ISSN: 1945-3078 (Print), 1945-3086 (Online) Volume 2, Number 2, February 2010

Wireless Sensor Network

Chief Editor : Kosai Raoof

www.scirp.org/journal/wsn/

Scientific
Research

Journal Editorial Board

ISSN 1945-3078 (Print)   ISSN 1945-3086 (Online)

http://www.scirp.org/journal/wsn/

***Wireless Sensor Network*, 2010, 2, 93-196**
Published Online February 2010 in SciRes (http://www.SciRP.org/journal/wsn/).

TABLE OF CONTENTS

Volume 2 Number 2 February 2010

Evaluation of Multiusers' Interference on Radiolocation in CDMA Cellular Networks
A. J. Bamisaye, M. O. Kolawole, V. S. A. Adeloye……93

Energy Harvesting Strategy Using Piezoelectric Element Driven by Vibration Method
D.-G. Kim, S.-N. Yun, Y.-B. Ham, J.-H. Park……100

Linear Pulse-Coupled Oscillators Model—A New Approach for Time Synchronization in Wireless Sensor Networks
Z. L. An, H. S. Zhu, M. L. Zhang, C. N. Xu, Y. J. Xu, X. W. Li……108

K-Nearest Neighbor Based Missing Data Estimation Algorithm in Wireless Sensor Networks
L. Q. Pan, J. Z. Li……115

Recharging Sensor Nodes Using Implicit Actor Coordination in Wireless Sensor Actor Networks
M. Sharifi, S. Sedighian, M. Kamali……123

Finding the Optimal Percentage of Cluster Heads from a New and Complete Mathematical Model on LEACH
A. B. M. Alim Al Islam, C. S. Hyder, H. Kabir, M. Naznin……129

Coordination for Networks of Dynamic Agents with Time-Varying Delays
H. W. Yu, B. S. Zhang, Y. F. Zheng……141

A Novel Approach for Finding a Shortest Path in a Mixed Fuzzy Network
A. Tajdin, I. Mahdavi, N. Mahdavi-Amiri, B. Sadeghpour-Gildeh, R. Hassanzadeh……148

Tree Based Energy and Congestion Aware Routing Protocol for Wireless Sensor Networks
A. H. Mohajerzadeh, M. H. Yaghmaee……161

An Energy-Efficient Access Control Algorithm with Cross-Layer Optimization in Wireless Sensor Networks
Z. Chen, S. Q. Li……168

A Study on Vehicle Detection and Tracking Using Wireless Sensor Networks
G. Padmavathi, D. Shanmugapriya, M. Kalaivani……173

Achieving Directionality and Transmit Diversity via Integrating Beam Pattern Scanning (BPS) Antenna Arrays and OFDM
P. K. Teh, S. A. Zekavat……186

Wireless Sensor Network (WSN)

Journal Information

SUBSCRIPTIONS

The *Wireless Sensor Network* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

E-mail: service@scirp.org

Subscription rates: Volume 2   2010
Print: $50 per copy.
Electronic: free, available on www.SciRP.org.
To subscribe, please contact Journals Subscriptions Department, E-mail: service@scirp.org

Sample copies: If you are interested in subscribing, you may obtain a free sample copy by contacting Scientific Research Publishing, Inc at the above address.

SERVICES

Advertisements
Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: service@scirp.org

COPYRIGHT

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: wsn@scirp.org

Wireless Sensor Network, 2010, 2, 93-99
doi:10.4236/wsn.2010.22013 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Evaluation of Multiusers' Interference on Radiolocation in CDMA Cellular Networks

A. J. Bamisaye, M. O. Kolawole, V. S. A. Adeloye
Department of Electrical and Electronics Engineering, The Federal University of Technology, Akure, Nigeria
E-mail: {*ayobamisaye, kolawolm, vadeloye*}@yahoo.com
Received October 17, 2009; *revised November* 15, 2009; *accepted December* 16, 2009

Abstract

Radiolocation has been previously studied for CDMA networks, the effect of Multiple Access Interference has been ignored. In this paper we investigate the problem of Radiolocation in the presence of Multiple Access Interference. An extensive simulation technique was developed, which measures the error in location estimation for different network and user configurations. We include the effects of lognormal shadow and Rayleigh fading. Results that illustrate the effects of varying shadowing losses, number of base stations involved in position location, early-late discriminator offset and cell sizes in conjunction with the varying number of users per cell on the accuracy of radiolocation estimation was presented.

Keywords: Code Division Multiple Access, Radiolocation, Multiple Access Interference, Base Station, Mobile Station.

1. Introduction

Radiolocation involves: a) identifying the base station (BS) that would participate in the process of subscriber location by selecting a set of BSs within the coverage area that receives intelligible levels of signal from the mobile station (MS) under consideration. b) estimating one-dimensional position which involves each BS, participating in the process, independently producing an estimate of the subscriber location based on its measurements; c) location estimation, that is, estimates from all the participating BSs are used by position location algorithms to produce an accurate estimate of the subscriber location within the coverage area. But, the estimates produced are not always very accurate. The major sources of error in subscriber location systems are: multipath propagation, non-line-of-sight (NLOS), and multiple access interference (MAI). In the case of multipath propagation, accuracy is greatly affected when the reflected rays arrive within a very small period of the first arriving ray. Case is even more worsened when the power of reflected rays is more than the first arriving ray [1]. Several methods have been developed to mitigate the effects of multipath on radiolocation. Typically propagation in wireless communications accrues up to an average of 400-700m [2,3] and biases the estimations. By using the *a prior* information about range error statistics, range estimations made over a period of time and corrupted by NLOS errors can be adjusted to near their correct values. An alternative approach is to reduce the weights of the BSs prone to NLOS reception while estimating location using position location algorithms [4,5]. Co-channel interference is a problem faced by all the cellular systems. In *Code Division Multiple Access* (CDMA) networks, users share the same frequency band, but use unique pseudo-noise (PN) codes. Near far effects in CDMA networks are the biggest source of errors in position estimation. Multiple cellular users who are using the same frequency allocation at the same time cause MAI: it greatly affects the performance of Time of Arrival (ToA) estimation of CDMA systems. In CDMA cellular systems, the MSs are power controlled to combat the near-far effect. Thus, for a CDMA network, time based approach is the most promising technique. This paper investigates the effect of MAI and the accuracy of Radiolocation in CDMA cellular network.

This paper models the intracellular and intercellular multiple access interference (MAI) in Section 2, simulation results are presented in Section 3, and finally in Section 4, the conclusion drawn from the study is summarised.

2. System Model

In a CDMA system all the users share the same frequency band. As a result, at a CDMA receiver signals from users, other than the intended user, act as interfering signals, thereby giving rise to multiple access interference. Figure 1 represents such a situation.

With reference to Figure 1, the coverage area comprises of cells, C_i, $C_j, \cdots\cdots$ and C_p. These cells have n_i, n_j $n_k, \cdots$and C_p users respectively. The BS's *i*, *j* ,*k* ,.... and *p* exercise power control over the users they serve. Let A_i, A_j , A_k...and A_p denote their areas. Only few cells have been considered for simplicity of explanation.

2.1. Modeling Intracellular Multiple Access Interference

In a CDMA system using binary signaling, the radio signal from the k^th user, arriving at the BS is given by:

$$S_k(t-\tau_k) = \sqrt{2P_k}\, c_k(t-\tau_k) b_k(t-\tau_k)\cos(\omega_c t + \varphi_k) \qquad (1)$$

where, P_k is the power received from the k^th user at the BS. Assuming, perfect power control is exercised, we can replace P_k by *P*, where *P* represents the nominal power received at the BS from a user under its power control.

$c_k(t)$ is the spreading (or chip) sequence for a user k.
$b_k(t)$ is the data sequence for a user k.
τ_k is the delay for user k relative to a user 0.
φ_k is the phase change for user k relative to a user 0.
ω_k is the carrier frequency.

In the above equation we assume that there is no multipath interference in the channel. A PN sequence $c_k(t)$ is of the form: [10]

$$c_k(t) = \sum_{j=-\infty}^{\infty}\sum_{i=0}^{M-1} a_{k,i} \prod\left(\frac{t-(i+jM)T_c}{T_c}\right) \qquad (2)$$

$$a_{k,i} \in \{-1.1\}$$

where

T_c represents the chip duration.
MT_c represents the chip repetition period.

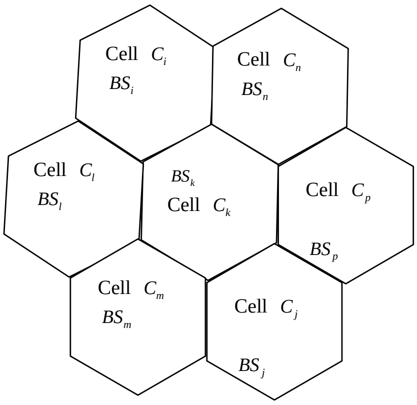


Figure 1. Coverage area.

Π represents the unit pulse function given by

$$\prod(t) = \begin{cases} 1 & 0 \le t \le 1 \\ 0 & otherwise \end{cases} \qquad (3)$$

i is an index to denote a particular chip within a PN cycle. For data sequence $b_k(t)$, *Tb* is the bit period such that

$$Tb = GTc$$

where *G* represents the spreading factor or gain of the CDMA system. It is not necessary that the gain *G* of a CDMA system be equal to *M*. In case they are same, a PN sequence would be repeated for every bit period *Tb*.

The user data sequence b_k(t) is given by

$$b_k(t) = \sum_{j=-\infty}^{\infty} b_{k,i} \prod\left(\frac{t-jT_b}{T_b}\right) \qquad (4)$$

$$b_{k,i} \in \{-1,1\}$$

the signal received at the BS is given by

$$r(t) = \sum_{k=0}^{n_p} S_k(t-\tau_k) + n(t) \qquad (5)$$

where *n(t)* represents a zero-mean white Gaussian noise with two sided power spectral density $\frac{N_o}{2}$, and n_p represents the number of users power controlled by BS p.

To derive a decision statistic, the received signal *r*(*t*) is mixed with the base band, multiplied with the PN sequence of the desired user, and integrated over the bit period T_b.

