3)} \tag{9}$$

The height of the five fuzzy sets depicted in Fig. 7 are $\bar{\xi}_1$=0.2, $\bar{\xi}_2$=0.5, $\bar{\xi}_3$=1.0, $\bar{\xi}_4$=3.0, $\bar{\xi}_5$=4.0.

The inputs of rescheduling-FLS are acquired from TRFR messages sent by all normal nodes. Before broadcasting schedules, cluster head estimates the influence degree of clock drifts and traffic intensity changes on communications using rescheduling-FLS. After obtaining ξ_i, the cluster head uses (8) to determine the value for the next interval of schedule broadcast.

D. Time-Slot Assignment

For classic TDMA systems, such as GSM system, the system time is divided into slots, and each user occupies cyclically repeating time slots. A typical TDMA system transmits data in a buffer-and-burst method, thus the transmission for any user is non-continuous and a high quality time synchronization is needed.

But, in ASCEMAC, there is no time synchronization and global clock in the system. In this case, the successful transmission possibility is supposed to be degraded if we still utilize that buffer-and-burst method to schedule communications. The following example illustrates the relationship between the length of super-time-slot and the mismatch operation. See Fig. 4, there is a Δt_1 time difference between nodes A, the source, and B, the destination. A starts sending at the beginning of its super-time-slot. k is the number of data packets sent during this transmission period. $T_{s,min}$ is the least time needed to detect the synchronization information of a data packet. We consider two cases:

1) If $T_{s,min} < \Delta t_1 < T_d$
   a) When k=1, no packet, sent during this slot, can be received by node B, i.e., 0% successful transmission rate;
   b) When k=3, two packets, sent during this slot, can be received by node B, i.e., 67% successful transmission rate;
   c) When k=n, n-1 packets, sent during this slot, can be received by node B, i.e., $\frac{n-1}{n}$% successful transmission rate.
2) If $T_d < \Delta t_1 < 2T_d$
   a) When k=1, no packet, sent during this slot, can be received by node B, i.e., 0% successful transmission rate;
   b) When k=3, one packet, sent during this slot, can be received by node B, i.e., 33% successful transmission rate;
   c) When k=n, n-2 packets, sent during this slot, can be received by node B, i.e., $\frac{n-2}{n}$% successful transmission rate.

Notice that with the increasing of k, more transmissions are done successfully under the same mismatch condition. Therefore, continuously occupying the common channel for several time-slots by one source-destination pair is an effective way to tolerate mismatch between source and destination.

In our algorithm, we adopt a non-buffer-and-burst method to transmit data. That is, based on the number of data packets waiting for transmission and unsuccessful transmission rate, we design an allocation-FLS to correspondingly allocates a certain size of super-time-slot to each node.

There are two antecedents for our allocation-FLS:

- traffic arrival rate (R_a);
- the transmission failure rate (R_{us}).

The consequent is the priority of this node performing transmission (P_t).

We also use antecedent MFs in Fig. 6 and consequent MFs in Fig. 7.

We design our allocation-FLS using rules with one example shown below:

R^l:IF *the traffic arrival rate* (x_1) is *High* and *the unsuccessful transmission rate* (x_2) is *Low*, THEN *the priority of this node performing transmission*(y) should be *Very Low*.

We summarize all rules in Table II.

With the allocation-FLS, the cluster head utilizes the information acquired from TRFR messages to calculate a priority for each node. The node owning the highest priority is the first one to make communications during an *On*-*P*hase.

In summary, we have described the whole process that how to determine, establish and maintain phase-switching schedules for saving energy and communicating successfully among nodes.

III. SIMULATIONS AND PERFORMANCE EVALUATION

We run simulations using OPNET. Nodes are deployed randomly in an area of $1000m \times 1000m$. The radio range is 30 meters, symbol rate is $40ksps$ and data frame length is 1024 bits. For each node, the clock drift rate ranges from 1 to $100\mu s$.

We use the same energy consumption model as in [11] for the radio hardware. To transmit an l-symbol message a distance d, the radio expends:

$$E_{Tx}(l,d) = E_{Tx-elec(l)} + T_{Tx-amp(l,d)} = l \times E_{elec} + l \times e_{fs} \times d^2 \tag{10}$$

TABLE II

THE RULES FOR SUPER-TIME-SLOT ALLOCATION. ANTECEDENT 1 IS TRAFFIC ARRIVAL RATE. ANTECEDENT 2 IS THE UNSUCCESSFUL TRANSMISSION RATE. AND CONSEQUENT IS THE PRIORITY OF THIS NODE PERFORMING TRANSMISSION.

Rule	*Antecedent1*	*Antecedent2*	*Consequent*
1	*Low*	*Low*	*Moderate*
2	*Low*	*Moderate*	*High*
3	*Low*	*High*	*VeryHigh*
4	*Moderate*	*Low*	*Low*
5	*Moderate*	*Moderate*	*Moderate*
6	*Moderate*	*High*	*High*
7	*High*	*Low*	*VeryLow*
8	*High*	*Moderate*	*Low*
9	*High*	*High*	*Moderate*

and to receive this message, the radio expends:

$$E_{Rx} = l \times E_{elec} \tag{11}$$

The electronics energy, E_{elec}, as described in [11], depends on the factors such as coding, modulation, pulse-shaping and matched filtering, and the amplifier energy, $e_{fs} \times d^2$ depends on the distance to the receiver and the acceptable bit error rate. In this paper, we choose: $E_{elec} = 50nJ/syn$, $e_{fs} = 10pJ/sym/m^2$.

A. TRFR Message Successful Transmission Probability

Fixing the duration of *T*RFR-*P*hase at 5, 10, 15, 20, 25 and 30 seconds separately and increasing the number of nodes in a cluster from 5 to 30, we obtain a series of curves on successful transmission rate of data packets (see Fig. 8). Notice that, if *T*RFR-*P*hase duration is longer than 10s, TRFR message for each node has almost 99% probability to be sent successfully to the cluster head. This result proves that, for our algorithm, the cluster head can acquire the necessary information from normal nodes to determine system schedules successfully.

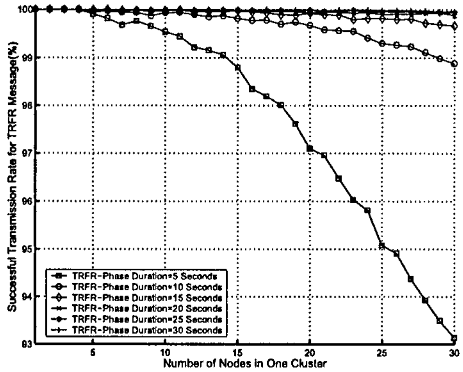


Fig. 8. Successful Transmission Rate for TRFR Message

B. ASCEMAC Adaptation

We investigate the influences of the network density and the traffic intensity on the system performance of our algorithm. In Fig. 9, we plot the number of nodes in a cluster versus successful transmission rate of data packets. We run the simulations under 4 different average clock drift rate, i.e., 0.0, 0.001, 0.01 and 0.1ms/s. Observe that, for each clock drift rate, the vibration of successful transmission rate with the change of nodes number is less than $85.714\% - 83.606\% = 2.108\%$. In Fig. 10, we compare the successful transmission rate at the traffic arrival rate of 0.1, 0.2 and 0.5 pks/s. It shows that, the vibration of successful transmission rate with the change of nodes number is less than $97.099\% - 96.087\% = 1.012\%$. These two experiments show that our ASCEMAC is a network density and traffic intensity adaptive method.

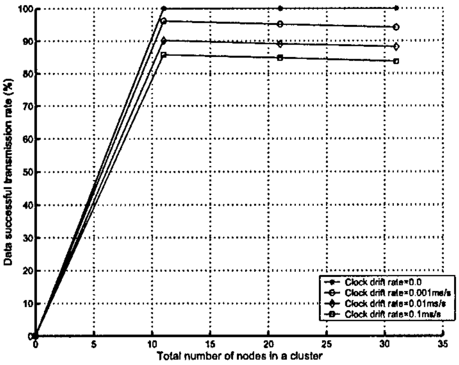


Fig. 9. Successful Transmission Rate

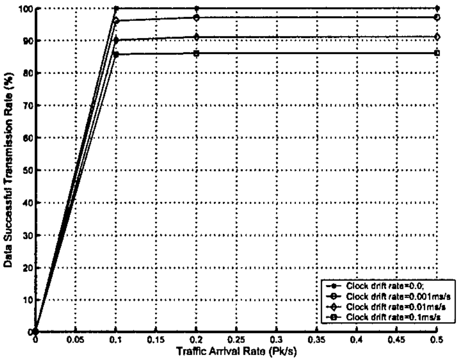


Fig. 10. Successful Transmission Rate

C. ASCEMAC vs. S-MAC and TRAMA

We compare our ASCEMAC against S-MAC and TRAMA. In Fig. 11, we plot the average clock drift rate versus average energy consumption. Notice that ASCEMAC can save about from 68.263% to 189.232% energy per packet compared to TRAMA and S-MAC. That means, when ASCEMAC is used instead of TRAMA, the lifetime for a same WSN can be increased at least one time, and for S-MAC the lifetime even can be increased at least three times. From this experiment, notice that the schedule-based MAC protocols have better performance on energy saving than contention-based MAC protocols. The reason is that some energy is consumed through making competition for accessing the common channel for contention-based MAC protocols.

In Fig. 12, we compare the average waiting time of data packets. Observe that our ASCEMAC has about 56.178% shorter waiting time than TRAMA, and about 8.648% shorter waiting time than S-MAC. Moreover, in this experiment, we set W_{max} to 12seconds. We found that the average waiting times for ASCEMAC are smaller than W_{max} even at different clock drift rates. But for S-MAC and TRAMA, the average waiting times are longer than W_{max} when the clock drift rate is bigger than 0.01ms/s. Hence, ASCEMAC is more sensitive to the traffic lifetime requirement than S-MAC and TRAMA.

In Fig. 13, we plot the average clock drift rate versus the successful transmission rate. It can be seen that our ASCEMAC outperforms the

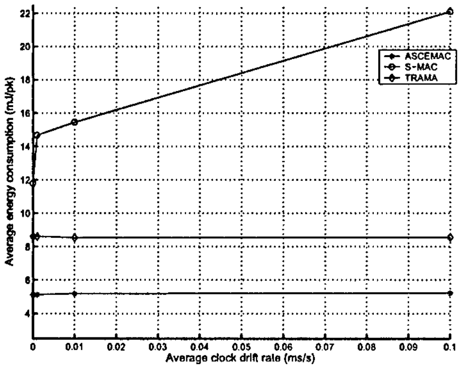


Fig. 11. Average Energy Consumption

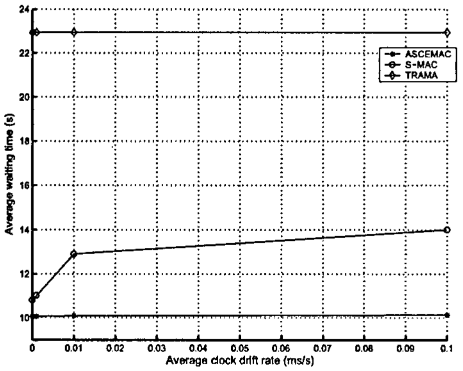


Fig. 12. Average Waiting Time

S-MAC for about 12.5% higher data successful transmission rate. There are almost same successful transmission rate for ASCEMAC and TRAMA. But, the energy saving and waiting time performance for ASCEMAC is much better than those for TRAMA.

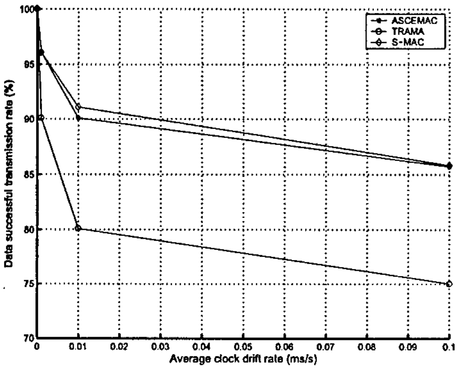


Fig. 13. Successful Transmission Rate

IV. Conclusion

In this paper, we propose a novel energy-efficient MAC protocol for wireless sensor nestworks: ASCEMAC. ASCEMAC makes following contributions, compared to existing energy-efficient MAC protocols:

- Saving energy at MAC layer through reducing the energy consumption on collision and idle, and trading off data waiting time;
- Utilizing free-running scheme and schedule broadcast to set up phase-switching schedules without establishing global clock within a cluster;
- Exploiting a rescheduling method, instead of time synchronization, to handle mismatch caused by clock drifts, as well as taking advantage of fuzzy logical theory, which has distinctive capabilities for coping with uncertainty, to act as our rescheduling-FLS;
- Designing a time-slot allocation system, allocation-FLS, based on traffic intensity and unsuccessful transmission rate;
- Proposing a traffic intensity and network density-based model to acquire optimal power on/off duration, interval of schedule broadcast, super-time-slot length and order.

Simulation results show that our algorithm successfully acquire the optimum values of essential algorithm parameters to ensure the average successful transmission rate, decrease the data packet average waiting time, and reduce the average energy consumption. Therefore, network performance is improved and network lifetime is extended by using our algorithm.

Acknowledgement

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Program Award under Grant N00014-03-1-0466.

References

[1] C. K. Toh, " Ad hoc mobile wireless networks: protocols and systems ", Prentice-Hall, NJ, 2002.

[2] V. Rajendran, K. Obraczka, J. Juslin, O. Koukousoula, " Energy-efficient, collision-free medium access control for wireless sensor networks ", *SenSys'03,* Novermber 5-7,2003, Los Angeles, California, USA.

[3] W. Ye, J. Herdemann, D. Estin " An energy-efficient MAC protocol for wireless sensor networks ", *IEEE INFOCOM,* 2002.

[4] T. V. Dam, K. Langendoen, " Adaptive energy-efficient MAC protocol for wireless sensor networks ", *SenSys'03,* Novermber 5-7,2003, Los Angeles, California, USA.

[5] G. Lu,B. Krishnamachari, C. S. Raghavendra, " Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks", *IEEE International Conference on Performance, Computing, and Communications,* pp.701-706, April 2004.

[6] C. Ma, M. Ma, Y. Yang, " Data-centric energy efficient scheduling for densely deployed sensor networks", *IEEE International Conference on Communications,* Vo. 6, June 20-24 2004.

[7] J. E. Elson " Time synchronization in wirless sensor networks", Dissertation of Computer Science Dept., Univ. of California Los Angeles, 2003.

[8] E. H. Mamdani, " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics,* vol. 26, no. 12, pp.1182-1191, 1977.