Assuming that the receiver is phase and delay synchronized with the k^th user, the output of the correlator can be written as [6,7]

$$Zk = \int_{jT_b}^{(j+1)T_b} r(t) c_k(t)\cos(\omega_c t)dt \qquad (6)$$

Assume $\tau_k = 0, \varphi_k = 0$, and that the desired user is user 0. Hence *k*=0. Substituting Equations (1) and (5) in Equation (6) we obtain

$$Z_0 = \int_{t=0}^{T_b} [r(t)c_0(t)\cos(\omega_c t)dt$$

$$= \int_{t=0}^{T_b} [(\sum_{k=0}^{n_p-1} \sqrt{2p_k}\, c_k(t-\tau k) b_k(t-\tau_k)\cos(\omega_c t - \varphi_k)$$

$$+n(t)]c_0(t)\cos(\omega_c t)dt \qquad (7)$$

Z_0 is a decision statistic for the desired user.
Equation (7) can be expressed as

$$Z_0 = I_0 + \zeta + \eta$$

where

1) I_0 is the contribution from the desired user, *i.e.*,

$$I_0 = \int_{t=0}^{T_b} \sqrt{2P_k}\, b_0(t) c_k^2(t)\cos^2(\omega_c t)dt \qquad (8)$$

As $c_k \in \{-1,1\}, c_k^2 = 1$

Hence I_0 reduces to

$$I_0 = \sqrt{\frac{P_0}{2}}\, b_0(t)T_b \qquad (9)$$

2) ζ represents the contribution of MAI and is the summation of $n_p - 1$ terms, I_k, where

$$I_k = \sqrt{2p_k}\, c_k(t-\tau_k) b_k(t-\tau_k)$$

$$\cos(\omega_c t - \varphi_k) c_0(t)\cos(\omega_c t)dt \qquad (10)$$

$$\zeta = \sum_{k=0}^{n_p-1} I_k \qquad (11)$$

3) η represents the contribution of noise and is given by

$$\eta = \int_{t=0}^{T_b} n(t)c_0(t)\cos(\omega_c t)]dt \qquad (12)$$

To determine the variance and mean of η

$$\mu_\eta = E[\eta] = \int_{t=0}^{T_b} E[n(t)c_0(t)\cos(\omega_c t)]dt = 0 \qquad (13)$$

Variance σ_η^2

$$\sigma_\eta^2 = E[(\eta-\mu_\eta)^2] = E[\eta^2]$$

$$= \int_{\lambda=0}^{T_b}\int_{t=0}^{T_b} E[n(t)n(\lambda)\cos(\omega_c\lambda)c_0(t)\cos(\omega_c t)dt\, d\lambda]$$

Now,

$$E[n(t)n(\lambda) = \frac{N_0}{2}\delta(t-\lambda)$$

$$\sigma_\eta^2 = \int_{\lambda=0}^{T_b}\int_{t=0}^{T_b}\frac{N_0}{2}c_0(\lambda)\cos(\omega_c\lambda)c_0(t)\cos(\omega_c t)\delta(t-\lambda)dt\, d\lambda$$

$$\sigma_\eta^2 = \int_{t=0}^{T_b}\frac{N_0}{2}c_0^2(t)\cos^2(\omega_c t)dt$$

$$c_0^2(t) = 1$$

Hence

$$\sigma_\eta^2 = \int_{\lambda=0}^{T_b}\int_{t=0}^{T_b}\frac{N_0}{4}(1+\cos(2\omega_c t))dt$$

$$\sigma_\eta^2 = \frac{N_0 T_b}{4} \qquad (14)$$

Assuming a large number of interferes, by virtue of the central limit theorem (CLT), the distribution of ξ can be approximated by a zero-mean Gaussian distribution [4,7,8] with variance σ_ζ^2 given by [7]

$$\sigma_\zeta^2 = \frac{GT_c^2\sum_{k=1}^{n_p-1}P_k}{6} \qquad (15)$$

Let,

$$\xi = \zeta + \eta \qquad (16)$$

Assuming that the MAI and noise are independent processes, the variance of ξ can be written as

$$\sigma_\xi^2 = \sigma_\zeta^2 + \sigma_\eta^2 \qquad (17)$$

$$= \frac{GT_c^2\sum_{k=1}^{n_p-1}P_k}{6} + \frac{N_0 T_b}{4} \qquad (18)$$

2.2. Modeling Intercellular Multiple Access Interference

Expression for the intercellular interference caused by the users of cell C_i at BS j, represented by I_{ij} can be derived. Let the path loss exponent be *m*. Let the fading on path from this user to cell C_i be Rayleigh distributed, and represented by x_i. Similarly, let the fading on the path from this user to cell C_j be Rayleigh distributed, and represented by x_j. The average of x_i^2 is the log-normal fading on the path from this user to cell *i*, *i.e*, $E[x_i^2|\zeta_i = 10^{\zeta_i/10}$, where ζ_i is the decibel attenuation due to shadowing, and is a Gaussian random variable with zero-mean and standard deviation σ_s [9,10]. Similarly the average of x_j^2 is the log-normal fading on the path from this user to cell· j Let P_k be the nominal power received at BS i from user n_i .It is assumed that the power control mechanism overcomes both the large scale path loss and shadow fading. However, it does not overcome fast fluctuations of signal power due to Rayleigh fading [8,11]. As BS i exercises a power control over the MS, the actual transmission power P_{ac_i} of the MS would be

$$P_{ac_i} = P_i[r_i(x,y)^m 10^{\zeta_i/10}] \qquad (19)$$

where r_i (*x*, *y*) is the distance of the MS from BS *i*. Consequently, assuming uniform user density in the cell, the relative average interference I_{ij} at cell C_j caused by all the users in cell C_i is given by [9]

$$I_{ij} = E[\frac{n_i}{A_i}\iint_{C_i} 1\, \frac{r_i^m(x,y)10^{\zeta_i/10}}{\frac{r_i^m(x,y)}{x_j^2}}dA(x,y)] \qquad (20)$$

by using iterated expectations,

$$E[10^{\frac{\zeta_i}{10}}.x_j^2] = E[E[10^{\frac{\zeta_i}{10}}.x_j^2|\zeta_i,\zeta_j]]$$

$$= E_{\zeta_i\zeta_j}[E[10^{\frac{\zeta_i}{10}}.x_j^2|\zeta_i\,\zeta_j]]$$

Given ζ_i and ζ_j

$$E[x_j^2|\zeta_i,\zeta_j]]$$

is log normal and is equal to

$$10^{-\frac{\zeta_i}{10}}$$

Thus,

$$E[10^{\frac{\zeta_i}{10}}.x_j^2] = E[10^{\frac{(\zeta_i - \zeta_j)}{10}}]$$

Let $X = \zeta_i - \zeta_j$

Thus, X is a Gaussian variable of zero-mean and variance equal to $2\sigma_S^2$

$$E[10^{\frac{\zeta_i}{10}}.x_j^2] = E[e^{\gamma x}]$$

$$E[10^{\frac{\zeta_i}{10}}.x_j^2] = \int_{-\infty}^{\infty} \frac{e^{\gamma x} e^{\frac{\gamma x^2}{4\sigma_S^2}}}{\sqrt{4\sigma_S^2}} dx$$

$$E[10^{\frac{\zeta_i}{10}}.x_j^2] = e^{(\gamma\sigma_s)^2} \tag{21}$$

where γ = In(10)/10 =.2303. Substituting the result back in (20) we obtain

$$I_{ij} = e^{(\gamma\sigma_s)^2} \frac{n_i}{A_i} \int\int_{c_i} \frac{1}{} \frac{r_i^m(x,y)}{r_j^m(x,y)} dA(x,y) \tag{22}$$

If α denotes the voice activity factor, then the above equation becomes

$$I_{ij} = e^{(\gamma\sigma_s)^2} \frac{n_i \alpha}{A_i} \int\int_{c_i} \frac{1}{} \frac{r_i^m(x,y)}{r_j^m(x,y)} dA(x,y) \tag{23}$$

Let K_{ij} denote inter-cell interference factor due to a user in cell *i* at BS *j*.
Hence,

$$K_{ij} = \frac{I_{ij}}{n_i} \tag{24}$$

$$= e^{(\gamma\sigma_s)^2} \frac{\alpha}{A_i} \int\int_{c_i} \frac{1}{} \frac{r_i^m(x,y)}{r_j^m(x,y)} dA(x,y) \tag{25}$$

In our model K_{ii} is zero at cell ii, but not zero otherwise (*i.e*, $K_{ij} \neq 0$). It is important to point out the importance of K_{ij}. K_{ij} gives the interference at BS *j* caused by a single user in cell *i*. Thus, if the total number of users in cell *i* were to change, the new interference levels can be obtained by simply taking a product of K_{ij} and the number of users. This simplifies our calculations as the interference need not be recalculated for the new number of users. Thus, using Equation (23) we can compute relative average intercellular interference for uniform user distribution. Thus, for an uniform user distribution, we can write the total intercellular interference at BS *j* due to users in cell *i* a

$$I_{ij} = n_i \times K_{ij} \tag{26}$$

It should be noted that the above interference calculations are assuming nominal power as unity. If *P* is the nominal power from a power controlled user received at home BS, then Equation (26) would be modified as

$$I_{ij} = p \times n_i \times K_{ij} \tag{27}$$

Equation (27) gives the total intercellular interference at cell C_j due to users in cell C_i.

3. Simulation and Results

Simulation was carried out with the following set of fixed parameters:

A chip rate of 1.2288MHz was selected. Hence, *Rc* = 1.2288 Mcps; gain of the CDMA system is 128. Hence *G*=128; Path loss exponent for mobile communications is 4 [11]. Hence, *m*=4;As we are restricting the interference from the first tier of interferer's, number of cells in the coverage area is 7. Hence, *NBS* = 7; Nominal power of MS, *Pk* = 1; Speed of radio signal, C = 3x10^8 m/s; Thermal noise, $\frac{N_0}{2} = 10^{-8}$; User distribution per cell is uniform and every cell has same number of users *N*. The variable sets of parameters include:

a) *N*: Number of users per cell;

b) σ_s: The standard deviation of shadowing losses in every cell;

c) N_{BS}: Number of BS's participating in radiolocation;

d) Δ: DLL(delay-locked loop) resolution.

e) *R*: Radius of the cells.

We studied the effects of varying shadow losses, varying the number of BS's participating in radiolocation, varying the DLL resolution and varying the size of cells on the accuracy of estimation.

3.1. Effect of Varying Shadowing Losses on the Accuracy of Radiolocation

This experiment was conducted to study the combined effect of varying shadowing losses and the number of users per cell on the error in radiolocation.

1) Set $\Delta = 1/8$, number of BSs participating in radiolocation = 3, radius of the cell = 1500m. and integration period, T_{int} = $128T_c$. Generate a cell site BS database for 7 cells of radius 1500m each, using any computer system that supports the graphical user interface, GUI. Set value of σ_s to 6dB.

2) For the given value of σ_s compute the interference matrix F_{ij}.