[9] L. Zhao, X. Hong, Q. Liang, " Energy-Efficient Self-Organization for Wireless Sensor Networks: A Fully Distributed Approach", *IEEE Globecom,* Dallas, TX, Nov. 2004.

[10] J. M. Mendel, " Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions", Prentice Hall, NJ, 2001.

[11] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, " An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS,* Vol. 1, no.4, OCT. 2002.

[12] L. F. W. van Hoesel, H. J. Kip, P. J. M. Havinga, " Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs ", *IEEE VTC,* Italy, May 2004.

Energy-Efficient Query in Sensor Database Systems with Uncertainties

Qingchun Ren and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019-0016
Email: ren@wcn.uta.edu,liang@uta.edu

Abstract— **Query processing methods have been studied extensively in the context of database systems. But they are not directly applicable in sensor database systems due to the characteristics of sensor networks: the decentralized nature of sensor networks, the limited computational power and energy scarcity of individual sensor node, and imperfect information recorded. In this paper, we propose an energy-efficient query optimization algorithm (QOA) for imperfect information in sensor database systems. We employ an in-network query processing method, which tasks sensor networks through declarative queries. Given a query, our QOA will generate an energy efficient query plan for in-network query processing. Moreover, our algorithm can explicitly exposes uncertainty and ambiguity of query results to database users. As we know, it is troublesome or even impossible to keep a large number of data in sensor database systems for network resource constraints. In our algorithm, we formulate the probability distribution functions (PDFs) of measurement uncertainties according to the knowledge on observation coverage and devices utilized, instead of estimating them from prior data. The simulation results demonstrate that our algorithm can vastly reduce resource usage and thus extend the lifetime of sensor database system.**

I. Introduction

Recent developments in integrated circuit technology have allowed the construction of low-cost small sensor nodes with signal processing and wireless communication capabilities. Distributed wireless sensor networks (WSNs), which are basically composed of sensor nodes through ad hoc networking, have increasing potential applications. WSNs have been applied in military sensing, physical security, air traffic control, environment monitoring and structures monitoring[1], etc..

From a data storage point of view, WSN can be regarded as a distributed database, sensor database system (SDS). Each node in a WSN takes time-stamped measurements of physical phenomena such as heat, sound, light, pressure, or motion. SDSs, compared to traditional database systems, store the data within the network and allow queries to be injected anywhere in the network. Data distribution along with data replication makes the entire system more robust to failures and can provide increased bandwidth and throughput, as well as greater data availability. But, this distributed nature makes the query processing significantly harder for query optimization.

Moreover, WSNs are often developed to run unattended for years. This calls for not only robust hardware and software, but also lasting energy resources. However, current sensor nodes are usually powered by limited batteries [2], and replacing or recharging batteries, in many cases, may be impractical or uneconomical. A recent study [11] shows that, in data collection application, about 40% of energy consumption is due to communication and 58% is due to sensing. Therefore, in devising the best overall execution plan, data queries designed for SDSs should be highly efficient and optimized in terms of energy on communication and data sensing.

Data aggregation [2] techniques have been investigated recently as efficient approaches to achieve significant energy reservation in SDSs. The main idea of data aggregation is that aggregation points combine data arriving from different sensor nodes, eliminate redundancy, and minimize the number of transmissions before forwarding data to the base station. Some important existed works are shown in [7], [8], [9], etc.. But, in this paper, we exploring a new approach, disk covering method, to reduce the information redundance on communication and sensing for energy reservation.

Another target of this paper is about how to reason query uncertainty from imperfect information in SDSs. "Imperfect information is ubiquitous-almost all the information that we have about the real world is not certain, complete and precise" [10]. These include examples such as measurement and recording errors, missing data, incompatible scaling, obsolescence, and data aggregation. Therefore, such imperfection is a fact in database systems. Nowadays, more and more database designers switch to study the whole problem, including certain information and uncertain information, to more accurately describe the real world through database systems.

An extensive survey of the work done in the database and artificial intelligence communities on imperfect information is given in [12]. It points out that in order to build useful information systems, it is necessary to learn how to represent and reason with imperfect information. Notice that uncertain information is typically handled by attaching a number, which represents a subjective measure of the certainty of the uncertain element according to some observer. The way in which the number is manipulated depends on the theory that underlies the number. There are possibilistic, probabilistic and fuzzy approaches [13] [14] [15] [16] [17]. Most of them just consider how to represent uncertain information in the database system, and how to make relational calculus among relations. But, how to reason uncertainty with imperfect information is seldom studied.

In this paper, we propose an energy-efficient query optimization algorithm (QOA) for imperfect information in SDSs. We employ an in-network query processing method, which tasks sensor networks through declarative queries. Given a query, our algorithm will generate an energy efficient query plan for in-network query processing. The optimized query plan can vastly reduce resource usage and thus extend the lifetime of SDSs. Moreover, our algorithm can explicitly exposes uncertainty and ambiguity of query results to database users. As we know, it is troublesome or even impossible to keep a large number of data in sensor database systems for network resource constraints and environment uncertainties. In our algorithm, we manage uncertainties using probability theory as in [6] and [5], but the probability distribution functions (PDFs) of measurement uncertainty are formulated according to the knowledge on observation coverage and devices.

The remainder of this paper is organized as follows: in Section II, we provide some preliminaries on vector space model and k-partial set cover problem; Section III presents our algorithm; simulation results are given in Section IV; Section V concludes this paper.

II. Preliminaries

A. Vector Space Model

Vector Space Model (VSM) [18] [19] is a way to represent documents through the words that they contain. It has been widely used in the traditional information retrieval (IR) field [20] [21]. Most search engines also use similarity measures based on this model to rank Web documents. VSM creates a space in which both documents and queries are represented by vectors. For a fixed collection of documents, an m-dimensional vector is generated for each document and each query from sets of terms with associated weights. Then, a vector similarity function, such as the inner product, can be used to compute the similarity between a document and a query.

In VSM, weights associated with the terms are calculated based on the following two numbers:

- term frequency, f_{ij}, the number of occurrence of term y_i in document x_i; and
- inverse document frequency, $g_i = log(N/d_j)$, where N is the total number of documents in the collection and d_j is the number of documents containing term y_i.

The similarity $sim_{vs}(q, x_i)$ between a query q and a document x_i can be defined as the inner product of the query vector Q and the document vector X_i:

$$sim_{vs}(q, x_i) = Q \cdot X_i = \frac{\sum_{j=1}^{m} v_j \cdot w_{ij}}{\sqrt{\sum_{j=1}^{m}(v_j^2) \cdot \sum_{j=1}^{m}(w_{ij}^2)}} \quad (1)$$

where m_i is the number of unique terms in the document collection. Document weight $w_{i,j}$ and query weight v_j are

$$w_{ij} = f_{ij}w_{ij} = f_{ij}log(N/d_j) \qquad and$$

$$v_j = \begin{cases} log(N/d_j) & y_j\, is\, a\, term\, in\, q \\ 0 & otherwise. \end{cases} \quad (2)$$

B. k-Partial Set Cover Problem

Covering problems are widely studied in discrete optimization. Basically, these problems involve picking a least-cost collection of sets to cover elements. Classical problems in this framework include general set cover problems and partial covering problems. k-partial set cover problem [23] as a partial covering problem is about how to choose a minimum number of sets to cover at least k elements, and which k elements should be chosen.

k-partial set cover problem can be formulated as an integer program as following.

MINIMIZE:

$$\sum_{j=1}^{m} c(S_j) \cdot x_j \quad (3)$$

SUBJECT TO:

$$y_i + \sum_{j: t_i \in S_j} x_j \geq 1 \qquad i = 1, 2, \ldots, n, \quad (4)$$

$$\sum_{i=1}^{n} y_i \leq n - k, \quad (5)$$

$$x_j \geq 0 \qquad j = 1, 2, \ldots, m, \quad (6)$$

$$y_i \geq 0 \qquad j = 1, 2, \ldots, n, \quad (7)$$

Where $x_i \in \{0,1\}$ corresponds to each $S_j \in S$. Iff set S_j belongs to the cover, then $x_j = 1$. Iff set t_j is not covered, then $y_i = 1$. $t_i \in \Gamma$. Clearly, there could be at most $n - k$ such uncovered elements.

III. Our Algorithm Description

In a SDS, when a query is submitted, common rules for active sensor nodes selection is generated based on the query. And then, each sensor node determines whether itself will participate this query processing or not according to its location, remaining energy and measurement accuracy through query optimization algorithm. Finally, the chosen sensor nodes collect data and send them back to the sink with uncertainty. In our algorithm, active sensor node is defined as a sensor node which collects data and makes responds during a query processing.

A. Network Model

In SDS, a large number of sensor nodes are deployed over an area. All nodes are interconnected to one or more gateways by means of wireless links. Gateways are in charge of relaying data to a base station.

For the reasons of deployment itself of SDSs, it is difficult or even impossible to exactly pre-determine the locations of sensor nodes. After all sensor nodes have been deployed, each node sends its location information to sinks through certain messages, such as beacons. The topologies of the area controlled by each gateway will be formed according to these information. Then, all topology information will be gathered at the base station. We assume that each sensor node in our algorithm is capable of acquiring its own location through certain methods.

B. Query VSM Design

With high network density and topology unpredetermination characteristics, sensing range overlapping in SDSs among nodes are unavoidable and space variant (the node density is not uniform over the network). It is the main reason to create redundancy data. Communicating and storing these redundancy data is one of the biggest sources of wasting energy during query processing. But on the other side of coin, it is a method to increase the confidence of query results. Therefore, there is a trade off between increasing the confidence of query answer and saving energy.

We solve the high information redundancy through controlling the number of nodes to response queries. It is obviously that the less the number of nodes communicate and sense during a query, the less the energy is consumed. But the problem is the query results supplied by partial nodes should reflect the whole area's condition at acceptable degree, otherwise it is uselessness for database users. Therefore, the key issue is to determine how many nodes and which nodes should be selected for a query.

Following factors are considered by us for this issue:

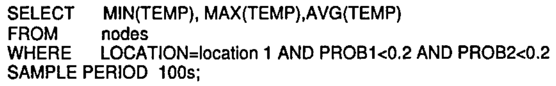
- Sensor Location
  Since sensors' location directly determines which area can be observed. Given a piece of area and some nodes over this area, in order to employ as few as possible nodes to cover as large as possible area, we should select those nodes which locate optimum locations. We discuss how to determine optimum locations for a query in the following part (Optimum Location Determination Section).
- Measurement Accuracy
  Since the cost and the measurement accuracy of sensor nodes are related with each other, sensor nodes owning different accuracy levels are deployed simultaneously in a SDS for economical reason. Furthermore, through a query, database users supply not only what information they want to retrieval, but also the requirement on uncertainties of query results. In this case, we should select the nodes whose measurement accuracy are close to database users' requirement.
- Battery Level
  The battery level of sensor nodes is our third factor of nodes selection. When the power of a node is used up, the data observed by this node will be missed, which will reduce the confidence at some degree. This inspires us to select the nodes with more remaining battery so that a query processing can be completed by all chosen nodes.

In our algorithm, we employ VSM to combine all considering factors to select the most related nodes to participate a query processing. Our goal is to use as little energy as possible and more suitable sensor nodes to supply satisfied query results.

In our query VSM, the query vector is designed as $\langle R_l, A_d, B_m \rangle$. Where

- R_l stands for location relativity. It is the indicator of the distance between the location of a sensor node and the optimum location. If their positions are exactly match, in this case, R_l=1.
- A_d stands for measurement accuracy. It is the indicator of sensor nodes' measurement accuracy. A_d equals to the probability distribution function (PDF) of each node' measurement error. For example, the measurement accuracy of CXM539 is $100\mu T$ (1mGauss). In this case, A_d= 0.002.
- B_m stands for remaining battery. It is the indicator on how much power remains for a sensor node. The unit of B_m is J.

After a database user submits a query (shown in Fig. 1), the base station selects the optimum locations of this query, and then translates the query from SQL [4] form into a query VSM vector Q. According to the query given in 1, Q=$\langle 1, 0.2, 5 \rangle$. We assume the maximum energy for nodes is $5J$.



```
SELECT   MIN(TEMP), MAX(TEMP),AVG(TEMP)
FROM     nodes
WHERE    LOCATION=location 1 AND PROB1<0.2 AND PROB2<0.2
SAMPLE PERIOD  100s;
```

Fig. 1. SQL query

After receiving Q, each node starts to updates its own query VSM vector (i.e., h_i (i=1,2,$\cdots$, n)), h_i=$\langle R_{l,i}, A_{d,i}, B_{m,i} \rangle$. We assume there are n nodes in this network totally. $R_{l,i}$ is defined as:

$$r_{l,i,j} = \frac{\sqrt{(x_i - x_{0,j})^2 + (y_i - y_{0,j})^2}}{K} \quad and$$
$$R_{l,i} = \{r_{l,i,1}, r_{l,i,2}, \cdots, r_{l,i,\nu}\} \tag{8}$$

Where (x_i, y_i) is the location of node i. (x_0, y_0) is position of an optimum location. K is the uniform factor, which ensure the value of $r_{l,i,j}$ is less than one. We assume that there are ν optimum locations for a query.

We design a query correlation indicator γ to express the correlation degree between each node and a query. We formulate γ in (9).

$$\begin{aligned}\gamma_{Q,h_i} &= \max_j\{q_j \cdot h_{i,j}\} \\ &= \frac{1 \times R_{l,i,j} + (1 - A_d) \times (1 - A_{d,i}) + B_m \times B_{m,i}}{\sqrt{(R_l^2 + (1 - A_d)^2 + B_m^2)(R_l^2 + (1 - A_d)^2 + B_m^2)}}\end{aligned} \tag{9}$$

The higher γ is, the more chance this sensor node take part in this query as active sensor nodes.

C. Active Sensor Nodes Choosing

The query correlation indicator γ_{Q,h_i} are exchanged between neighbors (nodes, which are only one hope apart, are neighbors and can communicate with each other directly). By employing cooperation among nodes, the nodes, which own highest query correlation degree among their neighbors, are picked up to participate this query. The pseudo-codes of active sensor nodes choosing algorithm is given in Fig. 2.

```
//initial the covering set
C $\leftarrow$ null;
//initial the uncovering set, N is the closure of all neighbors of a node
UC $\leftarrow$ N;
//select the active nodes
while UC is not null
        do
        select node i with the highest query correlation;
         if node i is not covered yet
                C $\leftarrow$ {i};
                UC $\leftarrow$ N/{i};
         else
                go to node j with the next highest query correlation;
        end
```

Fig. 2. Algorithm for active nodes choosing

Notice that, running our QOA in each sensor node, the most related sensor nodes are chosen to answer the query, which are mostly close to the optimum locations, satisfy the uncertainty requirement and own high battery level. But other nodes switch into energy saving mode, i.e., sleep mode. Therefore, the SDS composed by these optimized sensor nodes can highly improve the energy efficiency.