3) The number of user is vary per cell from 1 to 100, in steps of 10, and estimate the error in position estimation

4) Similarly, set σ_s =8dB and 10dB, and go to Step 2.

For every setting of σ_s, and number of users per cell, 50 random locations were chosen within the central cell, and estimations were carried out. The final results are an average of the results obtained at the 50 locations. Similar procedure is carried out for the remaining set of experiments.

The plot in Figure 2 shows the variation of error in radiolocation with number of users per cell and σ_s. The mean value of the radiolocation error, tabulated below is determined by taking an average of all the points plotted in the Figure 2. When the number of users is varied from 1 to 100, and other conditions remaining the same, the minimum, maximum and mean values of the observed errors are tabulated in Table 1. The mean error in two-dimensional position estimation remains almost constant when σ_s is increased from 8 dB to 10 dB.

3.2. Effect of Varying the Number of Participating BS's on the Accuracy of Radiolocation

We used 2,3 and 4 BS's to estimate the subscriber location under the above heading. The results are plotted in Figure 3. It was observed that:

1) Accuracy improves drastically if we use more than two BS's for estimation: The accuracy of estimation improves to 71.51m from 697.12m when we employ 3 BS's to estimate the subscriber location instead of 2. Thus, it is very evident that introducing the third estimator has a significant impact on the estimation accuracy and

2) There is no significant improvement in the estimation accuracy when the number of BS's is increased from 3 to 4**:** The mean radiolocation error improves by 6m when we increase the number of BS's to 4. This shows that, there is no effect obtained when we increase the number of BS's from 3 to 4. For applications with lower accuracy requirements, 3 BS's would be sufficient for radiolocation. Table 2, derived from Figure 3, outlines the minimum, maximum and mean values of estimation errors for various values of N_{BS} as the number of users per cell is varied from 1 to 100, other conditions remaining same.

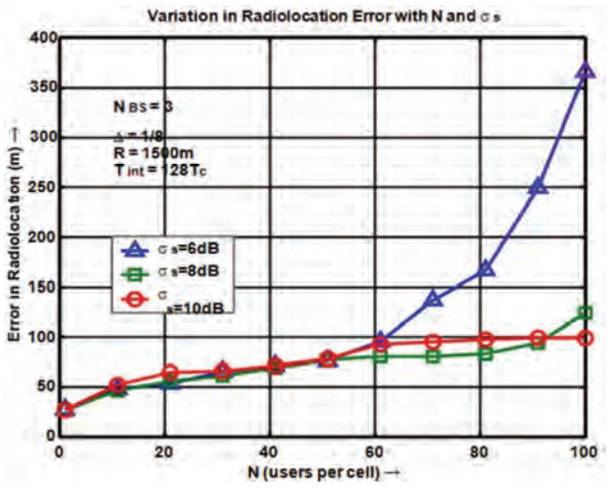


Figure 2. Variation of radiolocation error with N and σ_s.

Table 1. Min and max error in radiolocation for different values of σ_s

σ_s (dB)	Minimun Error (m)	Maximim Error (m)	mean Error (m)
6	26.30	364.33	122.40
8	25.83	122.30	71.51
10	25.85	96.92	75.40

3.3. Effect of Varying the Early-Late Discriminator Offset on the Accuracy of Radiolocation

For our work we have used a non-coherent DLL for estimating the TOA of the received signal. The accuracy of estimating the TOA using a DLL depends on how closely the DLL can track the incoming signal, and this is defined by the parameter Δ. To study the effect of variation of Δ on the accuracy of estimation, we have performed experiments with $\Delta = \frac{1}{2}, \frac{1}{4}$ and $\frac{1}{8}$. The results of the experiment are plotted in Figure 4.

Table 2. Min and max error in radiolocation for different values of N_{BS}.

N_{BS}	Minimun Error (m)	Maximim Error (m)	mean Error (m)
2	543.10	721.64	697.12
3	26.94	24.29	71.51
4	24.70	111.40	65.39

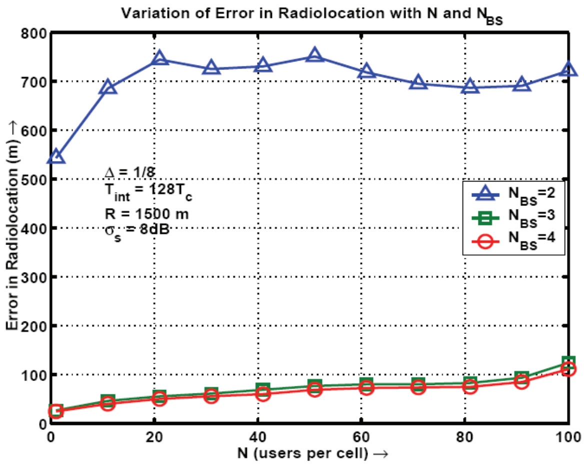


Figure 3. Variation of radiolocation error with N and N_{BS}.

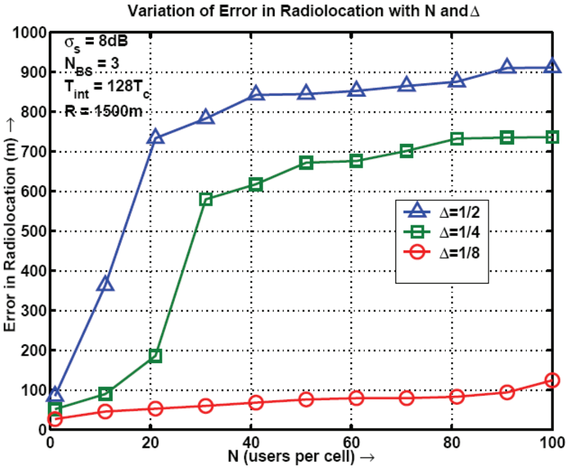


Figure 4. Variation of radiolocation error with N and Δ.

Table 3, derived from Figure 4, outlines minimum and maximum values of errors for different values of Δ as the number of users per cell are increased from 1 to 100, other conditions remaining same. The mean radiolocation error reduces to 524.22m and 70.79m from 733.13m and 524.22m respectively when Δ is reduced from $\frac{1}{2}$ to $\frac{1}{4}$ and from $\frac{1}{4}$ to $\frac{1}{8}$. But, the lowest value of Δ is limited by

a) In practice, the locally generated PN sequence will have to be phase delayed to generate the early and late PN sequences. As per IS-95 standards, one chip period corresponds to 813.80 nSec. Thus, if we were to deploy a tracking loop with Δ=1/16, the requirement on the timing resolution capability on the hardware will be

$$\Delta t = T_c \times \Delta = 813.08\text{nSec}/16 = 50.8175\text{nSec}$$

Implementing such high precision tracking loops is both challenging and expensive.

b) If the DLL employ's a serial search technique, it will have to search through all potential code delays until the correct delay is identified. Suppose, the incoming PN sequence is delayed by T_c. If Δ = 1/k, there are k potential delay values between 0 and T_c that the DLL will have to search through before it can lock to the subscriber signal. Thus, the size of the set of potential delays increases as the value of Δ decreases. The bigger the set of potential delays, the longer it will take for the tracking loop to achieve a lock. The situation becomes more complicated, if we are also estimating the velocity of the subscriber. The set of potential delays, soon transforms into a two-dimensional matrix defining the set of potential delays and velocities. A serial search technique would be inefficient for such cases.

Table 3. Min and max error in radiolocation for different values of Δ.

Δ	Minimun Error (m)	Maximim Error (m)	mean Error (m)
½	84.86	911.02	733.13
¼	51.72	735.97	524.22
⅛	26.94	124.29	70.79

Also, accuracy falls as number of users per cell increases. As seen in Figure 4, the accuracy of estimation, falls as the number of users per cell increases. This is because of the degradation of SNR (signal-to-noise) with increasing number of users per cell.

3.4. Effect of Varying the Cell Size on the Accuracy of Radiolocation

All the earlier experiments were carried out with cells, each of radius 1500m. In this case, we simulated coverage areas with cell radii 100m and 500m. Simulations were carried out under the following conditions: $\Delta = 1/8$; σ_s = 8dB; and number of BSs involved in radiolocation = 3. The number of users is varied from 1 to 100 in steps of 10.

The results were then compared with the results of the experiment carried out under identical situation but using cells of 1500m radius. The results indicate that under perfect power control, the degradation in SNR (signal-to-noise) with number of users is independent of the cell size; and the accuracy of estimation is better with smaller cells.

For the experiment conducted with cells of radii 100m, 500m and 1500m, it is found that accuracy of radiolocation is best for cells of radius 100.

4. Conclusions

Our study has investigated the possibility of accurate subscriber location in CDMA cellular networks in the presence of multiple access interference. Earlier works have ignored the effect of non-orthogonality of the pseudo-noise codes on the estimation accuracy. They usually consider the case of a single MS with no interferers, which is not a practical assumption. In our work, we have studied the effect of number of interferers on the accuracy of estimation by varying the number of users in every cell from 1 to 100. To study the effects of multiple access interference on the accuracy of estimation, we have assumed the presence of a line-of-sight component between the MS and the BS. We have studied the effect of MAI in conjunction with varying shadowing environments, varying tracking capability of the DLL, varying number of BSs participating in radiolocation, and varying cell sizes. The results obtained through the simulations carried out under these different conditions are encouraging and show that radiolocation is possible in a CDMA system, even when multiple access interference is present.

5. References

[1] R. Ilts, "Joint estimation of PN code delay and multipath using extended kalman filter," IEEE Transactions on Communications, Vol. 38, pp. 1677–1685, October 1990.

[2] M. Silventoinen and T. Rantalainen, "Mobile station emergency locating in GSM," Proceeding of IEEE International Conference on Wireless Communications, pp. 232–38, 1996.

[3] M. Wyile and J. Holtzman, "The non line of sight problem in mobile location estimation," Proceedings of IEEE ICUPC, pp. 827–831, 1996.

[4] J. Caffery, Jr, and G. Stuber, "Subscriber location in CDMA cellular networks," IEEE transactions vehicle technology, Vol. 47 No. 2, pp. 406–15, May 1998.

[5] M. Hata and T. Nagatsu, "Mobile location using signal strength measurements in a cellular system," IEEE Transactions Vehicle Technology, Vol. 29, pp. 245–251, May 1980.

[6] R. K Morrow, Jr., and J. S. Lehnert, "Bit-to-bit error dependence in slotted DS/SSMA packet systems with random signature sequences," IEEE Transactions on Communications, Vol. 37, No. 10, October 1989.

[7] T. S. Rappaport, "Wireless communications principles & practice," IEEE Press, New York, Prentice-Hall, 2002.