D. Optimum Location Determination

We model the problem, determining optimum locations for a query, as a k-partial set cover problem. We define this problem as follows: Let n be the number of all sensor nodes, p be a given positive integer such that $p \leq n$. If we have k same disks with radius r, which depends on the sensing range of sensor nodes, the k-partial set cover problem try to solve whether k disks can cover at least p nodes. In this paper, we only consider sensor nodes in a plane (the dimension is 2). This kind of k-partial set cover problem is a NP problem.

At present, all known algorithm for NP problems require time that is exponential in the problem size. It is unknown whether there are any faster algorithms. Therefore, to solve an NP problem for any nontrivial problem size, one of the approaches is approximation algorithm, which can acquire the solution during polynomial time. SETCOVER algorithm [23] is a good approximation method to determine the value of k and the locations of these k disks on the plane we interested.

SETCOVER "guesses" the set with the highest cost in the optimal solution by considering each set in turn to be the highest cost set. For each set that is chosen, to be the highest cost set, say S_j , S_j along with all the elements it contains is removed from the instance and is included as part of the cover for this guess of the highest cost set. The cost of all sets having a higher cost than c(S_j) is raised to ∞. I_j = (T^j ,S^j, c', k_j) is the modified instance. SETCOVER then calls PRIMALDUAL on I_j which uses a primal dual approach [24] to return a set cover for I_j. In PRIMALDUAL, the dual variables u_i are increased for all $t_i \in T_j$ until there exists a set S_a, so that $\sum_{a:t_i \in S_a} u_i = c'(S_a)$. Sets are chosen this way until the cover is feasible. The algorithm then chooses the minimum cost solution among the m solutions found.

After the value of k and the locations of k disks are obtained, in our algorithm, we choose the centers of those k disks as our optimum locations. Since these k disks can almost cover all sensor nodes in certain area, the sensor nodes, locating on these locations, can almost monitor all information of the interested area.

E. Uncertainty Acquisition

There are numerous factors introducing uncertainty into the query results, as we mentioned above. In most existed works, the uncertainties of query results are determined by PDFs of measurement uncertainty, which are pre-estimated through history data. If a large number of history data are not available, the performance of those existed algorithms will become worse or even they cannot deal with this condition.

As we know, for general SDSs, the memory size of sensor node is too limited to keep large history information compared to many network terminal devices. For instance, the Berkeley motes have at most 128KB program memory, 4KB RAM, and 512KB external nonvolatile storage [2].

Inspired by this demand, we formulate the PDFs of measurement uncertainties from other information instead of history data. Hereafter we analyze the main factors that introduce the uncertainty into the query results in order to formulate them. These include:

- Observation Coverage
  Observation coverage is the area covered by active sensor nodes during a query processing. Since, the physically observable world consists of a set of continuous phenomena in space, it is impossible to gather all relevant data through nodes whose observation coverages are not continue. In this case, some uncertainties are introduced into the query results by partial observation coverage. We define a observation coverage PDF (f_c) to stand the total coverage of all active sensor nodes for a query.
- Measurement Accuracy
  The quality of sensor node's sensing parts usually boils down to its measurement stability and measurement accuracy. In general, as measurement stability and accuracy increase, so do their power requirements and cost, which are all troublesome for general sensor nodes. For general application, different cost of sensor nodes are deployed. Hence, some uncertainties are introduced into the query results by measurement errors. For example, speed detect sensor node, CXM539, is a built-in magnetoresistive sensor. The measurement accuracy is 100 μT ($\pm$1mGauss) [22]. We define a measurement accuracy PDF (f_m) to stand the measurement error produced by related sensor nodes.

We employ formula (10) to calculate the corresponding uncertainty for a query result.

$$P = \int_{\phi} f_m(x)dx \times \int_{\varphi} f_m(y)dy \tag{10}$$

We present a classification of probabilistic queries and examples of common representative uncertainty for each class.

There are different definitions for f_m for each class.

1) *simple aggregation* class
   In this class, an value of an sensor node is returned only, such as MIN and MAX query. In this case, f_m=$f_{m,j}$. Node j is the node which detects the highest/lowest value during this query.
2) *complicate aggregation* class
   In this class, a derivative value over a group of sensor nodes' data is returned, such as AVG query. f_m, now, has the same distribution as $f_{m,j}$. But the mean and variance are $\frac{1}{M}\Sigma_{j=1}^{M}\mu_j$ and $\frac{1}{M}\Sigma_{j=1}^{M}\sigma_j$ separately. We assume there are M active nodes for this query. And for each node, $f_{m,j}$ complies with same distribution with different mean (μ_j) and variance (σ_j). But, if M is big enough, f_m complies with a normal distribution [25].

For example, a database user retrievals the highest, lowest and average temperatures of location 1 and location 2. The query results is given in Table I. Relation database model [3] is employed. TEMPERATURE relation (see I), is specially used to record the temperature information of interested areas. The values of PROB1, PROB2 and PROB3 are calculated using (10).

TABLE I
EXAMPLE OF UNCERTAIN RELATION: TEMPERATURE. PROB1 IS THE UNCERTAINTY WITH THE LOWEST TEMPERATURE. PROB2 IS THE UNCERTAINTY WITH THE HIGHEST TEMPERATURE. PROB3 IS THE UNCERTAINTY WITH AVERAGE TEMPERATURE.

LOCATION	MIN	PROB1	MAX	PROB2	MEAN	PROB3
location 1	30	7.5%	50	5.56%	44.5	10%
location 2	40	4.22%	55	0.02%	48.9	15.89%

IV. Simulations and Performance Evaluation

One hundred sensor nodes are deployed randomly in an area of 10 × $10m^2$, and sensing range is 1m. The initial energy of sensor nodes uniformly distributes within [0,5]J. We run Monte Carlo simulations 1000 times to remove the randomicity of simulation results. We compare our QOA against original query processing method without any query optimization.

The energy consumption model for data sensing is shown in (11).

$$E_{se} = E_{elec} * S_t \tag{11}$$

Where E_{se} is the energy consumed by one query processing. E_{elec} is the energy consumed by once data sampling. S_t is the duration of one query processing, which is defined by database users' query. In this paper, we choose: $E_{elec} = 5m\mu/sample,$

In Fig. 3, we plotted the sampling index versus the nodes dead time. We can see that after processing about 20 times of sampling, all nodes, without QOA, use up their energy. But for QOA, the whole network is not down until 53 times of sampling. Therefore, QOA extends the lifetime of network about 2.5 times.

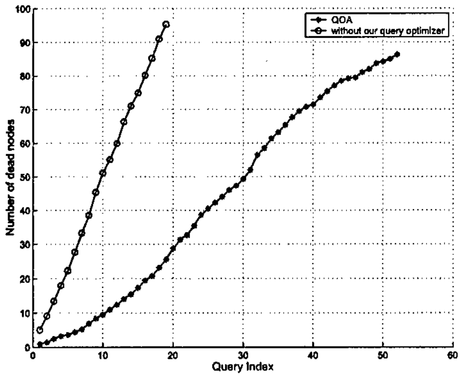


Fig. 3. Nodes Dead Time

In Fig. 4, we compare the observation covering rate of these two schemes. Observed that, QOA outperforms the original query processing method for about employing less $65 - 45 = 20$ nodes to cover 90% area interested.

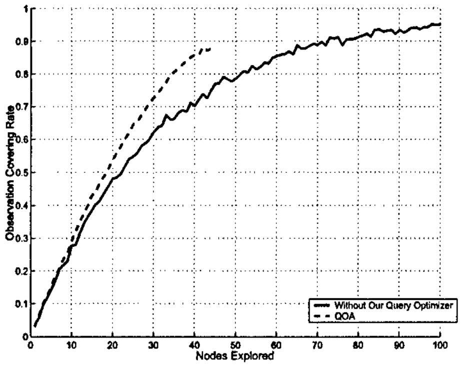


Fig. 4. Observation Coverage Rate

In Fig. 5, we plot the nodes selection rate for QOA. Observed that, at most about 40% nodes are chosen for a query and at least about 15% nodes are active nodes to respond a query. This simulation result illustrate the reason why our QOA can implement energy reservation. That is, about half nodes switch to energy saving model during query processing.

By employing our QOA, the energy is saved and the lifetime of network is extended. But the cost for using our algorithm is the decrease of observation covering rate. In Fig. 6, we plot the observation covering rate decreasing degree. It is shown that, the biggest observation covering rate decrease is 16.6%. But, most of time, the decreasing degree is less than 8%.

V. Conclusions

In this paper, we propose a energy-efficient query optimization algorithm for imperfect information in sensor database systems. We tasks sensor networks through declarative queries.

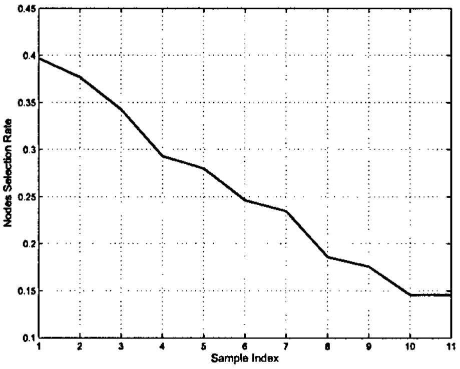


Fig. 5. Nodes Selection Rate

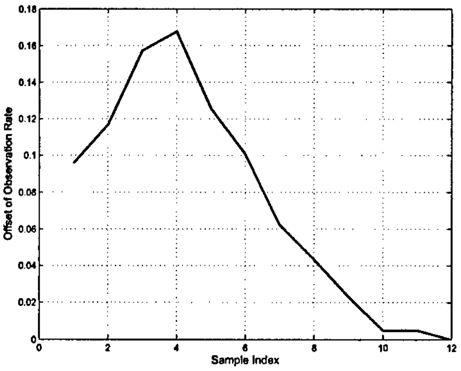


Fig. 6. Decrease of Observation Coverage Rate

Given a query, our query optimization algorithm generates an energy efficient query plan for in-network query processing. Moreover, our algorithm explicitly exposes uncertainty and ambiguity of query results to database users. We formulate the PDFs of measurement uncertainties according to the knowledge on observation coverage and devices employed, instead of estimating them from prior data.

The simulation results prove that our algorithm can vastly reduce resource usage and thus extend the lifetime of sensor database system.

ACKNOWLEDGEMENT

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Program Award under Grant N00014-03-1-0466.

REFERENCES

[1] D. Culler, D. Estin and M. Srivastava, " Overview of Sensor Networks", *IEEE Computer Society,* pp.41-49, Aug. 2004.

[2] F. Zhao, L. Guibas, " Wireless Sensor Networks: an Information Processing Approach", Morgan Kaufmann, CA, 2004.

[3] R. Elmasri, S. B. Navathe, " Fundamentals of Database Systems ", Addison Wesley, CA, 2003.

[4] J. R. Groff, P. N. Weinber, L. Wald, " SQL: The complete reference ", Berkeley, CA, McGraw-Hill/Osborne, 2002.

[5] A. Faradjian, J. Gehrke, P. Bonnet, " GADT: A probability space ADT for representing and querying the physical world", 2002.

[6] E. Wong, " A statistical approach to imcomplete information in database systems", *ACM TRANSACTIONS on DATABASE SYSTEMS,* Vol. 7, No.3, pp.470-488, Sep. 1982.

[7] S. J. Baek, G. D. Veciana, X. Su, " Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation", *IEEE JNL,* Vol. 22, No.6, Aug. 2004.

[8] F. Ordonez, B. Krishnamachari, " Optimal information extraction in energy-limited wireless sensor networks", *IEEE JNL,* Vol. 22, No.6, Aug. 2004.

[9] J. Carle, D. Simplot-Ryl, " Energy-efficient area monitoring for sensor networks", *IEEE JNL,* Vol. 37, No.2, Feb. 2004.

[10] V. Owei, " An intelligent approach to handling imperfect information in concept-based natural languange queries", *ACM TRANSACTIONS,* Vol. 20, No. 3, pp.291-328, Jul. 2002.

[11] S. Madden, " The design an evaluation of a query processing architecture for sensor networks", Master's thesis, UC Berkeley, 2003.

[12] S. Parsons, " Current approaches to handling imperfect information id data and knowledge bases", *IEEE TKDE,* pp.353-372, Aug. 3, 1996.

[13] H. Prade, C. Testemale, " Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries", Information Sciences, 34, pp.115-143, 1984.

[14] L. V. S. Lakshmanan, " ProbView: A flexible probabilistic database system", *ACM TRANSACTIONS,* Vol. 22, NO.3, Sep. 1997.

[15] D. Barbara, H. Garcia-Molina, D. Porter, " The management of probablistic data," *IEEE TRANS,* Vol. 4, No.5, Oct. 1992.

[16] H. Prade, C. Testemale, " Fuzzy relational database: representational issues and reduction using similarity measures", *Journal of the American Society for Information Science,* 38, pp.118-126, 1987.

[17] K. V. S. V. N. Raju, A. K. Majumdar, " Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems", *ACM TRANS,* Vol. 13, NO.2, Jun. 1988.

[18] D. Chow, C. T. Yu, " On the construction of feedback queries", *Journal of the Association for Computing Machinery,* Vol. 29, NO.1, pp.127-151, Jan. 1982.

[19] S. K. M. Wong, W. Ziarko, V. V. Raghavan, P. C. N. Wong, " On modeling of information retrieval concepts in vector spaces ", *ACM Transactions on Database Systems (TODS),* Vol. 12, NO.2, 1987.

[20] L. Gravano, H. Garcia-Molina, and A. Tomasic, " Gloss: Text-source discovery over the internet", *ACM TRANS,* Vol. 24, NO.2, Jun. 1999.

[21] D. Grossman, O. Frieder, D. Holmes, and D. Roberts, " Integrating structured data and text: A relational approach", *Journal of the American Society for Information Science,* Vol. 48, NO.2, Feb. 1997.

[22] "http://xbow.com" Products Description.

[23] R. Gandhi, S. Khuller, A. Srinivasan, " Approximation algorithm for partial covering problems", *Journal of Algorithms,* Vol. 53, No. 1, pp.55-84, Oct. 2004.

[24] M. X. Goemans, D. P. Willianson, " A general approximation technique for constrained forest problems ", SIAM J. Comput. pp.296-317, 1995.

[25] D. Bersekas and R. Gallager, "Data Networks", Prentice Hall, NJ, 1987.