[8] A. J. Bamisaye and M. O. Kolawole, "Evaluation of downlink performance of a multiple-cell, rake receiver assisted CDMA mobile system," Journal of Wireless Sensor Network, 2009.

[9] R. G. Akl, M. V. Hegde, A. Chandra, and P. S. Min, "CCAP: CDMA capacity allocation and planning," Tech Republic, Washington University, April 1998.

[10] W. C. Y. Lee, "Mobile cellular telecommunications, analog and digital systems," 2nd Edtion, McGraw-Hill, Inc, 1990.

[11] K. Krizman, T. Biedka, and T. Rappaport, "Wireless position location: Fundamentals implementation strategies, and sources of error," in Proceedings of Vehicular Technology Conference, pp. 919–923, 1997.

Wireless Sensor Network, **2010, 2, 100-107**
doi:10.4236/wsn.2010.22014 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Energy Harvesting Strategy Using Piezoelectric Element Driven by Vibration Method

Dong-Gun Kim, So-Nam Yun, Young-Bog Ham, Jung-Ho Park
Energy Plant Research Div., Korea Institute of Machinery& Materials, Daejeon, Korea
E-mail: {*dgkim*82, *ysn*688, *hyb*665}@*kimm.re.kr*

Received October 28, 2009; *revised November* 17, 2009; *accepted December* 8, 2009

Abstract

This study demonstrates a method for harvesting the electrical power by the piezoelectric actuator from vibration energy. This paper presents the energy harvesting technique using the piezoelectric element of a bimorph type driven by a geared motor and a vibrator. The geared motor is a type of PWM controlled device that is a combination of an oval shape cam with five gears and a speed controller. When using the geared motor, the piezoelectric element is size of 36L×13W×0.6H. The output voltage characteristics of the piezoelectric element were investigated in terms of the displacement and vibration. When using the vibrator, the electric power harvesting is based on piezoelectric effect and piezoelectric vibrator consists of a magnetic type oscillator, a cantilever, a bimorph actuator and controllers. Low frequency operating technique using piezoelectric vibrator is very important because normal vibration sources in the environment such as building, human body, windmill and ship have low frequency characteristics. We can know from this study results that there are many energy sources such as vibration, wind power and wave power. Also, these can be used to the energy harvesting system using smart device like piezoelectric element.

Keywords: Energy Harvester, Piezoelectric Element, Wind Energy, Vibration Energy, USN

1. Introduction

The Brilliant technological advancements have allowed low-power consuming elements and modules to be developed, and the Ubiquitous Sensor Network (USN), to be commercialized in 2010, is attracting a high level of interest. The USN is an intelligent social infrastructure that can create information and knowledge from situation perceptions, by detecting, saving, processing and integrating the objects and environmental information obtained through tags and sensor nodes, which are maintained in many locations.

However, when a battery or electrical wires are used to drive the sensor nodes in the USN, it can become quite costly, as well as damaging to the environment. Recognition of this fact has led to the pursuit of studies on wireless energy harvesters that can easily convert energy sources such as solar energy, vibration, heat, wind power or wave energy, which can be commonly found in the environment, into electricity [1–3].

An example of this type of research is the energy harvesting devices invented by a group of MIT researchers. The researchers inserted devices under the sole of shoes, one of which used PZT and the other of which used PVDF. They reported that when using PZT, the maximum voltage of PZT was 150V at a load resistance of 250kΩ. In this case, the maximum electricity was recorded as 80mW, and the average electricity was 1.8mW. For PVDF, the maximum voltage was 60V at a load resistance of 250kΩ. In this case, the maximum electricity was recorded to be 20mW, and the average electricity to be 1.1Mw [4,5].

At Nebraska University (U.S.), Prof. Stephen R. Platt *et al*. conducted, by inserting an energy harvester into a keen joint, a study on a PZT energy harvester to be used as the energy source for a sensor and microprocessor that can convey the information of the status of the energy harvester using the electricity converted from the force put on the knee joint [6]. Shashank Priya of Texas University (U.S.) suggested a theoretical model to calculate electricity generated from a piezoelectric bimorph element in the low frequency band, and announced that it is possible to generate 7.5mW of electricity at a wind velocity of 10mph when a prototype windmill was used to verify the theoretical analysis [7].

These devices are some examples of power generation based on the PZT method, which converts unconsciously wasted energy such as human power, vibration, wind power, or wave power into electricity using a simple structure. Representative materials for piezoelectric elements include Pb(ZrTi)O_3, and PVDF. A great deal of research has been conducted to investigate methods for maximizing the effectiveness of piezoelectric power generation. This paper suggests a method of generating electrical energy with a piezoelectric ceramic using wind, an energy source that is easily applied and from which we can obtain “clean” energy [8].

A characteristic of the piezoelectric ceramic is that it is very difficult to generate energy using small vibrations. Worse still, the piezoelectric ceramic will break when it is the object of a large deformation. This paper presents the energy harvesting technique using the piezoelectric actuator of a bimorph type driven by a geared motor and a vibrator. The geared motor of this study is an equivalent model of the windmill which is rotated by natural wind. The geared motor is a type of PWM controlled device that is a combination of an oval shape cam with five gears and a speed controller. The piezoelectric element is a size of 36L×13W×0.6H. The output voltage characteristics of the piezoelectric element were investigated in terms of the displacement and vibration. The experiments were performed at the geared motor rpm of 40, 81, 126, 172rpm and the piezoelectric element displacement of 100µm, 200µm, 300µm, 400µm, 500µm. The power of 0.068mW was generated in the motor speed of 172 rpm and the piezoelectric element displacement of 500µm. The piezoelectric cantilever (the method by which a piezoelectric element can be attached to a long board with good elasticity) was applied in order to complement this weakness, improve the efficiency of energy harvesting, and effectively convey the mechanical energy to the piezoelectric element.

In addition, to efficiently deliver the mechanical energy to the piezoelectric cantilever, resonance, which is generated only when the natural frequency of the piezoelectric cantilever is the same as the vibration source, should be generated. However, the natural frequency of the piezoelectric element is higher than the 10 to 300Hz generated by the electric products, such as buildings, automobiles, airplanes, refrigerators and washing machines, on which an energy harvester can be attached. As such, the natural frequency of the piezoelectric element should be lowered, so that it can be attached to such places. There are several ways that have been suggested to lower the natural frequency of a piezoelectric cantilever, including lengthening the cantilever or making the end weight heavier.

2. Experimental Apparatus

2.1. Type of Geared Motor

Figure 1 shows the block diagram of geared motor type energy harvester which consists of the energy conversion setup using a geared motor, energy accumulating part and data measuring part. Experimental setup for energy conversion consists of electric energy storage part which is generated by piezoelectric effect, measuring part of piezoelectric element output voltage and displacement measuring part of the piezoelectric element which is caused by a geared motor operation.

Figure 2 shows the photos of the experimental setup for energy conversion and the measuring equipment of output voltage and displacement of the piezoelectric element respectively. In the Figure 2 with energy conversion experimental setup, 36L×13W×0.6H of piezoelectric element and geared motor which has a reduction ratio 1:36 were used. The revolution of geared motor was controlled by PWM method. The displacement ranges were adjusted in the 0 to 600µm for brittleness of the study object piezoelectric element.

Figure 3 shows the energy storage circuit. Bridge diode and condenser were used for voltage change AC to

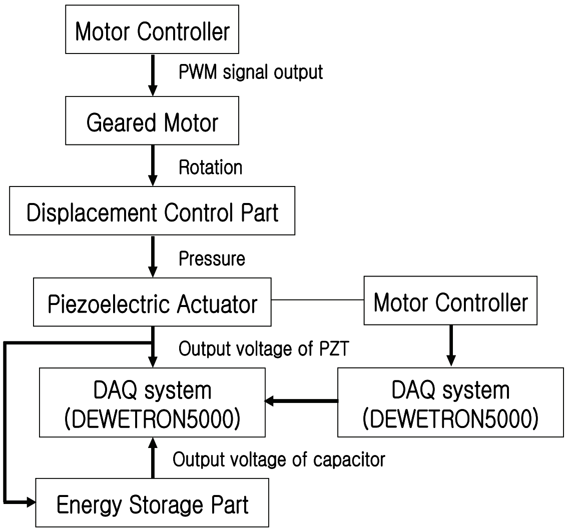


Figure 1. Block diagram of test system and measuring equipment.

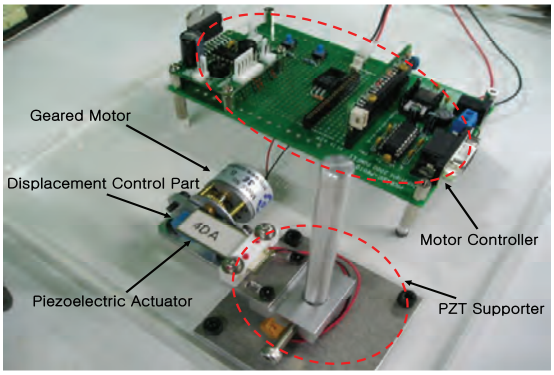


Figure 2. Energy conversion test system.

DC. In the Figure 3, upper circuit shows the rectified one using bridge diode and lower circuit shows the condenser one for filtering the DC voltage fluctuation.

Experiments were operated by three cases.

First, one condition of the variables of piezoelectric element displacement or geared motor speed was fixed and the output voltage from piezoelectric element by change the range variation of an anther condition was measured. Displacement and output voltage of the piezoelectric element were recorded through the A/D converter. Displacement ranges of experimental piezoelectric element were from 100μm to 500μm and geared motor speed with PWM control were 40rpm, 81rpm, 126rpm, 172rpm, respectably.

Second, rectified voltage and filtered voltage with condenser were compared.

Third, electrical energy using Figure 3 was accumulated to the super capacitor for three days and accumulated energy was analyzed. Experiment in the range of motor speed 126rpm with piezoelectric element displacement 500μm was operated.

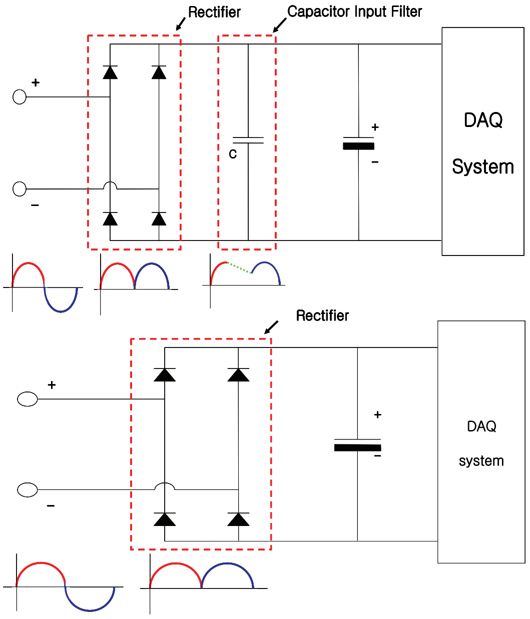


Figure 3. Schematic diagram of energy storage circuit.