CIHSPS 2005 - IEEE International Conference on
Computational Intelligence for Homeland Security and Personal Safety
Orlando, FL, USA, 31 March – 1 April 2005

Fuzzy Deployment for Wireless Sensor Networks

Liang Zhao and Qilian Liang

Department of Electrical Engineering,
University of Texas at Arlington,
Arlington, TX 76010, USA
Phone: +1 817 272 3488, Fax: +1 817 272 2253, E-mail: zhao@wcn.uta.edu, liang@uta.edu.

Abstract *– In this paper, we are concerned with developing a fuzzy deployment for wireless sensor networks. Traditional deployments often assume a homogeneous environment, which ignores the effect of terrain profile and obstacles such as buildings, trees and so on. Nevertheless, in many applications, some areas need to be more critically monitored. All these factors are combined together through Fuzzy Logic System in our proposed scheme. Simulation results show that the Fuzzy Deployment improves the worst-case coverage by around 5 dB.*

Keywords *– Deployment, fuzzy logic, propagation modeling.*

I. INTRODUCTION

In this paper, we are concerned with developing a fuzzy deployment for wireless sensor networks (WSN). Traditional deployments often assume a homogeneous environment [1], which ignores the effect of terrain profile and obstacles such as buildings, trees and so on. Such approaches have proved inaccurate in the practice of cellular networks. In fact, many propagation models, based on theoretical calculation and/or empirical data, have been proposed to predict path loss over irregular terrain. For instance, the Longley-Rice model [2, 3], also known as the ITS irregular terrain model, was proposed to predict large-scale median transmission loss relative to free space loss over irregular terrain. The Longley-Rice method operates in two modes, namely, point-to-point and area modes. Taking a similar approach, Durkin et al. [4, 5] proposed a computer simulator to predict field strength contours over irregular terrain, which was adopted by U.K. JRC for the estimation of effective mobile radio coverage areas. As a standard for system planning in Japan, Okumura's model [6] is widely used for signal prediction in urban areas. None of these works have not been taken into consideration in current WSN research.

Nevertheless, in many applications, some areas need to be more critically monitored. For example, if there is a road through the area of interest, and chances are that targets would follow this road, then it would be advisable to deploy more sensors around this road. In this paper, we utilize Fuzzy Logic System to combine all these factors together to achieve a better deployment.

The rest of this paper is organized as follows. Section II introduces the preliminaries that our research is based on. A Fuzzy Deployment scheme is proposed in Section III and simulations are given in Section IV. Section V concludes this paper.

II. PRELIMINARIES

A. Overview of Fuzzy Logic Systems

Figure 1 shows the structure of a fuzzy logic system (FLS) [7]. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets. Consider a p-input 1-output FLS, using singleton fuzzification, *center-of-sets* defuzzification [8] and "IF-THEN" rules of the form [9]

$$R^l : \text{IF } x_1 \text{ is } \mathrm{F}_1^l \text{ and } x_2 \text{ is } \mathrm{F}_2^l \text{ and } \cdots \text{ and } x_p \text{ is } \mathrm{F}_p^l, \text{ THEN } y \text{ is } \mathrm{G}^l.$$

Assuming singleton fuzzification, when an input $\mathbf{x}' = \{x_1', \dots, x_p'\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{\mathrm{F}_1^l}(x_1') \star \mu_{\mathrm{F}_2^l}(x_2') \star \cdots \star \mu_{\mathrm{F}_p^l}(x_p') = \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i') \quad (1)$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm. There are many kinds of defuzzifiers. In this paper, we focus, for illustrative purposes, on the center-of-sets defuzzifier [8]. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')$, so that

$$y_{cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c_{\mathrm{G}^l} \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')}{\sum_{l=1}^{M} \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')} \quad (2)$$

where M is the number of rules in the FLS.

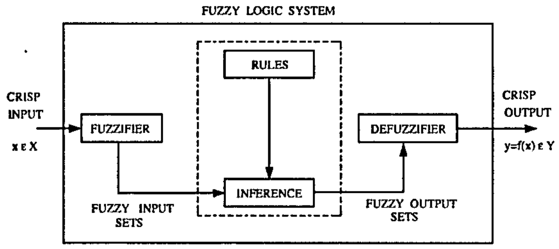


Fig. 1. The structure of a fuzzy logic system.

B. Coverage

Grid-based approaches are often used to compute the coverage provided by the sensor networks [10, 11]. However, for resolution and complexity considerations, Voronoi-based approaches are required in some situations. Thanks to its property that the Voronoi vertexes partitions the plane into a set of convex polygons such that all point inside a polygon are closest to only one site, it has been widely used to determine the best- and worst-case coverages [12]. For illustrative purposes, only worst-case coverages are considered in this paper. Considering the targets as sources of signals, the received signal strength can be found by subtracting overall path loss from the radiated power plus antenna gains, expressed in dB. Thus, assuming the propagation is bidirectionally symmetric, the coverage can be represented by the overall path loss observed at the vertexes of Voronoi polygons.

C. Propagation Model

In previous work [13], general long-distance path loss models are often used, which assume the average large-scale path loss is expressed as a function of distance by using a path loss exponent, n [14].

$$\bar{P}L(d) = \bar{P}L(d_0)(\frac{d}{d_0})^n \tag{3}$$

where n is the path loss exponent, which indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance, which is determined from measurement close to the transmitter, and d is the distance from the source to the receiving point. However, the propagation often takes place over irregular terrain, and the effect of terrain profile in many cases is not negligible. Based on a systematic interpretation of measurement data obtained in different areas, a number of propagation models are developed to predict signal strength. For example, work by Walfisch and Bertoni [15] considers the impact of rooftops and building height by using diffraction to predict average signal strength at street level. Since the rows or blocks of buildings are viewed as diffracting cylinders lying on the earth in the development of this model, it is also applicable to obstacles such as trees, shrubs and so on. In this model, the pass loss, S, is a product of three factors, namely, P_0, Q^2 and P_1, which is due to free space path loss, the reduction in the rooftop signal and diffraction, respectively. When expressed in dB, the overall path loss is given by

$$L_p = L_0 + L_{ex} \tag{4}$$

where L_0 is the free space path loss and L_{ex} is the excess path loss due to terrain profile. In this paper, we only consider area-mode path loss, in which the whole area of interest is divided in to smaller subareas and each subarea has a different value of L_{ex}. Although this is a simple model, it satisfies our requirement of accuracy. More accurate model could be used at the cost of complexity.

III. FUZZY DEPLOYMENT

A. Problem Formulation

In this work, a simple theoretical propagation is used. However, the fuzzy deployment is also applicable to any other propagation model. First, we assume the area of interest is divided into square subareas, each of which has its own terrain profile and required level of surveillance, which can be translated into area path loss $PL(i)$ and required threshold of path loss $PL_{TH}(i)$. Second, it is possible to control the number of sensor nodes sprayed in each subarea. Third, such spray is uniformly random in each subarea. Then the problem is to determine the number of sensor nodes needed in the ith subarea, $n(i)$. Any deployment that assumes a homogeneous environment would have to deploy the same amount of sensors into each subarea, which can not meets the requirements most likely. From common sense, we know that it would advisable to deploy more sensors to those areas with larger area path loss and higher level of surveillance, though such relationship is not easy to determine. However, fuzzy logic systems have demonstrated their power in utilizing such subjective knowledge. Therefore, we apply fuzzy logic to this problem to determine the best number of sensors for each subarea.

B. Scheme description

For the ith subarea, its area path loss $PL(i)$ and required threshold of path loss $PL_{TH}(i)$ are normalized to $[0, 10]$ by convention. Either antecedent, $PL(i)$ or $PL_{TH}(i)$, has three overlapping membership functions covering the whole input space as shown in Fig.2. The overlaps between the membership functions are to guarantee that more rules are fired for a specific input so that the decision is distributed to more rules and thus robustness is improved. Such a choice of membership functions provides $M = 3^2 = 9$ rules.

The rules are designed such as:

R^l : IF $PL(i)$ is *low* and $PL_{TH}(i)$ is *high*,
THEN $weight(i)$ is $\bar{w}^l$.

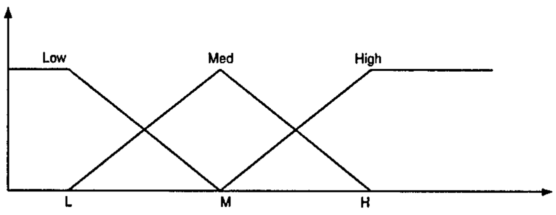


Fig. 2. Membership functions for $T(PL)$ and $T(PL_{TH}) = \{Low, Medium, High\}$.

where $\bar{w}^l$ is an integer in $[1,9]$. In our design, a different $\bar{w}^l$ is used for each of the M rules. The whole rule base is shown in Fig.3. The numbers in the figure are the values of $\bar{w}^l$'s associated with the respective rules. For example, the upper-right cell with the number "9" means:

$$R^9 : \text{IF } PL(i) \text{ is } high \text{ and } PL_{TH}(i) \text{ is } high, \\ \text{THEN } weight(i) \text{ is } 9.$$

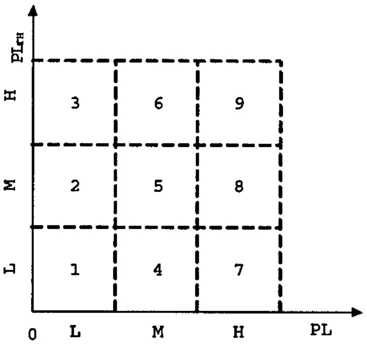


Fig. 3. Rule base for the Fuzzy Deployment.

Although fuzzy logic systems are universal approximators, the desired dynamic can only be captured by enough free parameters. Dividing the input space into more overlapping zones can give us finer resolution at the cost of higher complexity, thus, the number of design parameters is a measurement of both resolution and complexity. Our design has 30 free parameters as shown in Table 1.

Table 1. Number of free parameters in the Fuzzy Deployment.

Number of antecedent parameters	21
Number of consequent parameters	9
Total	30

However, for comparative purposes, the number of free parameters is often counted when all the membership functions are chosen to be unnormalized Gaussian functions, i.e.,

$$\mu_A(x) = exp\{-\frac{1}{2}\frac{(x-m_A)^2}{\sigma_A^2}\}, \tag{5}$$

where m_A and σ_A are the mean and standard deviation, respectively. If we replace the triangle and trapezoidal membership functions in Fig.2 by the Gaussian membership functions in Fig.4, the number of free parameters is 45 as listed in Table 2.

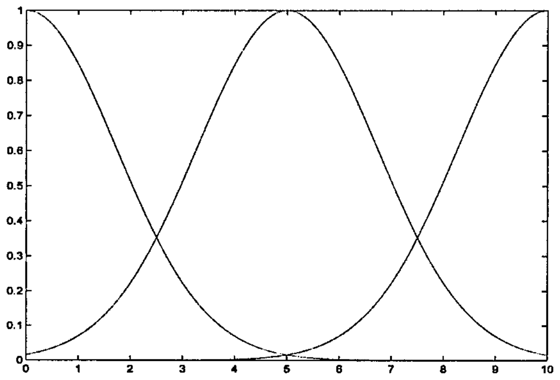


Fig. 4. Gaussian Membership functions for $T(PL)$ and $T(PL_{TH}) = \{Low, Medium, High\}$.

Table 2. Number of free parameters for Gaussian membership functions in the Fuzzy Deployment.

Number of antecedent parameters	36
Number of consequent parameters	9
Total	45

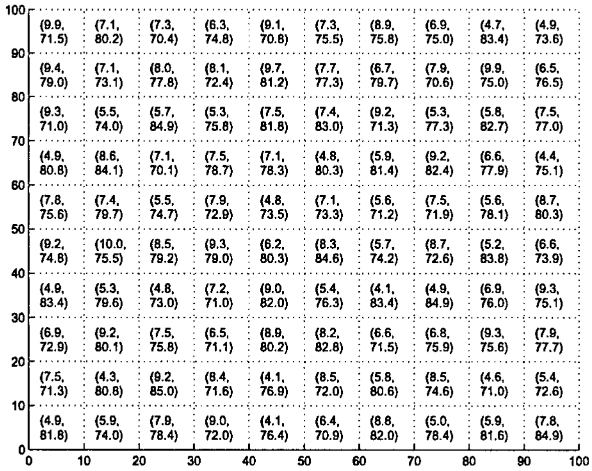
Finally, for the ith subarea, the FLS computes an output $weight(i)$ according to (2). Then $n(i)$ is determined by (6).

$$n(i) = \frac{weight(i)}{\sum_j weight(j)} \tag{6}$$

IV. SIMULATIONS

We conducted computer simulations to compare the Fuzzy Deployment with traditional uniform deployment. Specifically, we consider a scenario in which irregular terrain profile causes variation in the propagation. A $1km \times 1km$ area is divided into 100 square subareas with the size of $100m \times 100m$, and each subarea has its own specific terrain profile. Given 1000 sensors to deploy in this area, a traditional deployment would spray 10 nodes into each subarea, while the Fuzzy Deployment would adaptively determine the number needed for each subarea.

We ran traditional and fuzzy deployment on 200 randomly generated maps and took the average. The random maps were generated as follows. For the ith $100m \times 100m$ subarea, there is an area path loss $PL(i)$ and a path loss threshold $PL_{TH}(i)$, which represent the area terrain profile and required



(9.9, 71.5)	(7.1, 80.2)	(7.3, 70.4)	(6.3, 74.8)	(9.1, 70.8)	(7.3, 75.5)	(8.9, 75.8)	(6.9, 75.0)	(4.7, 83.4)	(4.9, 73.6)
(9.4, 79.0)	(7.1, 73.1)	(8.0, 77.8)	(8.1, 72.4)	(9.7, 81.2)	(7.7, 77.3)	(6.7, 79.7)	(7.9, 70.6)	(9.9, 75.0)	(6.5, 76.5)
(9.3, 71.0)	(5.5, 74.0)	(5.7, 84.9)	(5.3, 75.8)	(7.5, 81.8)	(7.4, 83.0)	(9.2, 71.3)	(5.3, 77.3)	(5.8, 82.7)	(7.5, 77.0)
(4.9, 80.8)	(8.6, 84.1)	(7.1, 70.1)	(7.5, 78.7)	(7.1, 78.3)	(4.8, 80.3)	(5.9, 81.4)	(9.2, 82.4)	(6.6, 77.9)	(4.4, 75.1)
(7.8, 75.6)	(7.4, 79.7)	(5.5, 74.7)	(7.9, 72.9)	(4.8, 73.5)	(7.1, 73.3)	(5.6, 71.2)	(7.5, 71.9)	(5.6, 78.1)	(8.7, 80.3)
(9.2, 74.8)	(10.0, 75.5)	(8.5, 79.2)	(9.3, 79.0)	(6.2, 80.3)	(8.3, 84.6)	(5.7, 74.2)	(8.7, 72.6)	(5.2, 83.8)	(6.6, 73.9)
(4.9, 83.4)	(5.3, 79.6)	(4.8, 73.0)	(7.2, 71.0)	(9.0, 82.0)	(5.4, 76.3)	(4.1, 83.4)	(4.9, 84.9)	(6.9, 76.0)	(9.3, 75.1)
(6.9, 72.9)	(9.2, 80.1)	(7.5, 75.8)	(6.5, 71.1)	(8.9, 80.2)	(8.2, 82.8)	(6.6, 71.5)	(6.8, 75.9)	(9.3, 75.6)	(7.9, 77.7)
(7.5, 71.3)	(4.3, 80.8)	(9.2, 85.0)	(8.4, 71.6)	(4.1, 76.9)	(8.5, 72.0)	(5.8, 80.6)	(8.5, 74.6)	(4.6, 71.0)	(5.4, 72.6)
(4.9, 81.8)	(5.9, 74.0)	(7.9, 78.4)	(9.0, 72.0)	(4.1, 76.4)	(6.4, 70.9)	(8.8, 82.0)	(5.0, 78.4)	(5.9, 81.6)	(7.8, 84.9)

Fig. 5. The simulation scenario. The parameter pairs (PL, PL_{TH}) are labeled in each subarea.

level of surveillance, respectively. As shown in Fig.5, $PL(i)$ and $PL_{TH}(i)$ are uniformly distributed in $[4dB, 10dB]$ and $[70dB, 85dB]$, respectively. The worst-case coverage was determined by the path loss observed in the vertexes of Voronoi diagrams, and the path loss was calculated according to Section II. C. Note that the *path loss* is defined as the difference between the effective transmitted power and the received power and thus takes on positive values, the lower path loss indicates better coverage in our experiments.

Table 3 shows that the Fuzzy Deployment achieves 5.71dB improvement in the worst-case coverage. Based on the simulations, we can conclude that using fuzzy logic to find the optimal number of nodes in each subarea is very beneficial because the path loss observed at the worst points is reduced, which typically means better coverage for the whole network. It is also worth pointing out that these results are not the best performance achievable by the Fuzzy Deployment, because all the rules used in Fuzz Deployment are all extracted from linguistic information so far. For example, the assignment of the consequent parameters, i.e. $\bar{w}^l$'s, is quite arbitrary and intuitive, which means the choices in Fig.3 are not necessarily the best. It is also possible to extract rules from representative data collected from real scenarios, such training often improves the performance of a fuzzy logic system by orders of magnitude.

Table 3. Worst-case coverage.

Deployment	Worst-case coverage(dB)
uniform	93.68
fuzzy	87.97

V. CONCLUSION AND FUTURE WORK

In this paper, a new fuzzy deployment is presented and compared to traditional ones that assume homogeneous environment. Although the simulation results show a 5.71 dB improvement in the worst-case coverage, the power of the fuzzy deployment is not thoroughly exploited. As shown in previous fuzzy applications, proper training often betters the performance of fuzzy logic systems. Generally, a back propagation training, also referred to as a steepest descent algorithm, can be used to tune all the free parameters enclosed in a fuzzy logic system. However, in this case of the Fuzzy Deployment, since the explicit dependence of the worst-case coverage on the area path loss and the required threshold is unknown, the gradient could be elusive to our knowledge. In this case, a random search algorithm could be used to optimize the coverage provided by the Fuzzy Deployment.

ACKNOWLEDGMENT

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

REFERENCES

REFERENCES

[1] V. Ravelomanana, "Extremal properties of three-dimensional sensor networks with applications," *IEEE Trans. on Mobile Computing*, vol. 3, pp. 246 – 257, July 2004.

[2] P. L. Rice, A. G. Longley, K. A. Norton, and A.P. Barsis, "Transmission loss predictions for tropospheric communication circuits," *NBS Tech Note 101*, 1967.

[3] A. G. Longley and P. L. Rice, "Prediction of tropospheric radio transmission loss over irregular terrain; a computer method," *ESSA Technical Report*, 1968.

[4] R. Edwards and J. Durkin, "Computer prediction of service area for vhf mobile radio networks," in *Proceedings of the IEE*, 1969, vol. 116, pp. 1493–1500.

[5] C. E. Dadson, J. Durkin, and E. Martin, "Computer prediction of field strength in the planning of radio systems," *IEEE Trans. Veh. Technol.*, vol. VT-24, no. 1, pp. 1–7, Feb 1975.

[6] T. Okumur, E. Ohmori, and K. Fukuda, "Field strength and its variability in vhf and uhf land mobile service," *Review Electrical Communication Laboratory*, vol. 16, no. 9-10, pp. 825–873, Sept-Oct 1968.

[7] J. M. Mendel, "Fuzzy logic systems for engineering : A tutorial," in *Proceedings of the IEEE*, March 1995, vol. 83, pp. 345–377.

[8] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems*, Prentice-Hall, Upper Saddle River, NJ, 2001.

[9] E. H. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, no. 12, pp. 1182–1191, 1977.

[10] K. Chakrabarty, S.S. Iyengar, Hairong Qi, and Eungchun Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, pp. 1448 – 1453, Dec 2002.

[11] H. Chen, H. Wu, and N.-F. Tzeng, "Grid-based approach for working node selection in wireless sensor networks," in *IEEE International Conference on Communications*, June 2004, vol. 6, pp. 3673 – 3678.

[12] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM'01)*, April 2001, vol. 3, pp. 1380 - 1387.

[13] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660 - 670, Oct. 2002.

[14] T. S. Rappaport, *Wireless Communications:Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, 2002.

[15] J. Walfisch and H. L. Bertoni, "A theoretical model of uhf propagation in urban environments," *IEEE Trans. Antennas Propagat.*, vol. AP-36, pp. 1788–1796, October 1988.

Clustering with Fuzzy Cluster Radius

Liang Zhao and Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76010, USA
Email: zhao@wcn.uta.edu, liang@uta.edu

***Abstract*—Previous research shows that restraining cluster size helps energy efficiency in sensor networks. However, it is often ignored that the distance estimation in sensor networks is inaccurate enough for fine-grained clustering decision. In this paper, we are concerned with developing a fuzzy cluster size to handle the distance error and non-linearity. A fuzzy logic system is developed to make clustering decision based on the received signal strength. Simulation results show that the proposed Fuzzy Cluster Size scheme can keep the performance near the optimal range when distance estimation is distorted by log-normal shadowing.**

I. Introduction

Recent technological advances have made it possible to develop distributed sensor networks consisting of a large number of low-cost, low-power, and multi-functional sensor nodes that communicate in short distances through wireless links [1]. Example applications of sensor networks include target tracking, scientific exploration, and data acquisition in hazardous environments.

Wireless sensors provide a clear advantage in cost, size, flexibility and distributed intelligence over their wired counterparts. However, the energy constraint remains a major concern for wireless sensor networks [2]–[4]. Clustering has been proposed and heavily studied for improvement in energy efficiency [5], [6]. [7] shows that clustering should done with cluster size constraint, however, this cluster size constraint come in form of fixed cluster radius, which could be insufficient to model the complexity of clustering in sensor networks. Furthermore, the distance estimation in sensor networks is inaccurate enough for fine-grained clustering decision. In this paper, we are concerned with developing a fuzzy cluster size to handle the distance error and non-linearity in the clustering.

The rest of this paper is organized as follows. Section II introduces the preliminaries that our research is based on. The problem with fixed cluster size is discussed in Section III. A Fuzzy Logic System is developed to make the clustering decision in Section IV and simulations are given in Section V. Section VI concludes this paper.

II. Preliminaries

A. Overview of Fuzzy Logic Systems

Figure 1 shows the structure of a fuzzy logic system (FLS) [8]. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets. Consider a p-input 1-output FLS, using singleton fuzzification, *center-of-sets* defuzzification [9] and "IF-THEN" rules of the form [10]

$$R^l : \text{IF } x_1 \text{ is } \mathrm{F}_1^l \text{ and } x_2 \text{ is } \mathrm{F}_2^l \text{ and } \cdots \text{ and } x_p \text{ is } \mathrm{F}_p^l, \text{ THEN } y \text{ is } \mathrm{G}^l.$$

Assuming singleton fuzzification, when an input $\mathbf{x}' = \{x_1', \ldots, x_p'\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{\mathrm{F}_1^l}(x_1') \star \mu_{\mathrm{F}_2^l}(x_2') \star \cdots \star \mu_{\mathrm{F}_p^l}(x_p') = \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i') \quad (1)$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm. There are many kinds of defuzzifiers. In this paper, we focus, for illustrative purposes, on the center-of-sets defuzzifier [9]. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')$, so that

$$y_{cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c_{\mathrm{G}^l} \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')}{\sum_{l=1}^{M} \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')} \quad (2)$$

where M is the number of rules in the FLS.

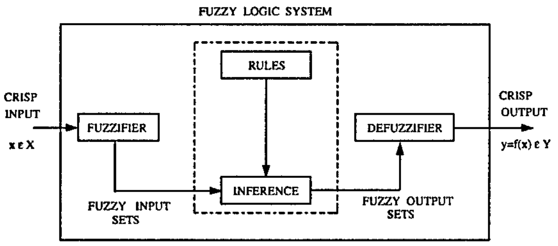


Fig. 1. The structure of a fuzzy logic system.

B. Ranging Techniques

Distance is often estimated based on received signal strength, time of arrival(TOA), time difference of arrival(TDOA) or angle of arrival [11]. The angle-of-arrival based ranging requires directive antennas or arrays, which is not suitable for most microsensors. Similarly, measuring time of flight requires timing device with satisfactory resolution like in GPS. Although TDOA needs much less resolution, it often requires extra acoustic or ultrasound emission, which comes with higher price, larger size and more energy consumption, all seeming impractical for microsensors. Thus, most technically available ranging is based on received signal strength; in fact, RSSI(Received Signal Strength Indication) is widely used in wireless communications to provide distance estimation. The underlying observation is that the average large-scale path loss can be expressed as a function of distance by using a path loss exponent, n [12].

$$\bar{P}L(d) = \bar{P}L(d_0)(\frac{d}{d_0})^n \tag{3}$$

where n is the path loss exponent, which indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance, which is determined from measurement close to the transmitter, and d is the distance from the source to the receiving point. Measurements have also shown that at any value of d, the path loss $PL(d)$ at a particular location is random and distributed log-nomally (normal in dB) about the mean distance-dependent value.

$$PL(d)[dB] = \bar{P}L(d)[dB] + X_\sigma, \tag{4}$$

where X_σ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB). The log-normal shadowing is the main source of distance error for received-signal-strength-based ranging methods. The values of n and σ are often estimated empirically, for example, n could vary from 2 to 10 for different environments, and typical value of σ in urban area is around 10 dBs.

C. Radio Energy Consumption

The following model is adopted from [5] where perfect power control is assumed. To transmit l bits over distance d, the sender's radio expends

$$E_{TX}(l,d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2 & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4 & d \geq d_0 \end{cases} \tag{5}$$

and the receiver's radio expends

$$E_{RX}(l,d) = lE_{elec}. \tag{6}$$

E_{elec} is the unit energy consumed by the electronics to process one bit of message, ϵ_{fs} and ϵ_{mp} are the amplifier factor for free-space and multi-path models, respectively, and d_0 is the reference distance to determine which model to use. The values of these communication energy parameters are set as in Table I.

D. Data Correlation Model

The data collected by neighboring sensors have a lot of redundancy, thus, [5] assumes perfect data correlation that all individual signals from members of the same cluster can be combined into a single representative signal. Nevertheless, this assumption cannot hold when the cluster size increases to some extent. Therefore, we develop a complementary exponential data correlation model based on the observations in distributed data compression [13], [14].

Considering the phenomenon of interest as a random process, the correlation between data collected by two sensors is generally a decreasing function of the distance r between them. After the data aggregation removes most of the redundancy, the residue can be assumed an increasing function of r. Based on the above observation, the data aggregation effect is modeled as below.

Suppose there are M_k non-head members in cluster k ($k = 1,2,3,...,c$), the ith member ($i = 1,2,3,...,M_k$) collects l bits and sends them back to its head k at distance r_{ki}, the head expends $2lE_{DA}$ Joules on the data aggregation of the $2l$ bits (l bits collected by itself and another l bits by its ith member), where E_{DA} is set as $5nJ/bit$ as in [5] and listed in Table I. The resulting data is assumed of $l(1+\eta_{ki})$ bits, where η_{ki} is data aggregation residue ratio and assumed to be complementary exponential, specifically,

$$\eta_{ki} = 1 - e^{-\alpha r_{ki}}, 0 < \alpha < 1, \tag{7}$$

where α is a small positive real number whose magnitude depends on specific phenomenon of interest. For example, the light, acoustic, seismic and thermal signals often show a strong correlation at short distance, and thus, α will have smaller values for such data. Since η is a monotonically decreasing function of α and r, η approaches zero for smaller α and r. This model can approach the perfect-data-correlation assumption in [5] by decreasing α or approach the no-data-aggregation assumption in [15], [16] by increasing α, thus, different scenarios can easily be set up by varying α.

III. Fixed Cluster Size

Expellant Self-Organization (ESO) was proposed to replace the problematic random election in LEACH. ESO used an individual clustering criterion (8) to distribute the clustering decision to each sensor node. That is

$$J_{CM}(i) \underset{CM}{\overset{CH}{\gtrless}} J_{CH}(i), \tag{8}$$

where $J_{CM}(i)$ (and $J_{CH}(i)$) is the energy cost if the ith node chooses to be a cluster member (and cluster head), respectively. If we substitute the data correlation and energy consumption model into this criterion, we obtain

$$\begin{aligned} E_{elec} + \epsilon_{fs}r_{ik}^2 + E_{elec} + E_{DA} + \eta(r_{ik})(E_{elec} + \epsilon_{mp}d_k^4) \\ \underset{CM}{\overset{CH}{\gtrless}} E_{DA} + E_{elec} + \epsilon_{mp}d_i^4. \end{aligned} \tag{9}$$

The non-linearity in (9) makes it difficult to evaluate in real application. Thus, an easier evaluation is needed. Note that r is the dominating factor in this comparison, (9) reduces to

$$\epsilon_{fs}r_{ik}^2 + \eta(r_{ik})(E_{elec} + \epsilon_{mp}d_k^4) \underset{CM}{\overset{CH}{\gtrless}} \epsilon_{mp}d_i^4 - E_{elec}.$$

Suppose there exists a solution for r, which is denoted by $R_c(d, d_{ch})$. Then,

$$r \underset{CM}{\overset{CH}{\gtrless}} R_c(d, d_{ch}) \tag{10}$$

$R_c(d, d_{ch})$ can be determined analytically or empirically. In [7], $R_c(d, d_{ch})$ is simplified into a constant R_c in order to fit in the limited computational capacity of sensors. FLS is especially useful here because it can do non-linear mapping using only linear computations.