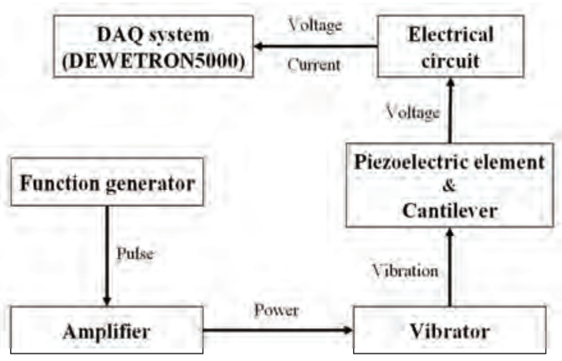


Figure 4. Block diagram of experimental setup.

2.1. Type of Vibrator

Figure 4 shows a block diagram of experimental setup and equipments. The experimental equipments which measure the generated electric energy, current and voltage from piezoelectric element which is operated by vibrator are a power analyzer (NORMA 4000) and a data acquisition system (DEWETRON-5000). A used vibrator is a magnetic type, and frequency and displacement gain can be adjusted by amplifier.

Figure 5 shows a schematic of cantilever with a piezoelectric element for experiment. The material of cantilever is an aluminum and four kinds of cantilever which sizes are 150mm×13mm×1.5mm, 170mm×13mm×1.5mm, 190 mm×13mm×1.5mm, 210mm×13mm×1.5mm were experimented, respectively. The used piezoelectric element is bimorph type ceramic (PI, PL140.10, piezoelectric constant d_{31}=−130×10^{-12}C/N, kp=55%) with a size of 45mm× 11mm×0.6mm.

Figure 6 shows a photo view of experimental setup which consists of an electromagnetic type vibrator with an energy harvesting equipment, a data acquisition system, a function generator and a camera.

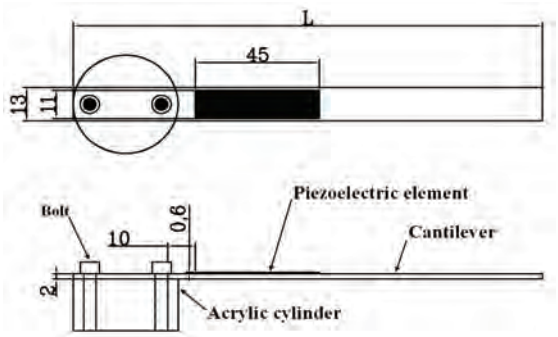


Figure 5. Schematic of cantilever with a piezoelectric element.

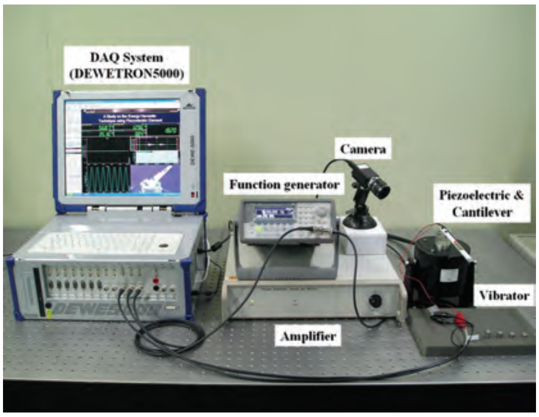


Figure 6. Photo view of experimental setup.

Figure 7 shows the electric circuit used to measure the output of electric current and the electric power generated from the piezoelectric element.

In order to drive the vibrator, the output terminal of the amplifier for the vibrator was linked to the vibrator, and then the output terminal of the function generator was linked to the vibrator as well. The signal coming out of the function generator was set to ±1Vpp, and the output voltage of the amplifier was set to ±12V for the experiment. At this time, the frequency of the function generator ranged from 10~200Hz, which was within the frequency range of the vibrator. The natural frequency of the cantilever was changed to be 0.1Hz unit of the function generator. As the natural frequency, we selected the frequency of the function generator when the voltage generated from piezoelectric element was at a peak.

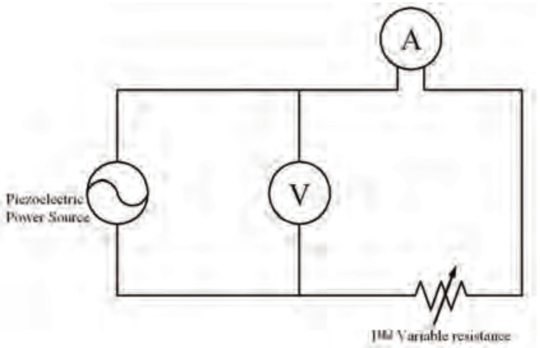


Figure 7. Electric circuit for power measurement.

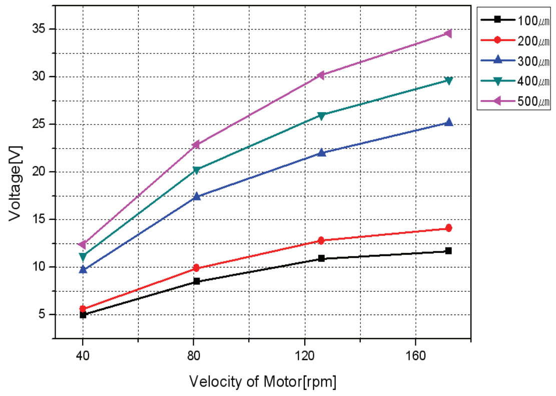


Figure 8. Output voltage by motor speed variation.

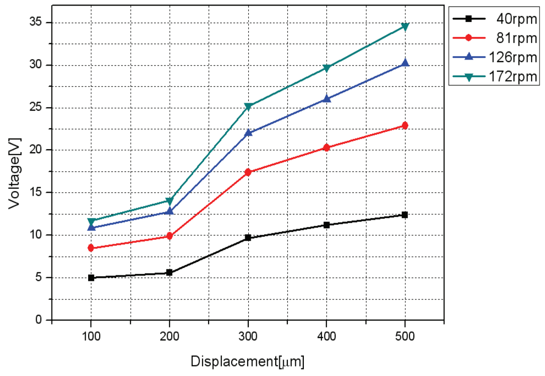


Figure 9. Output voltage by displacement variation.

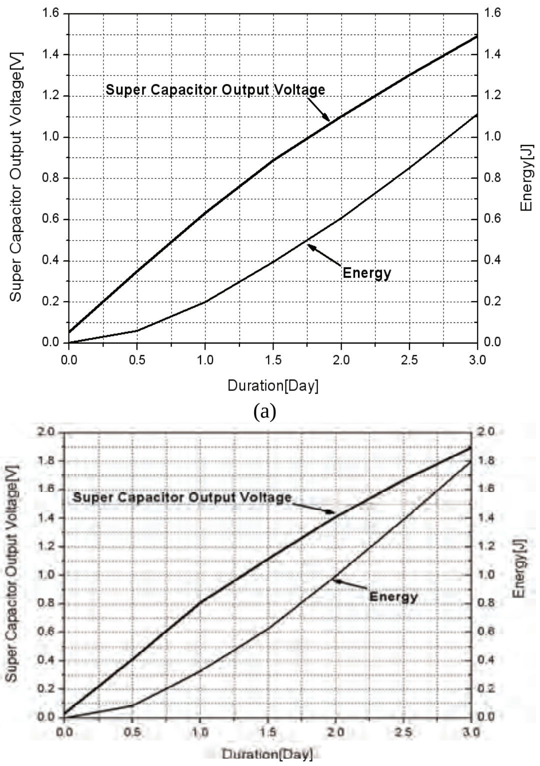
In order to verify the manner in which the characteristics of the voltage were affected by changes in length and weight at the end of the cantilever, we chose a length from among 150, 170, 190,and 210mm in turn, and changed the weight from 0, to 2.22, 4.34, 5.87, 8.66, and to 11.01g, in that order. The electric power generated from the piezoelectric cantilever was calculated by measuring electric current and voltage generated at each part when changing the resistance value.

3. Results and Discussion

3.1. Type of Geared Motor

Figure 8 and Figure 9 are the experimental results of the output voltage characteristics by variation of the piezoelectric element displacement and motor speed. From the Figure 8 and Figure 9, we can know that the output voltage of the piezoelectric element is proportional to the increment of the piezoelectric element displacement and the motor speed. In order to get a static characteristic, the analysis for improvement of the brittleness and the energy density must be researched.

Figure 10 shows the accumulated energy characteris-



(a)

(b)

Figure 10. Energy harvesting characteristics. (a) With rectified & filtered circuit; (b) With rectified circuit.

tics. Experiment was accomplished under the condition of the fixed motor speed, 126rpm and piezoelectric element displacement, 500μm. Electrical energy was accumulated for three days and the characteristics of rectified and filtered one were compared. Energy of the Figure 10 was calculated using Equation (1).

$$E_e = \frac{1}{2} c_r V_r^2 \tag{1}$$

where, E_e: Charged energy in the super capacitor, c_r: Capacitor capacity, V_r: Output voltage.

The thin line of Figure 10 shows the energy derived from the Equation (1). Charged energy using filtered circuit is 1,116mJ and charged energy using rectified circuit is 1,801mJ. From the Figure 10, we can know that the capacitor is an important parameter and optimal circuit of energy conversion system using windmill is needed.

Figure 11 shows the full charge characteristics of the super capacitor and experiment was accomplished under the same condition of Figure 11. The maximum charge voltage of used capacitor is 5V. It was confirmed from the Figure 11 that charging time for 5V is 16 days and charged energy is 12,720mJ. This means that charged energy per an hour is 33.125mJ.

Table 1 shows the wireless sensor system consists of microprocessor, temperature sensor and wireless transmission module. And this table shows operating voltage, consumption current and consumption power for one time transmission of the temperature data.

The minimum energy for one time transmission of the temperature data using a wireless system is calculated by Equation (2).

$$\begin{aligned} U_{\min} &= 55(mW)\cdot 600(ms) + 145(mW)\cdot 10(10ms) \\ &= 34.45(mJ) \end{aligned} \tag{2}$$

From the Equation (2), consumed energy for one time transmission of the temperature information is 34.45 (*mJ*).

Finally, we can know from this study results that there are many energy source such as vibration, wind power and wave power. Also, these can be used to the energy harvesting system using smart device like piezoelectric element.

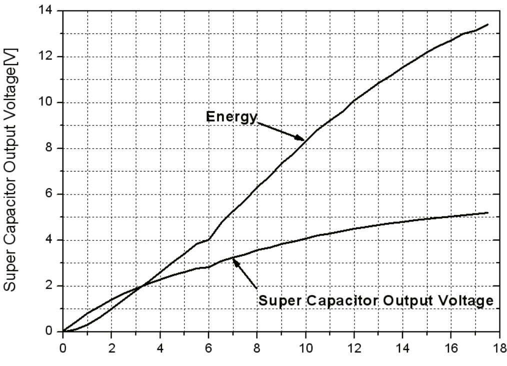


Figure 11. Energy harvesting characteristics of piezo-electric element.

Table 1. Consumption power of energy conversion system.

Items	Operating Voltage [V]	Consumption current [㎃]	Consumption power [㎽]	Remarks
MCU(PIC16F)	5	7	35	Operating
		25	125(10㎳)	Source/Sink
Sensor	5	1	5	Temp. sensor
Tx Module	5	3	15	Supply
		4	20(10㎳)	Transmission
Full load power consumption[㎽]			200	

3.2. Type of Vibrator

Figure 12 indicates the natural frequency measured by changing the length and weight at the end of the cantilever. When there was no mass applied, the natural frequency stood at 132.6, 99.5, 58.9, and 50.0Hz for weights of 0, to 2.22, 4.34, 5.87, 8.66, and 11.01g, respectively, from which it can be observed that as the weight at the end increases, the natural frequency decreases.

The natural frequency of the cantilever(fn) is known to exist in N inverse proportion to the weight applied at the end. The experimental results are consistent with this theory. It is confirmed that as the weight applied at the end increased, the natural frequency was lessened. In addition, as the length of the cantilever was longer, the natural frequency became lower.

Figure 13 shows the results of measuring output voltage depending on the weight applied at the end of the cantilever. The weights applied were 0, 2.22, 4.34, 5.87, 8.66, and 11.01g, respectively. The output voltage was written using RMS values. It is confirmed that as the weight applied at the end of the cantilever increased, the natural frequency became lower.

The experimental results from Figure 13 (a) to (d) con-

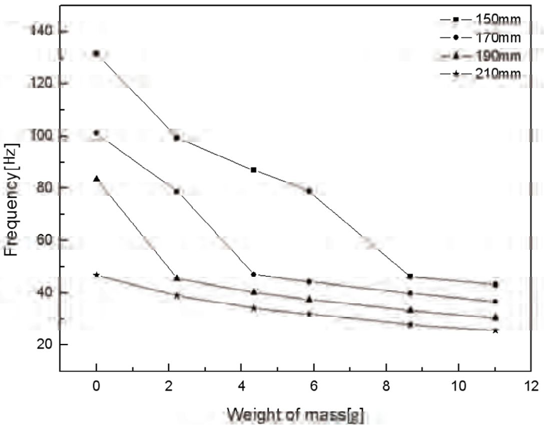
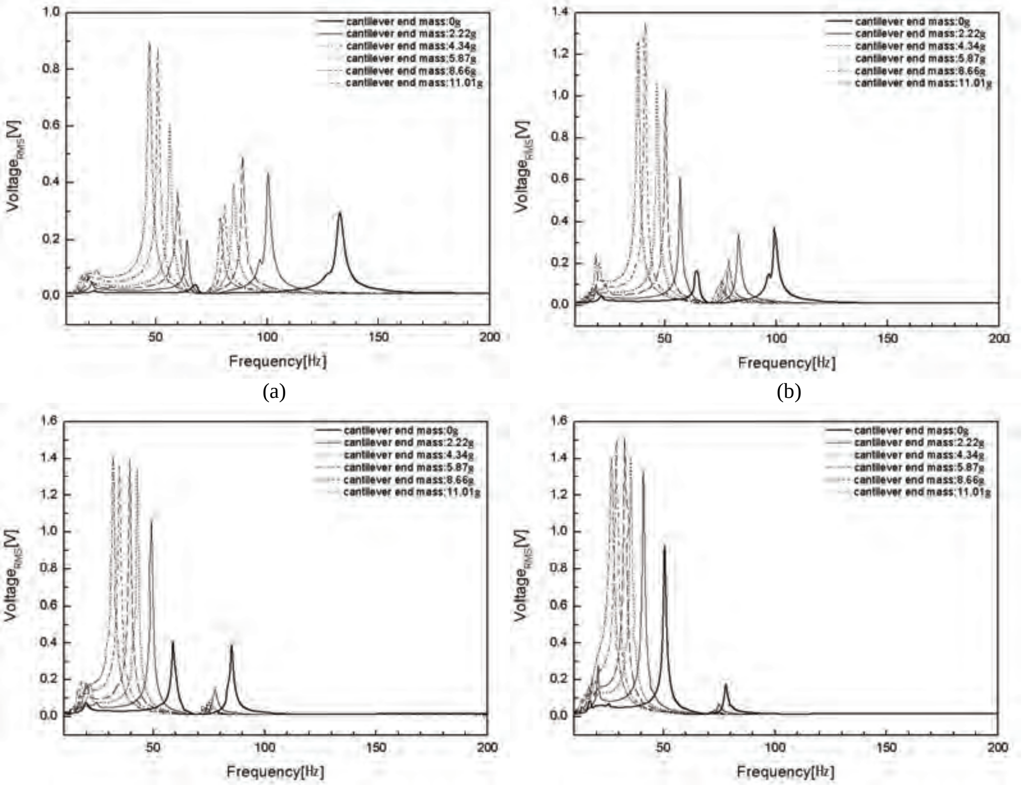


Figure 12. Resonance frequency characteristics by change of cantilever length and mass.



(a) (b)

(c) (d)

Figure 13. Output voltage characteristics by weight variation. (a) Cantilever length, 150mm; (b) Cantilever length, 170mm; (c) Cantilever length, 190mm; (d) Cantilever length, 210mm.

firm that as the weight applied at the end of the cantilever increased, the natural frequency became lower. The experiment also confirms that two distinct natural frequencies were shown, which was caused by the property of the aluminum. The output voltage was higher at the lower natural frequency than at the higher natural frequency. When the length of the cantilever was 150mm, a heavier weight applied at the end of the cantilever resulted in a higher output voltage. We assumed that when the length was 170, 190, or 210mm, the output voltage would be higher as the natural frequency became lower, but as the weight changed from 0 to 5.87g, the output voltage gradually became higher. On the other hand, when the weight was 8.66 or 11.01g, the output voltage became lower or constant. As shown in Figure 13, when the weight applied at the end of the cantilever was 11.01g, the natural frequency became lower, while the output voltage became higher at a length ranging from 150 to 190mm. On the other hand, when the length was 210mm, the output voltage was lower than that at 190mm. The experimental results tell us that if the length of the cantilever is too long, or if the weight at the end of the cantilever is too heavy, we may be able to lower the natural frequency, but we cannot efficiently generate energy.

Figure 14 and Figure 15 show the measured output voltage and electric power, and the natural frequency stood at 58.9 and 31.9Hz when the weight applied at the end of the cantilever was 0g and 11.01g, respectively.

As the value of the variable resistance increased, the output voltage became gradually higher and then constant. At each of resistance, 678Ω and 993Ω, the maximum power was obtained. At this time, electric impedance seems to have occurred. When the output was at the peak, the impedance of the piezoelectric element can be calculated using the following Equation (3).

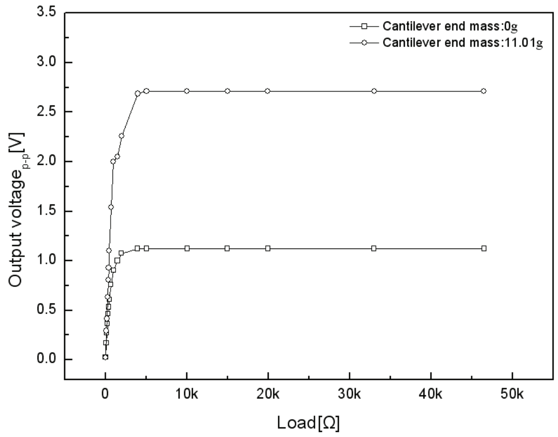


Figure 14. Variation of measured output voltage by electrical load.

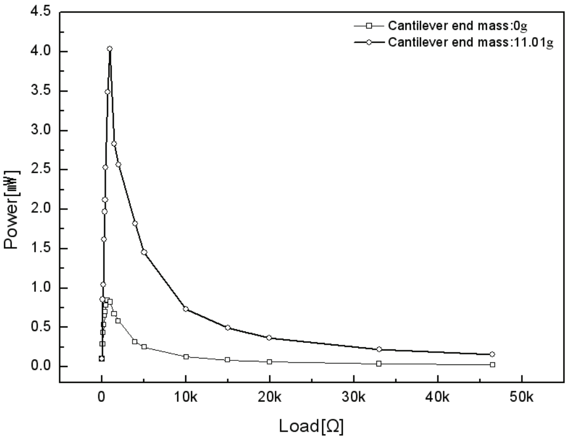


Figure 15. Variation of measured power by electrical load.

$$Z = \frac{1}{2\pi fC} \quad (3)$$

Z, *f*, and *C* represent the impedance of the cantilever, input frequency, and the electricity capacity of the ceramic, respectively. When calculated using the Equation 3 above, the impedance of the cantilever was calculated to be 675.5Ω and 1.247kΩ, respectively, which were shown to be almost identical to the experimental results. When 11.01g was applied, the electricity capacity was identical to that when 0g was applied, while the natural frequency became lower than that when no weight was applied, so that the level of impedance at which the electric power became the highest became higher. When the length of the cantilever was 190mm, and 0 and 11.01g of weight were separately applied, the electric power stood at 0.845mW and 4.036mW, respectively. This represents a 4.78 times difference in electric power between the application of 0g and the application of 11.01g. This result demonstrates that when a heavier weight is applied at the end of the cantilever, the electric power generated becomes higher.

Figure 16 indicates the experimental results of the study, "Frequency Tuning of Unimorph Cantilever for Piezoelectric Energy Harvesting," which is provided for the sake of comparison with this experimental result. However, the size of the piezoelectric element, the piezoelectric constant, electromechanical coupling coefficient, material, size and the weight applied at the end of the cantilever used for this study were different from those used for the aforementioned study, and thus several variables should be taken into account in order to compare the two studies. Nevertheless, the natural frequency becomes lower when a heavier weight is applied than when a lighter weight is applied, an identical tendency shown in both studies. The piezoelectric element used for this study generated nine times more electricity than that of the piezoelectric element shown in Figure 16. It can be analyzed that the output of electricity was improved through such variables as the piezoelectric constant and permittivity, and the size of the cantilever.