The above derivation indicates that the clustering decision is mainly based on the distance, and the distance is often estimated with error. In this paper, we consider the ranging error with received signal strength because it is most widely used. The log-normal shadowing could distort the clustering decision dramatically if it is not taken care of. In the next session, we design a Fuzzy Logic System to address this problem.

IV. Fuzzy Cluster Size

Since the self-organization details are described in [7], we concentrate on the clustering decision making. Consider a node, [i], which is making it clustering decision, i.e., to be a cluster member or a cluster head. Its RSSI meter can give it a RSSI reading from the base station, $RSSI\_d$, and from the neighboring cluster head [k], $RSSI\_r$. The cluster head [k] also has it RSSI reading from the base station, $RSSI\_d\_ch$, which is available to node [i] through local broadcast. Based on these three parameters, the FLS gives out a resulting ''support" from this node to the neighboring cluster head, which indicate the degree to which this node should join the neighboring cluster head as a cluster member. If the support is above the threshold zero, then this node should join the neighboring cluster head; otherwise, it should claim itself as a cluster head. The whole process is depicted in Fig.2.

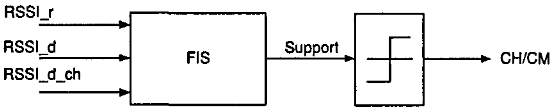


Fig. 2. System diagram.

The rules are designed such as:

$$R^l : \text{IF } RSSI\_r \text{ is } Low \text{ and } RSSI\_d \text{ is } High \text{ and } RSSI\_d\_ch \text{ is } Medium \text{ THEN } support \text{ is } \bar{w}^l.$$

where $\bar{w}^l$ is a real number.

Three Gaussian membership functions are used for each antecedent, and a constant $\bar{w}^l$ is assigned to each rule. The Gaussian membership function is given by

$$\mu_A(x) = exp\{-\frac{1}{2}\frac{(x - m_A)^2}{\sigma_A^2}\}, \quad (11)$$

where m_A and σ_A are the mean and standard deviation, respectively. Note that there are two free parameters for each Gaussian membership function and there are $M = 3^3 = 27$ rules, there are $2 \times 3 \times 3 = 18$ antecedent parameters and 27 consequent parameters, for a total of 45 parameters. These parameters need to be tuned using a set of training data. Another set of data, called checking data, is often used in training to prevent overfitting, which can be observed when checking error begins to increase while the training error is still decreasing. The antecedent parameters m_A's and σ_A's are tuned using back-propagation while the consequent parameters $\bar{(w)}^l$'s are determined with Least-Square method [17].

The training and checking data are collected by evaluating (9) at difference step size. The desired support is defined as the difference between J_{CM} and J_{CH} so that the support is positive when $J_{CM} > J_{CH}$. The initial membership functions are equally space on the input space, for example, the initial membership functions for the first input, $RSSI\_r$, are depicted in Fig.3. The consequent weight $\bar{w}^l$ are randomized. And the output surface of the trained FLS is plotted in Fig.4.

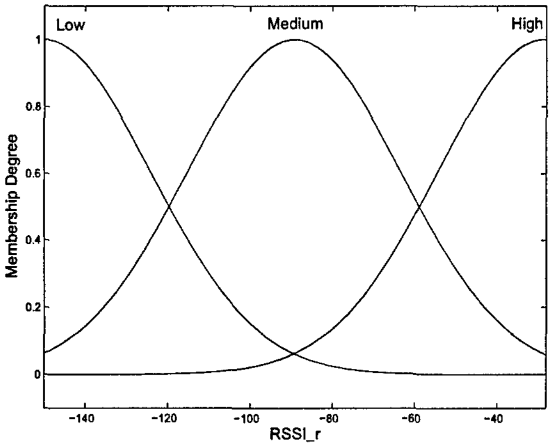


Fig. 3. Example: initial membership functions for $RSSI\_r$.

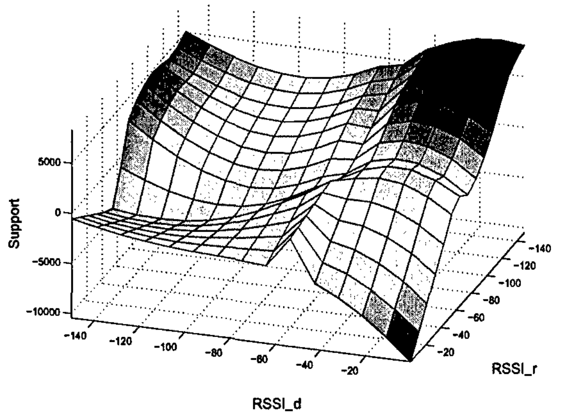


Fig. 4. Output surface of the trained FLS. $RSSI\_d\_ch$ fixed at 70.

The $support$ is used in two ways. Firstly, when a node is to make clustering decision, that is, to be a cluster head or cluster member, it should count the $support$ from its potential supporters, i.e., those nodes who would support this node with positive $support$ if this node choose to be a cluster head. The sum of its $support$ and its energy level are the basis of this node's clustering decision. Secondly, after some node successfully becomes cluster head, the other nodes should consider joining this cluster head. They would do so only when they find themselves supporting this cluster head with positive *support*. Those with negative *support* to all existing cluster heads should consider themselves ''unclustered", and try to organize themselves into clusters through ESO.

V. Simulations

We compared the performance of clustering with fuzzy and fixed cluster size using computer simulations. 100 nodes with 2J initial energy were evenly distributed in a circular region with diameter $100m$, and the base station was located at $(125m, 0)$. The standard deviation of log-normal shadowing is set at 11.8dB. The exponent α of data correlation model is set at 0.001. The communication Energy Parameters are set as in Table tab:Pm. We ran simulations over 1000 random network topologies and took average of collected data.

TABLE I

Communication Energy Parameters

Name	Value
d_0	86.2m
E_{elec}	$50nJ/bit$
E_{DA}	$5nJ/bit$
ϵ_{fs}	$10pJ/bit/m^2$
ϵ_{mp}	$0.0013pJ/bit/m^4$

In Fig.5(a), (b), (c) and (d), the amount of data received at the base station over time, the amount of data received at the base station per given amount of energy, the number of survival nodes over time and the umber of survival nodes per amount of data received in the base station are plotted respectively for $R_c = 10, 30, 40, 80(m)$. Fig.5(a) and (c) show that the network lifetime is maximized at $R_c = 40m$, (a) and (c) also show that the amount of data delivered is maximized at $R_c = 40m$. The Data/Energy Ratio, indicated by the slop in (b), also reaches its maximum at $R_c = 40m$. However, if R_c is not set at this optimal values, the performance degrades dramatically.

The performance of Fuzzy Cluster Size is plotted in Fig.6 and compared with fixed $R_c = 40m$. The two curves in all four subfigures are very close to each other, which clearly show that fuzzy cluster size could always keep the performance near the optimal size. When confronted with distance error, this feature can guarantee robust results.

VI. Conclusion

In this paper, we propose using fuzzy cluster size to address the non-linearity and distance uncertainty in clustering. Thanks to Fuzzy Logic System's power in handling non-linearity and uncertainty, the Fuzzy Cluster Size scheme keeps the clustering performance near the optimal range when the distance estimation is distorted by log-normal shadowing.

Acknowledgment

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 20, pp. 102–114, Aug. 2002.

[2] A. Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 48 – 59, Aug 2002.

[3] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40 – 50, March 2002.

[4] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 28 – 39, Aug. 2002.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660 – 670, Oct. 2002.

[6] J.-S. Liu and C.-H. Lin, "Power-efficiency clustering method with power-limit constraint for sensor networks," in *Proc. of the 2003 IEEE International Conference on Performance, Computing, and Communications*, Apr 2003, pp. 129 –136.

[7] L. Zhao, X. Hong, and Q. Liang, "Energy-efficient self-organization for wireless sensor networks: A fully distributed approach," in *IEEE Globecom'04*. Dallas, TX: IEEE, Dec 2004.

[8] J. M. Mendel, "Fuzzy logic systems for engineering : A tutorial," in *Proceedings of the IEEE*, vol. 83, no. 3, March 1995, pp. 345–377.

[9] ——, *Uncertain Rule-Based Fuzzy Logic Systems*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[10] E. H. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, no. 12, pp. 1182–1191, 1977.

[11] J. Hightower and G. Borriello, "Location sensing techniques," University of Washington, Department of Computer Science and Engineering, Seattle, WA, UW CSE 01-07-01, July 2001.

[12] T. S. Rappaport, *Wireless Communications:Principles and Practice*. Upper Saddle River, NJ: Prentice-Hall, 2002.

[13] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 51–60, Mar 2002.

[14] A. Boulis, S. Ganeriwal, and M. Srivastava, "Aggregation in sensor networks: an energy-accuracy trade-off," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003, pp. 128–138.

[15] T. Shepard, "A channel access scheme for large dense packet radio networks," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1996, pp. 219–230.

[16] M. Ettus, "System capacity, latency and power consumption in multihop-routed ss-cdma wireless networks," in *Proc. Radio and Wireless Conf. (RAWCON'98)*, Colorado Springs, CO, Aug. 1998, pp. 55–58.

[17] J.-S. R. Jang, "Anfis: Adaptive-network-based fuzzy inference systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, May 1993.

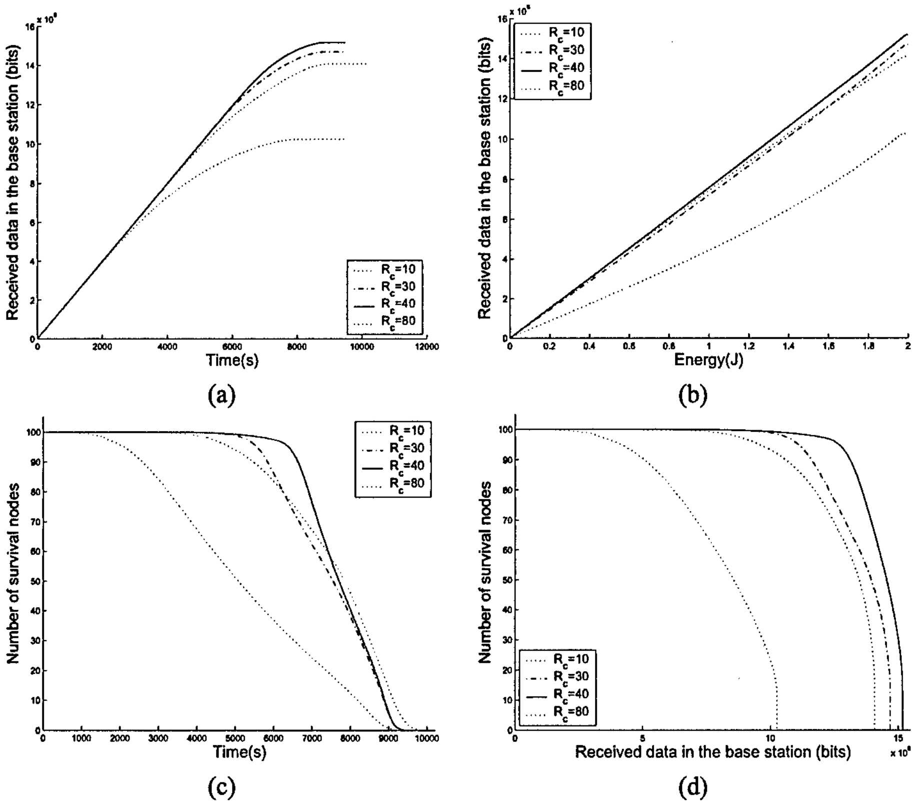
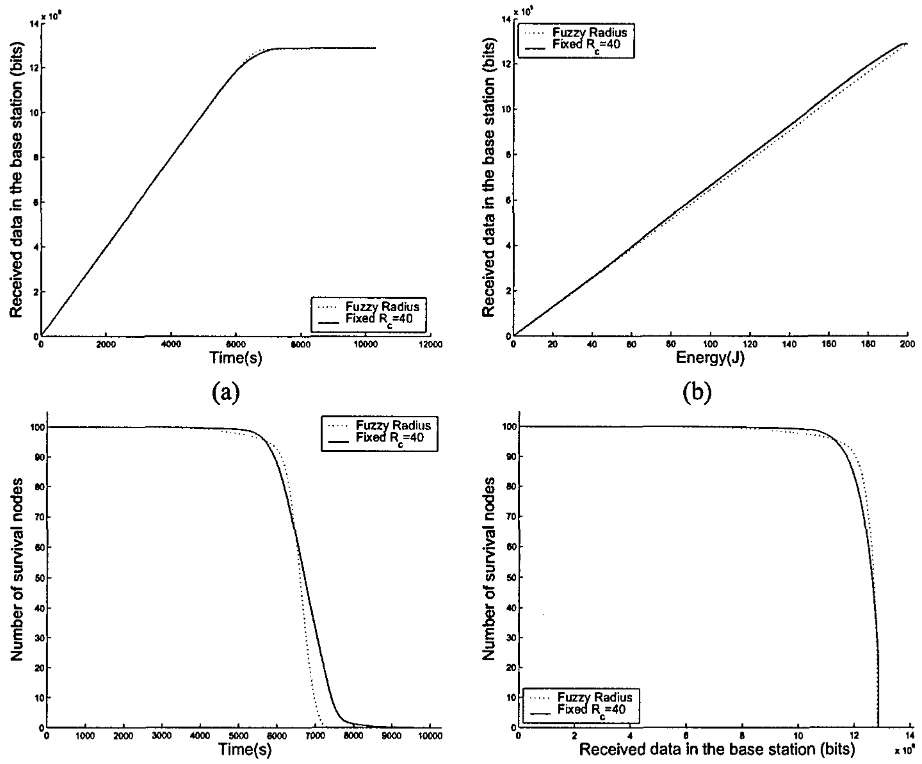


Fig. 5. Performance of clustering at different fixed R_C. (a) Amount of data received at the base station over time. (b) Amount of data received at the base station per given amount of energy. (c) Number of survival nodes over time. (d) Number of survival nodes per amount of data received in the base station.



(a)

(b)

(c)

(d)

Fig. 6. Performance of clustering at different fixed R_c. (a) Amount of data received at the base station over time. (b) Amount of data received at the base station per given amount of energy. (c) Number of survival nodes over time. (d) Number of survival nodes per amount of data received in the base station.