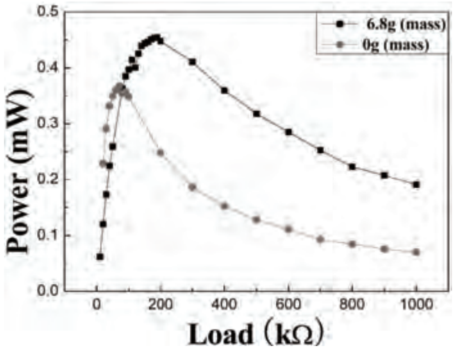


Figure 16. Result of "Frequency tuning of Unimorph cantilever for piezoelectric energy harvesting [9]".

4. Conclusions

The purpose of this study is to understand the characteristics of the energy harvesting devices using vibration or wind. The geared motor method depicts the windmill system and the vibrator is representive vibration source such as buildings, factories, vehicles and human body.

The geared motor method is to simulate the windmill system and consists of geared motor, cam, and energy harvesting equipment with controller. In this study, the equivalent speed of the geared motor by windmill speed was controlled by controller and the output energy from the piezoelectric element with a cantilever was measured by data acquisition system.

The vibrator method is representive the vehicle and consists of the vibrator and energy harvesting system with vibrator controller. In this case, the most important this is to reduce the vibration frequency of piezoelectric element because the environmental vibration source is very low about 300Hz.

In this study, the piezoelectric element with a cantilever and with a mass for reducing the natural frequency of the energy harvesting system was fabricated and experimented. The relation between the geared motor method and the vibrator is to get a green energy from the vibration source and to use waste energy application techniques. The proposed model can be used to predict performance and to provide insights for improving the designs of energy harvesting for the windmill and vibration systems. It will also be useful in the design and analysis of control systems that utilize this in high performance application of the windmill and vibration. Also, this study suggests that the stored energy to the capacitor or secondary battery can be used to the USN and WSN.

5. References

[1] A. Joseph, Paradiso, and T. Starner, "Energy scavenging for mobile and wireless electronics," IEEE Pervasive

Computing, Vol. 4, Issue 1, pp.18–27, 2005.

[2] R. Stephen, Platt, S. Farritor, and H. Haider, "On low-frequency electric power generation with PZT ceramics," IEEE/ASME Transactions on Mechatronics, Vol. 10, No. 2, pp. 240–252, 2005.

[3] T. Starner, "Thick clients for personal wireless devices," IEEE Computer, Vol. 35, No. 1, pp.133–135, 2002.

[4] S. R. Anton and H. A. Sodano, "A review of power harvesting using piezoelectric materials (2003–2006)," Smart material and Structures, Vol. 16, No. 3, pp. 1–21, 2007.

[5] J. Kymissis, C. Kendall, J. Paradiso, and N. Gershenfeld, "Parasitic power harvesting in shoes," Proceeding of the Second IEEE International Conference on Wearable Computing (ISWC), pp. 132–139, 1998.

[6] R. P. Stephen, S. Farritor, K. Garvin, and H. Haider, "The use of piezoelectric ceramics for electric power generation within orthopedic implants," IEEE/ASME Transactions on Mechatronics, Vol. 10, No. 4, pp. 455–461, 2005.

[7] S. Priya, "Modeling of electric energy harvesting using piezoelectric windmill," Applied Physics Letters, Vol. 87, No. 18, 184101–1–3, 2005.

[8] H. A. Sodano, G, Park and D. J. Inman, "Estimation of electric charge output for piezoelectric energy harvesting," Blackwell Publishing Ltd, Vol. 40, No. 2, pp. 49–58, 2004.

[9] H. C. Kim, H. C. Song, D.-Y. Jeong, H.-J. Kim, S.-J. Yoon, and B. K. Ju, "Frequency tuning of unimorph cantilever for piezoelectric energy harvesting," Korean Journal of Materials Research, Vol. 17, No. 12, 660–663, 2007.

Wireless Sensor Network, **2010, 2, 108-114**
doi:10.4236/wsn.2010.22015 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Linear Pulse-Coupled Oscillators Model—A New Approach for Time Synchronization in Wireless Sensor Networks

Zhulin An[1,2], Hongsong Zhu[1,3], Meilin Zhang[1], Chaonong Xu[4], Yongjun Xu[1], Xiaowei Li[1]
[1]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[2]*Graduate University of Chinese Academy of Sciences, Beijing, China*
[3]*Institute of Software, Chinese Academy of Sciences, Beijing, China*
[4]*Department of Computer Science and Technology, China University of Petroleum-Beijing, Beijing, China*
Email: {*anzhulin, xyj, lxw*}*@ict.ac.cn, zhuhongsong@is.iscas.ac.cn, meilin.zhang*1988*@gmail.com, xuchaonong@cup.edu.cn*

Received November 13, 2009; *revised December* 14, 2009; *accepted December* 7, 2009

Abstract

Mutual synchronization is a ubiquitous phenomenon that exists in various natural systems. The individual participants in this process can be modeled as oscillators, which interact by discrete pulses. In this paper, we analyze the synchronization condition of two- and multi-oscillators system, and propose a linear pulse-coupled oscillators model. We prove that the proposed model can achieve synchronization for almost all conditions. Numerical simulations are also included to investigate how different model parameters affect the synchronization. We also discuss the implementation of the model as a new approach for time synchronization in wireless sensor networks.

Keywords: Synchronization, Biologically Inspired Algorithms, Pulse-Coupled Oscillators, Wireless Sensor Networks

1. Introduction

Synchronous flashing of fireflies is a fascinating phenomenon that a large number of scientists have been drawn to research on. The mechanism behind this phenomenon has been investigated for nearly a century. In 1915, Blair observed and tried to examine the scientific reason behind it [1]. He analogize firefly to electric battery—each flash temporarily exhausts the battery, and a period of recuperation is required before the next flash can be emitted. The flash of a leader stimulates the discharge of others, and in the end this makes all the fireflies flash in concert. Richmond presented a similar postulation that if one firefly is ready to flash and sees flashes of others, it starts sooner than otherwise [2]. In 1988, Buck summarized these two battery-analogy mechanisms, and proposed the phase-advanced model. He defined "late sensitivity window" which is a time interval during the period between a firefly's flashings, and concluded that when a photic stimulus (flashing) occurs during the late sensitivity window, it initiates an immediate flash and resets the status of the firefly.

Although the phase-advanced model gives a fine explanation to certain varieties of fireflies' synchronization behavior, the interaction, which is usually called coupling, between fireflies is narrowly limited to late sensitivity window. Peskin extended coupling to any time of the cycle. In his book published in 1975 [3], Peskin proposed a more detailed pulse-coupled oscillators model for the natural pacemaker of a human heart. He modeled a pacemaker as a system consisting of mutual coupled "integrate-and-fire" oscillators. Each oscillator is characterized by state x, which satisfies

$$\frac{dx}{dt} = -\gamma x + S_0 \qquad 0 \le x \le 1 \tag{1}$$

where γ and S_0 are intrinsic properties of the oscillators. When $x = 1$ an oscillator fires then jumps back to $x = 0$, and the states of the other oscillators will be kicked up by coupling strength ε. Through his research, Peskin found that due to coupling, the states of the oscillators tend to come to the same. And as the system evolves, all oscillators would eventually achieve the state of discharging in steps. Peskin proved that for a two oscillators system with small ε and γ, the system approaches a state in which the oscillators are firing synchronously. Mirollo and Strogatz extended Peskin's work, and proved that an N-oscillators system with non-linear dynamics will achieve synchronization for almost all conditions [4].

The models discussed above were all based on pulse-coupling. That is, the oscillators interacted with each other only when one of them fires. In 1975, Kuramoto presented a phase-coupling model [5]. In Kuramoto model, the dynamic of oscillator i in an N-oscillators system can be described as

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N}(\sin\theta_j - \sin\theta_i) \qquad (2)$$

where ω_i is the natural frequency of oscillator i, K is the coupling strength. Kuramoto proved numerically that as the coupling strength is increased above a critical value, the system exhibits a spontaneous transition from incoherence to collective synchronization despite the difference in the natural frequencies of the oscillators [6]. The analytical results of Kuramoto model were obtained by Crawford ten years later. Using center manifold theory, Crawford calculated the weakly nonlinear behavior of the infinite-dimensional system in the neighborhood of the incoherent state. A comprehensive review can be found in [7].

When reviewing the development of studies in synchronous flashing of firefly, it can be observed that the main researchers ranged from biologists to mathematicians and physicians, then to computer scientists and engineers. Recently, the application of the mechanism in synchronization of computer network and neural network makes the research of pulse-coupled oscillators again a popular topic. When applying oscillator based methods to network synchronization, phase-coupling is not ideal, because the coupling during all oscillating cycle is difficult to be implemented. However, the pulse-coupling models proposed by Peskin and Mirollo & Strogatz are not suitable for direct application either, because there are certain assumptions in the model that are difficult to be guaranteed in practical applications. Firstly, those models are all based on instant coupling, implying the pulse is received without any delay, while the propagation delay in wireless communication cannot be neglected. Ernst, Pawelzik, *et al.* [8,9] presented a complete mathematical analysis of two oscillator system with delay, and numerical simulation of multi-oscillators. They came to the conclusion that the synchronization can still be achieved if inhibitory couplings ($\varepsilon < 0$) are adopted. Secondly, all-to-all coupling limits the application in computer networks which are by nature distributed systems. A comprehensive summary of works on Mirollo and Strogatz's model (M&S model) with neighbor communication can be found in [10]. There is also work reported for the application of the pulse-coupled model. Hong and Scaglione firstly implemented the M&S model on a Ultra Wideband network [11], and in [12] they comprehensively investigated how the parameters in pulse-coupled model affected the synchronization precision. Werner-Allen, Tewari, *et al.* [13] discussed their encounter problems when implementing the model on a wireless sensor network (WSN) testbed, and proposed some programming technologies to overcome them.

From the above discussion, we know that the application of pulse-coupled model to the synchronization of wireless network is no easy work. Moreover, when this model is applied to WSN, which usually adopts a micro-controller as its processor, the limitation of computational ability must also be considered. The non-linear dynamic makes it difficult for a micro-controller to work efficiently. (e.g. Ref. [13] used first order Taylor expansion for approximation.) In this paper, we propose a pulse-coupled oscillators model with linear dynamic. The synchronization issue is discussed, and we prove that the presented model can achieve synchronization for almost all conditions. We also include numerical simulations to validate the effectiveness of the model and investigate how model parameters affect the synchronization.