Bottom-up Cross-Layer Optimization for Mobile Ad Hoc Networks

Xinsheng Xia and Qilian Liang
Department of Electrical Engineering
The University of Texas at Arlington
416 Yates Street, Rm 518
Arlington, TX 76010-0016
Email: xia@wcn.uta.edu, liang@uta.edu

Abstract—**In this paper, we introduce a new method for cross-layer design in mobile ad hoc networks. We use fuzzy logic system (FLS) to coordinate physical layer, data-link layer and application layer for cross-layer design. Ground speed, average delay and packets successful transmission ratio are selected as antecedents for the FLS. The output of FLS provides adjusting factors for the AMC (Adaptive Modulation and Coding), transmission power, retransmission times and rate control decision. Simulation results show that our cross-layer design can reduce the average delay, increase the throughput and extend the network lifetime. The network performance parameters could also keep stable after the cross-layer optimization.**

I. Introduction

The demand for energy efficiency and Quality of Service (QoS) in mobile ad hoc networks is growing in a rapid speed. To enhance the energy efficiency and QoS, we consider the combination of physical layer, data-link layer and application layer together, a cross-layer approach. A strict layered design is not flexible enough to cope with the dynamics of the mobile ad hoc networks [1]. Cross-layer design could introduce the layer interdependencies to optimized overall network performance. The general methodology of cross-layer design is to maintain the layered architecture, capture the important information that influence other layers, exchange the information between layers and implement adaptive protocols and algorithms at each layer to optimize the performance.

Lots of previous works have focused on cross-layer design for QoS provision. Liu [2] combine the AMC at physical layer and ARQ at the data link layer. Ahn [3] use the info from MAC layer to do rate control at network layer for supporting real-time and best effort traffic. Akan [4] propose a new adaptive transport layer suite including adaptive transport protocol and adaptive rate control protocol based on the lower layer information.

Some works related to energy efficiency have been reported. Banbos proposes a power-controlled multiple access schemes in [5]. This protocol reveals the trade-off of the transmitter power cost and backlog/delay cost in power control schemes. Zhu [6] proposes a minimum energy routing scheme, which consider the energy consumption for data packets as well as control packets of routing and multiple access. In [7], Sichitiu proposes a cross-layer scheduling method. Through combining network layer and MAC layer, a deterministic, schedule-based energy conservation scheme is proposed. This scheme drives its power efficiency from eliminating idle listening and collisions.

However, cross-layer design can produce unintended interactions among protocols, such as an adaptation loops. It is hard to characterize the interaction at different layers and joint optimization across layers may lead to complex algorithm.

Our algorithm is quite different from all the previous works. We propose to use the Fuzzy Logic System (FLS) in the cross-layer design. We define a coherent time, a certain period of time. During this coherent time, the AMC (Adaptive Modulation and Coding), transmission power, retransmission times and rate control decision are used for packet transmission. After this time, we adaptively adjust these parameters by FLS again basing on current ground speed, average delay and the packets successful transmission ratio. By applying the FLS mechanism to the cross-layer, a better QOS provision and energy efficiency are achieved.

The remainder of this paper is structured as following. In section II, we introduce the preliminaries. In section III, we make a overview of fuzzy logic systems. In section IV, we apply the FLS into the cross-layer design. Simulation results and discussions are presented in section V. In section VI, we conclude the paper.

II. Preliminaries

A. IEEE 802.11a OFDM PHY

The physical layer is the interface between the wireless medium and the MAC [8]. The principle of OFDM is to divide a high-speed binary signal to be transmitted over a number of low data-rate subcarriers. A key feature of the IEEE 802.11a PHY is to provide 8 PHY modes with different modulation schemes and coding rates, making the idea of link adaptation feasible and important, as listed in Table I. BPSK, QPSK, 16-QAM and 64-QAM are the supported modulation schemes. The OFDM provides a data transmission rates from 6 to 54MBPS. The higher code rates of 2/3 and 3/4 are obtained by puncturing the original rate 1/2 code.

TABLE I

Eight PHY Modes of the IEEE802.11A PHY

$Mode$	$Modulation$	$CodeRate$	$DataRate$	Bps
1	$BPSK$	1/2	$6Mbps$	3
2	$BPSK$	3/4	$9Mbps$	4.5
3	$QPSK$	1/2	$12Mbps$	6
4	$QPSK$	3/4	$18Mbps$	9
5	$16-QAM$	1/2	$24Mbps$	12
6	$16-QAM$	3/4	$36Mbps$	18
7	$64-QAM$	2/3	$48Mbps$	24
8	$64-QAM$	3/4	$54Mbps$	27

B. IEEE 802.11 MAC

The 802.11 MAC uses Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) to achieve automatic medium sharing between compatible stations. In CSMA/CA, a station senses the wireless medium to determine if it is idle before it starts transmission. If the medium appears to be idle, the transmission may proceed, else the station will wait until the end of the in-progress transmission. A station will ensure that the medium has been idle for the specified inter-frame interval before attempting to transmit.

Besides carrier sense and RTS/CTS mechanism, an acknowledgment (ACK) frame will be sent by the receiver upon successful reception of a data frame. Only after receiving an ACK frame correctly, the transmitter assumes successful delivery of the corresponding data frame. The sequence for a data transmission is: RTS-CTS-DATA-ACK.

A mobile node will retransmit the data packet when finding failing transmission. Retransmission of a signal packet can achieve a certain probability of delivery. There is a relationship between the probability of delivery p and retransmission times n:

$$n = 1.451n\frac{1}{1-p} \tag{1}$$

The IEEE 802.11 standard requires that the transmitter's MAC discard a data frame after certain number of unsuccessful transmission attempts. According to the requirement of probability of delivery, we choose the minimum number of retransmission. The advantage is we can save energy through avoiding unnecessary retransmission, and ensure probability of delivery.

C. Application Layer

Traffic in application layer is divides into two classes: real-time and best-effort. Each node in the mobile ad hoc networks independently regulates best effort traffic. It is proposed to control the rate of the best-effort traffic to avoid excessive delays of the real-time traffic by using local per-hop delays as a feedback to local rate controller [3]. The general behavior of a congestion-controlled system is illustrated in Fig.1. The control algorithm ensures that the system operates around, or preferably close to the "cliff", which ensure maximum system throughput, but at the cost of large average packets delay. The control algorithm discussed, one the other hand, keep the system at the delay "knee" where the system throughput is almost the same as the at the cliff, but the buffers are significantly less loaded, so the delay is close the minimum. Due to loss typically happens at the cliff, while delays start to increase at the knee, we use the per-hop MAC delay as a feedback for local control instead of the packet loss.

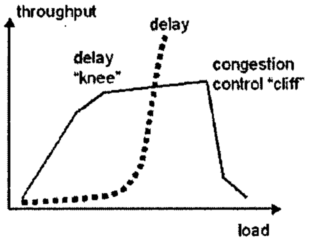


Fig. 1. General Behavior of a Congestion-controlled System

When MAC layer acquires access to the channel, the nodes will exchange the RTS-CTS-DATA-ACK packets. After the transmitters receive an ACK packet, a packet is transmitted successfully. The packet delay represents the time it took to send the packet between the transmitter and the next-hop receiver, including the deferred time and the time to fully acknowledge the packet. In this paper, we assume that there will be always best-effort traffic present that can be locally and rapidly rate controlled in an independent manner at each node to yield necessary low delays and stable throughputs.

D. Energy

A mobile node consumes significant energy when it transmits or receives a packet. But we will not consider the energy consumed when the mobile node is idle.

The distance between two nodes are variable in the mobile ad hoc networks and the power loss model is used. To send the packet, the sender consumes [9],

$$P_{tx} = P_{elec} + \epsilon_{fs} \cdot d^2 \tag{2}$$

and to receive the packet, the receiver consumes,

$$P_{rx} = P_{elec} \tag{3}$$

where P_{elec} represents the power that is necessary for digital processing, modulation, and ϵ_{fs} represents the power dissipated in the amplifier for the free space distance d transmission.

A joint characteristic of most application scenarios of mobile ad hoc networks is that mobile nodes only have a limited energy supply which might not even be rechargeable, hence they have to be energy-efficient as possible. Transmitter power control allows interfering communication links sharing the same channel to achieve their required QoS levels, minimizing the needed power, mitigating the channel interference, and maximizing the network user/link capacity.

E. Delay

The packet transmission delay between the mobile nodes includes three parts: the wireless channel transmission delay, the Physical/MAC layer transmission delay, and the queuing delay [10].

Defining D as the distance between two nodes and C as the light speed, the wireless channel transmission delay as:

$$Delay_{ch} = \frac{D}{C} \tag{4}$$

The Physical/MAC layer transmission delay will be decided by interaction of the transmitter and the receive channel, the node density and the node traffic intensity etc.

The queuing delay is decided by the mobile node I/O system-processing rate, the subqueue length in the node.

In order to make the system "stable", the rate at which node transfers packets intended for its destination must satisfy all nodes that the queuing lengths will not be infinite and the average delays will be bounded.

F. Node Mobility and Channel Fading

Mobility of a mobile node generates a doppler shift, which is a key parameter of fading channel. The doppler shift is

$$f_d = \frac{v}{c} f_c \tag{5}$$

where v is the ground speed of a mobile node, c is the speed of light ($3 \times 10^8 m/s$), and f_c is the carrier. In our simulation, we used the carrier is $6GHz$. For reference, if a node moves with speed $10m/s$, the doppler shift is $200Hz$.

We model channel fading in ad hoc networks as Rician fading. Rician fading occurs when there is a strong specular (direct path or line of sight component) signal in addition to the scatter (multipath) components. For example, in communication between two infraed sensors, there exist a direct path. The channel gain,

$$g(t) = g_I(t) + jg_Q(t) \tag{6}$$

can be treated as a wide-sense stationary complex Gaussian random process, and $g_I(t)$ and $g_Q(t)$ are Gaussian random processes with non-zero means $m_I(t)$ and $m_Q(t)$, respectively; and they have same variance σ_g^2, then the magnitude of the received complex envelop has a Rician distribution,

$$p_\alpha(x) = \frac{x}{\sigma^2} \exp\{-\frac{x^2 + s^2}{2\sigma^2}\} I_0(\frac{xs}{\sigma^2}) \quad x \geq 0 \tag{7}$$

where

$$s^2 = m_I^2(t) + m_Q^2(t) \tag{8}$$

and $I_0(\cdot)$ is the zero order modified Bessel function. This kind of channel is known as Rician fading channel. A Rician channel is characterized by two parameters, Rician factor K which is the ratio of the direct path power to that of the multipath, i.e., $K = s^2/2\sigma^2$, and the Doppler spread (or single-sided fading bandwidth) f_d. We simulate the Rician fading using a direct path added by a Rayleigh fading generator. The Rayleigh fade generator is based on Jakes' model [11] in which an ensemble of sinusoidal waveforms are added together to simulate the coherent sum of scattered rays with Doppler spread f_d arriving from different directions to the receiver. The amplitude of the Rayleigh fade generator is controlled by the Rician factor K.

BPSK, QPSK, 16-QAM and 64-QAM are the supported modulation schemes for IEEE 802.11a OFDM physical layer. We can show their performance curves with Rician fading in Fig. 2.

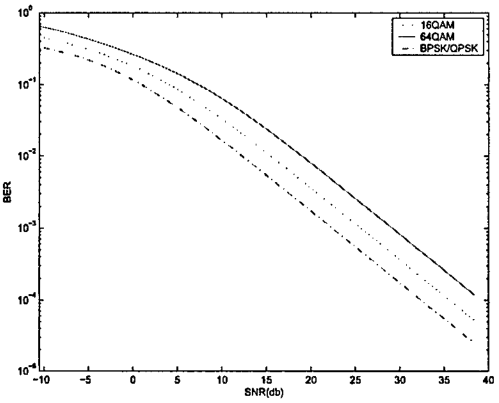


Fig. 2. Modulation Curves with Rician Fading

After we introduce the channel coding and node mobility into the modulation schemes, the mudualtion curves will change a lot. For the same SNR, channel coding will improve the BER performance and the mobility will degrade the BER performance.

G. One-step Markov Path Model

The mobile nodes are roaming independently with variable ground speed. The mobility model is called one-step Markov path model [12]. The probability of moving in the same direction as the previous move is higher than other directions in this model, which means this model has memory. Fig.3 shows the probability of the six directions.

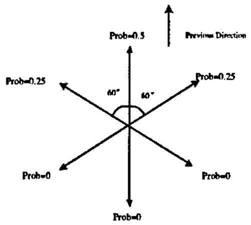


Fig. 3. One-step Markov Path Model

III. Overview of Fuzzy Logic Systems

Figure 4 shows the structure of a fuzzy logic system (FLS).

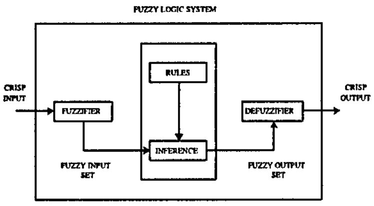


Fig. 4. The structure of a fuzzy logic system

When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets [13]. Consider a p-input 1-output FLS, using singleton fuzzification, *center-of-sets* defuzzification [14] and "IF-THEN" rules of the form [15]

$$R^l : \text{IF } x_1 \text{ is } \mathrm{F}_1^l \text{ and } x_2 \text{ is } \mathrm{F}_2^l \text{ and } \cdots \text{ and } x_p \text{ is } \mathrm{F}_p^l, \\ \text{THEN } y \text{ is } \mathrm{G}^l.$$

Assuming singleton fuzzification, when an input $\mathbf{x}' = \{x_1', \ldots, x_p'\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{\mathrm{F}_1^l}(x_1') \star \mu_{\mathrm{F}_2^l}(x_2') \star \cdots \star \mu_{\mathrm{F}_p^l}(x_p') = \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i') \quad (9)$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm. There are many kinds of defuzzifiers. In this paper, we focus, for illustrative purposes, on the center-of-sets defuzzifier. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')$, so that

$$y_{cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c_{\mathrm{G}^l} \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')}{\sum_{l=1}^{M} \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x_i')} \quad (10)$$

where M is the number of rules in the FLS.

IV. Fuzzy Application For Cross-layer Design

AMC, transmission power, retransmission times and rate control decision will manage the energy consumption and QoS provision. How to choose a proper adjusting factor for these parameters will determine the wireless ad hoc networks performance.

We collect the knowledge for adjusting factor selection based on the following three antecedents:

1) Antecedent 1. Ground speed.
2) Antecedent 2. Average delay.
3) Antecedent 2. Packets successful transmission ratio.

The linguistic variables used to represent the Ground speed, average delay and packets successful transmission ratio were divided into three levels: *low*, *moderate*, and *high*. The consequents – the adjusting factor for the AMC, transmission power, retransmission times and rate control decision were divided into 9 levels, *decrease one, decrease two, decrease three, decrease four, unchanged, increase one, increase two, increase three and increase four*. Fig.5 show the FLS application for the cross-layer design.

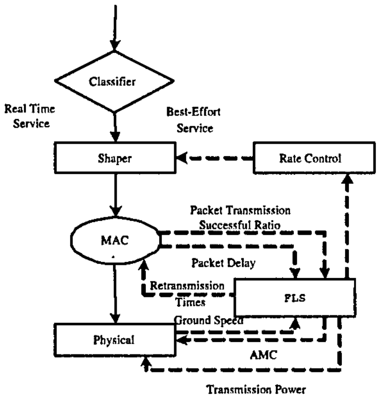


Fig. 5. Cross-layer Design Algorithm

We designed questions such as:

IF *ground speed* is *low*, *average delay* is *low* and *packets successful transmission ratio* is *high*, THEN the adjusting factor is ______________.

So we need to set up $3^3 = 27$ (because every antecedent has 3 fuzzy sub-sets, and there are 3 antecedents) rules for this FLS. We summarized these rules in Table II.

We used trapezoidal membership functions (MFs) to represent *low*, *high*, *increase four* and *decrease four*; and triangle MFs to represent *moderate*, *unchange*, *increase one increase two*, *increase three*, *decrease one*, *decrease two* and *decrease three*. We show these MFs in Fig.6 and Fig.7.

In our approach to form a rule base, we chose a single consequent for each rule. We design a fuzzy logic system using rules such as:

R^l : IF *ground speed* (x_1) is F_l^1,*average delay* (x_2) is F_l^2, and *packets successful transmission ratio* (x_3) is F_l^3, THEN the adjusting factor (y) is c^l.

For every input (x_1, x_2, x_3), the output is computed

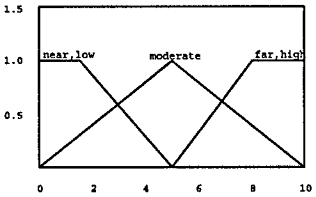


Fig. 6. MFs for antecedents

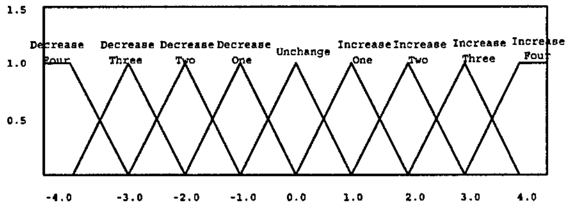


Fig. 7. MFs for conseqents

using

$$y(x_1, x_2, x_3) = \frac{\sum_{l=1}^{27} \mu_{\mathrm{F}_l^1}(x_1)\mu_{\mathrm{F}_l^2}(x_2)\mu_{\mathrm{F}_l^3}(x_3)c_G^l}{\sum_{l=1}^{27} \mu_{\mathrm{F}_l^1}(x_1)\mu_{\mathrm{F}_l^2}(x_2)\mu_{\mathrm{F}_l^3}(x_3)} \quad (11)$$

We apply (11) to compute the adjusting factors and adjust the network parameters dynamically. Comparing to the constant parameters, the fuzzy optimization for cross-layer design can meet QoS and energy requirement.

V. SIMULATIONS

We implemented the simulation model using the OPNET modeler. The simulation region is 300×300 meters. There were 12 mobile nodes in the simulation model, and the nodes were roaming independently with variable ground speed between 0 to 10 meters per second. The mobility model was called one-step Markov path model. The movement would change the distance between mobile nodes.

1) Average Delay: Because data communications in the mobile networks had trimming constraints, it was important to design the network algorithm to meet a kind of end-end deadline [16]. We used the average delay to evaluate the network performance.

$$d_{average} = \frac{\sum_{i=1}^{k} d_i}{k} \quad (12)$$

Each packet was labeled a timestamp when the source mobile node generated it. When its destination mobile node received it, the time interval was the transmission delay.

Fig.8 showed the delay performance of the constant parameters and the one after cross-layer optimization for the real time traffic, the best effort traffic and all the traffic. Cross-layer optimization made a tradeoff for the average delay between the real time traffic and the best effort traffic. For the real time traffic, the cross-layer optimization would enlarge about 0.6 seconds. However for the best effort case, the cross-layer optimization could reduce the delay by up to 90.53%. For the all traffic, the cross-layer optimization could reduce the delay by up to 71.85%, which meant the cross-layer optimization could improve the average delay performance for the whole system. As showed in the best effort case, the cross-layer optimization could make the average delay "stable", which was important for the communication system design.

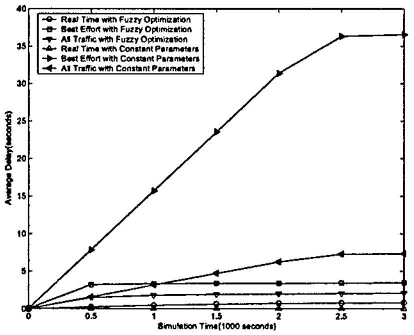


Fig. 8. Average Delay

2) Energy Efficiency: It was not convenient to recharge the battery, so the energy efficiency was extremely important for mobile ad hoc networks. The network should keep an enough number of "live" mobile nodes to collect data, that meant the network need to keep the energy among the mobile nodes in balance. We used the number of remaining alive nodes as the parameter of the energy efficiency.

In (2) and (3), we assumed P_{elec} was equal to 6.0×10^{-4} and ϵ_{fs} was equal to 6.0×10^{-4}. We assumed that the energy of each mobile node was 0.07 J.

When the remaining energy of a mobile node was lower than a certain threshold, the node was considered as "dead". In this simulation, we chose 1.2×10^{-3} as the threshold. A sensor was "dead" meant it could not transmit/receive packets any longer, so it would be ignored by network. The number of nodes of mobile ad hoc networks which was below a certain threshold meant this network does not work.

As Fig.9 showed, after fuzzy optimization, the duration of the first node "dead" is 1.67 times longer than that of the constant parameters, which is 1589 seconds.

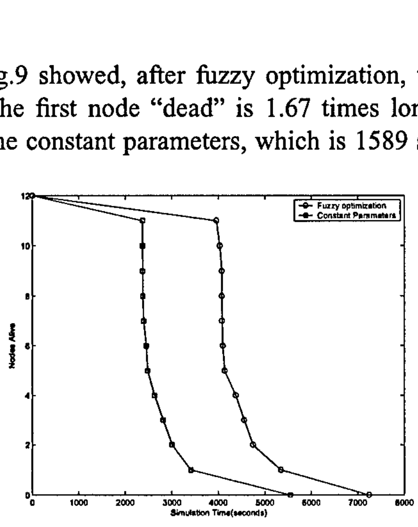


Fig. 9. Node Alive

3) Networks Efficiency: The mobile ad hoc networks were used to collect data and transfer packets. The throughput of packets transmitted was one of the parameters to evaluate the networks efficiency. In our simulation, we assumed the collecting data distribution of the mobile node was Poisson distribution and the arriving interval was 0.2 second. Observing from Fig.10, the cross-layer optimization made a tradeoff between the real time traffic and the best effort traffic. For the real time traffic, after the cross-layer optimization, the throughput of the network was about 0.02% smaller than that of the constant parameters. However, for the best effort traffic, the throughput of the network was up to 71.99% larger. For the all the traffic case, after the cross-layer optimization, the throughput of the network was up to 32.52% larger, which meant the cross-layer optimization could improve the throughput performance for the whole system. As the performance of the average delay, the cross-layer optimization could achieve a "stable" throughput performance.

We introduced the fuzzy logic system in the cross-layer design. Comparing with other algorithms for cross-layer design, the fuzzy method could be flexible and simpler to implement and the performance outputs were also impressive.

VI. Conclusion

Cross-layer design is a effective method to improve the performance of the mobile ad hoc network. We apply the fuzzy logic system to combine physical layer, data-link layer and application layer together. We selected ground speed, average delay and packets transmission successful ratio as antecedents. The output of FLS

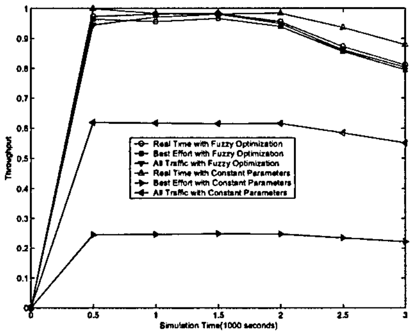


Fig. 10. Throughput

provides adjusting factors for the AMC, transmission power, retransmission times and rate control decision. Simulation shows the FLS application in cross-lay design could reduce the average delay, increase the throughput and extend the network lifetime. After the cross-layer optimization, the network performance parameters could also keep stable. In the future, we can consider other layers, such as network layer for the cross-layer design.

ACKNOWLEDGMENT

This work was supported by the U.S. Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466.

REFERENCES

[1] A.J. Goldsmith and S.B. Wicker; " Design Challenges for Energy-Constrained Ad Hoc Wireless Networks, ," *IEEE Wireless Comm.,*, vol. 9, No. 4 ,2002 , pp. 8-27,

[2] Q.Liu, S.Zhou and G.Giannakis; " Cross-Layer Combining of Adaptive Modulation and Coding with Truncated ARQ over Wireless Links ," *Wireless Communications, IEEE Transactions on*, Volume: 3 , Issue: 5 , Sept. 2004 Pages:1746 - 1755,

[3] G. Ahn, A. Campbell, A. Veres and L. Sun; " Support Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN),", *Mobile Computing, IEEE Transactions on*, Volume: 1 , Issue: 3 , July-Sept. 2002 Pages:192 - 207,

[4] O.B.Akan and I.F.Akyildiz; " ATL, An Adaptive Transport Layer Suite for Next Generation on Wireless Internet ,", *IEEE Journal on Selected Areas in Communcations*, June 2004,

[5] N.Bambos, S.Kandukuri " Power-Controlled Multiple Access Schemes for Next-Generation Wireless Packet Networks ,", *IEEE Wireless Communications*, June 2002,

[6] J.Zhu, C.Qiao and X.Wang " A Comprehensive Minimum Energy Routing Scheme for Wireless Ad Hoc Networks ,", *IEEE INFOCOM2004*,

[7] M.L.Sichitiu " Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks ,", *IEEE INFOCOM2004*,

[8] D.Qiao, S. Choi, and K.G. Shin " IEEE Trans. On Mobile Computing ,", *IEEE Trans. On Mobile Computing*, Oct. 2002,

[9] Heinzelman, W.B.; Chandrakasan, A.P.; Balakrishnan, H.; " An application-specific protocol architecture for wireless microsensor networks ," *IEEE Transactions on Wireless Communications*, Volume: 1 Issue: 4, Oct 2002

[10] Xia, X.;Liang, Q.; "Latency-aware and energy efficiency tradeoffs for sensor networks" *Accepted by Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE* ,

[11] G.L. Stuber; "Principles of Mobile Communcation" *Kluwer Academic Press*, 2001,

[12] Hou T. C., and Tsai T. J.; "Adaptive clustering in a hierarchical ad hoc network" *Proc. Int. Computer Symp.,Tainan, Taiwan,R.O.C.*, Dec.1998, pp.171-176.,

[13] Liang,Q.; "Clusterhead election for mobile ad hoc wireless network" *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE*, Proceedings on ,Volume: 2 , Sept. 7-10, 2003 Pages:1623 - 1628,

[14] Mendel,J. M.; *Uncertain Rule-Based Fuzzy Logic Systems*, Prentice-Hall, Upper Saddle River, NJ, 2001.

[15] Mamdani,E. H.; " Applications of fuzzy logic to approximate reasoning using linguistic systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 12, pp. 1182-1191, 1977.,

[16] Lu, C. et al; " RAP: a real-time communication architecture for large-scale wireless sensor networks ," *Proceeding of the eighth IEEE real-time and embedded technology and applications Symposium*, September 25 - 27, 2002,San Jose, California, Pages:55 - 66

TABLE II

THE FUZZY RULES FOR CROSS-LAYER DESIGN

Antecedent 1 is *its ground speed*, Antecedent 2 is *its average delay* and Antecedent 3 is *its packets successful transmission ratio.* Consequent 1 is *adjusting factor for retransmission times*, Consequent 2 is *adjusting factor for AMC*, Consequent 3 is *adjusting factor for transmission power* and Consequent 4 is *adjusting factor for rate control decision.*

Rule #	Antecedent 1	Antecedent 2	Antecedent 3	Consequent 1	Consequent 2	Consequent 3	Consequent 4
1	low	low	low	increase two	decrease two	unchange	unchanged
2	low	low	moderate	unchanged	unchanged	decrease two	decrease two
3	low	low	high	decrease two	increase two	decrease four	decrease four
4	low	moderate	low	increase one	decrease one	increase one	increase one
5	low	moderate	moderate	decrease one	increase one	decrease one	decrease one
6	low	moderate	high	decrease three	increase three	decrease three	decrease three
7	low	high	low	unchanged	unchanged	increase two	increase two
8	low	high	moderate	decrease two	increase two	unchanged	unchanged
9	low	high	high	decrease four	increase four	decrease two	decrease two
10	moderate	low	low	increase three	decrease three	increase one	increase one
11	moderate	low	moderate	increase one	decrease one	decrease one	decrease one
12	moderate	low	high	decrease one	increase one	decrease three	decrease three
13	moderate	moderate	low	increase two	decrease two	increase two	increase two
14	moderate	moderate	moderate	unchanged	unchanged	unchanged	unchanged
15	moderate	moderate	high	decrease two	increase two	decrease two	decrease two
16	moderate	high	low	increase one	decrease one	increase three	increase three
17	moderate	high	moderate	decrease one	increase one	increase one	increase one
18	moderate	high	high	decrease three	increase three	decrease one	decrease one
19	high	low	low	increase two	decrease four	increase two	increase two
20	high	low	moderate	increase two	decrease two	unchanged	unchanged
21	high	low	high	unchanged	unchanged	decrease two	decrease two
22	high	moderate	low	increase three	decrease three	increase three	increase three
23	high	moderate	moderate	increase one	decrease one	increase one	increase one
24	high	moderate	high	decrease one	increase one	decrease one	decrease one
25	high	high	low	increase two	decrease two	increase four	increase four
26	high	high	moderate	unchanged	unchanged	increase two	increase two
27	high	high	high	decrease two	increase two	unchanged	unchanged