The rest of the paper is organized as follows. Section 2 describes the model and coupling among oscillators. In Section 3 and 4 respectively, we prove two- and multi-coupled oscillators can achieve synchronization for the presented model. Section 5 presents numerical simulation and analysis of the results. In Section 6, we summarize our major work, and discuss the implementation of the model as a new approach for time synchronization in wireless sensor networks.

2. Model Descriptions

For the Peskin model $\frac{dx}{dt} = -\gamma x + S_0$, let $\gamma = 0$, $S_0 = \frac{1}{T}$ and $t \in [0,T]$, we have $\frac{dx}{dt} = \frac{1}{T}$. Integrating the differential equation above yields $x = \frac{t}{T}$. We define T as the cycle period and $\phi = \frac{t}{T}$ as the phase variable. Then we obtain our linear model

$$x = f(\phi) = \phi \qquad \phi \in [0,1] \qquad (3)$$

Due to the fact that the state variable always equals to the phase variable, we use ϕ to represent both the state variable and the phase variable.

Coupling is an important mechanism. It is the only communication method among oscillators. Therefore, a multi-oscillator system can be modeled as an "integrate-and-fire" oscillator network. Each oscillator in the system evolves following linear relationship mentioned in (3). When $\phi_i = 1$, the i th oscillator "fires", and returns to the state $\phi_i = 0$. At the same time, it pulls all the other oscillators up by its coupling strength, or pulls them up to firing, whichever is less. That is,

$$\phi_i = 1 \Rightarrow \phi'_j = \min(1, \phi_j + \varepsilon_i), \quad \forall j \neq i \tag{4}$$

where ε_i is the coupling strength of i. We assume that all the oscillators' coupling strength stay constant and are distributed in a close interval $[a,b]$.

3. Proof of Synchronization of Two Coupled Oscillators

An oscillators system consisting of two coupled oscillators is the simplest, and hence can be studied thoroughly. Therefore, we first discuss the synchronization of two coupled oscillators. We define and compute the firing map and return map, based on which we present the synchronization condition of two oscillators. Then we prove that, if the condition is satisfied, the two oscillators will always achieve synchronization.

3.1. Firing Map and Return Map of Two Coupled Oscillators

Firing map and return map are effective tools for study of the evolution process of oscillators system. Snapshots are taken when an oscillator fires, and by studying these snapshots we can explore the relationship of oscillators phases.

Definition 1 [return map of B about A]: Given two oscillators A and B, assuming that at the instant after one firing of A the phase of B is ϕ, the return map of B about A $R_{B|A}(\phi)$ is defined as the phase of B after the next firing of A.

Definition 2 [firing map of A about B]: Given two oscillators A and B, assumes that at the instant after one firing of A the phase of B is ϕ, the firing map of A about B $h_{A|B}(\phi)$ is defined as the phase of A after the next firing of B.

For oscillators A and B, assume at the instant after A fires, the phase of B is ϕ. After a time period of $1-\phi$, B reaches its firing threshold. At the same time the phase of A changes from zero to $1-\phi$. B fires after an instant, and A jumps to $\phi_A = 1-\phi+\varepsilon_B$ or 1, whichever is less. If $\phi_A = 1$, the two oscillators achieve synchronization; therefore we assume that $\phi_A = 1-\phi+\varepsilon_B < 1$, we have the firing map of A about B

$$h_{A|B}(\phi) = 1-\phi+\varepsilon_B \tag{5}$$

From the analysis above, after one firing, the system has evolved from the initial state $(\phi_A, \phi_B) = (0, \phi)$ to the current state $(\phi_A, \phi_B) = (h_{A|B}(\phi), 0)$. This implies the system is similar as what it was at the beginning, with ϕ being replaced by $h_{A|B}(\phi)$ and two oscillators being interchanged. Therefore, the return map of B about A can be calculated as

$$R_{B|A}(\phi) = h_{B|A}(h_{A|B}(\phi)) = \phi + (\varepsilon_A - \varepsilon_B) \tag{6}$$

3.2. Synchronization Condition of Two Coupled Oscillators

From (6), it can be established that each time when A fires, the phase of B increases by $\varepsilon_A - \varepsilon_B$ from the last firing of A. With this fact, we can infer the following synchronization condition for two coupled oscillators.

Theorem 1 [synchronization condition of two coupled oscillators]: Given two oscillators A and B with their coupling strengths satisfying

$$\varepsilon_A \neq \varepsilon_B \tag{7}$$

they will achieve synchronization.

Proof: From our assumption, we know that ε_A and ε_B maintain constant during the evolution. Hence, since $\varepsilon_A \neq \varepsilon_B \Leftrightarrow \varepsilon_A - \varepsilon_B \neq 0$, we obtain

$$R_{B|A}(\phi_B) < \phi_B \quad \text{if} \quad \varepsilon_A - \varepsilon_B < 0$$
$$R_{B|A}(\phi_B) > \phi_B \quad \text{if} \quad \varepsilon_A - \varepsilon_B > 0$$

Therefore, from any initial state of A and B, the phases of the two oscillators move monotonically toward 0 or 1. In other words, the two coupled oscillators will always reach synchronization.

4. Proof of Synchronization of Multi-Oscillators System

The evolution of a multi-oscillators system is much more complicated than that of two coupled oscillators. When two oscillators fires synchronously, they will clump together, and absorb to a group that acts as one single oscillator with a bigger coupling strength. This makes it easier for other oscillators to join their group, and leads to a positive feedback process. There may exist several groups during the evolution, but as this process goes on, the number of groups decreases, and eventually, all groups will clump to one big group, when the whole system achieves synchronization.

As with the discussion of two coupled oscillators, we first define firing map and return map of multi-oscillators system, and then discuss the absorption, through which the oscillators clump together into groups. Base on these definitions, we present the synchronization condition for a multi-oscillators system. Finally, we prove that the synchronization condition can be satisfied, except for a set of coupling strengths with zero Lebesgue measure.

4.1. Firing Map, Return Map and Absorption in Multi-Oscillators System

Firing map and return map are also essential in discussing synchronization in multi-oscillators system. Consider two oscillators *i* and *j* in an N oscillators system that never synchronize with other oscillators in the system during their evolution. Assume the phase of *j* is ϕ, at the instant that *i* has just fired. Without considering the firings of other oscillators, after $1-\phi$, *j* will fire. However, the firings of other oscillators decrease this period to $1-\phi-\sum_{k\in\Omega}\varepsilon_k$ (Ω is the set of subscripts of oscillators which will fire before *j* fires). Similarly, the firings of oscillators in Ω also increase the phase of *j* by $\sum_{k\in\Omega}\varepsilon_k$. Hence, we have the firing map of *i* about *j*:

$$h_{i|j}(\phi)=1-\phi-\sum_{k\in\Omega}\varepsilon_k+\varepsilon_j+\sum_{k\in\Omega}\varepsilon_k=1-\phi+\varepsilon_j \tag{8}$$

Similar to the case of two coupled oscillators, the return map of *i* about *j* can be written as

$$R_{i|j}(\phi)=\phi+(\varepsilon_j-\varepsilon_i) \tag{9}$$

When two oscillators synchronized, they will clump together and form a synchronous firing group which acts as a single oscillator with larger coupling strength. If that happens we call an absorption occurred, and the coupling strength of a group formed by A and B can be computed as

$$\varepsilon_{AB}=\varepsilon_A+\varepsilon_B \tag{10}$$

4.2. Synchronization Condition of Multi-Oscillators System

Similar to the discussion of two coupled oscillators, our analysis of multi-oscillators system is also based on the return map. The difference is when discussing two oscillators in a multi-oscillator, the firing of other oscillators must also be considered.

Theorem 2 [synchronization condition of multi-oscillators system]: Given an N oscillators system, let $S=\{\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_N\}$ be the set of the coupling strengths of all oscillators in the system. The system will achieve synchronization, if the following conditions are satisfied.

$$\sum_{\varepsilon_m\in S_1}\varepsilon_m\neq\sum_{\varepsilon_n\in S_2}\varepsilon_n \quad \forall S_1,S_2\subset S,\quad S_1\bigcap S_2=\varnothing \tag{11}$$

Proof: First, we are to prove by contradiction that if the condition is satisfied, absorption is sure to occur. Assume absorption never occurs during the evolution of an N oscillators system. For two individual oscillators or oscillator groups *i*, *j* in the system, let

$$\varepsilon_i=\sum_{\varepsilon_m\in S_1}\varepsilon_m,\quad \varepsilon_j=\sum_{\varepsilon_n\in S_2}\varepsilon_n$$

Since ε_i and ε_j are the sums of several εs and none of the oscillators in *i* and *j* are identical, we have

$$S_1,S_2\subset S,\quad S_1\bigcap S_2=\varnothing$$

From (11), we know that

$$\varepsilon_i\neq\varepsilon_j$$

Furthermore, for a multi-oscillator in which absorption never occurs, ε_i and ε_j stay constant. Therefore, similar to the discussion in the case of two coupled oscillators, from the return map (9) we know the phases of *i*, *j* are driven monotonically toward $\phi=0$ or $\phi=1$. That is to say, absorption must occur, which contradicts with our assumption. Therefore, absorption in an N oscillators system always occurs.

From the analysis above, we know that absorption always occurs in a multi-oscillators system satisfying condition (11). And after the absorption, an N oscillators system evolves to an N-1 oscillators system with a slightly different set of parameters. As this process continues, all N oscillators will eventually evolve into one single group, and the synchronization of the whole system is achieved.

We now prove the synchronization condition (11) can be satisfied except for a set of coupling strengths with zero measure.

Theorem 3: For an N oscillators system, each oscillator in the system has a coupling strength within $[a,b]$. The system will achieve synchronization, except for a set of coupling strengths in $[a,b]^N$ with zero Lebesgue measure.

Proof: Let $\mathrm{E}=(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_n)$, which is an element in an N-dimensions subset $[a,b]^N$ of R^n, and $S=\{\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_N\}$ be a set consisting of all the components of E. We are now going to prove the set of E in $[a,b]^N$ which satisfies

$$\sum_{\varepsilon_m\in S_1}\varepsilon_m=\sum_{\varepsilon_n\in S_2}\varepsilon_n \quad \exists S_1,S_2\subset S,\quad S_1\bigcap S_2=\varnothing \tag{12}$$

has a Lebesgue measure of zero.

Let $f(\mathrm{E})=\sum_{\varepsilon_m\in S_1}\varepsilon_m-\sum_{\varepsilon_n\in S_2}\varepsilon_n$, then $f(E)$ can be represented as

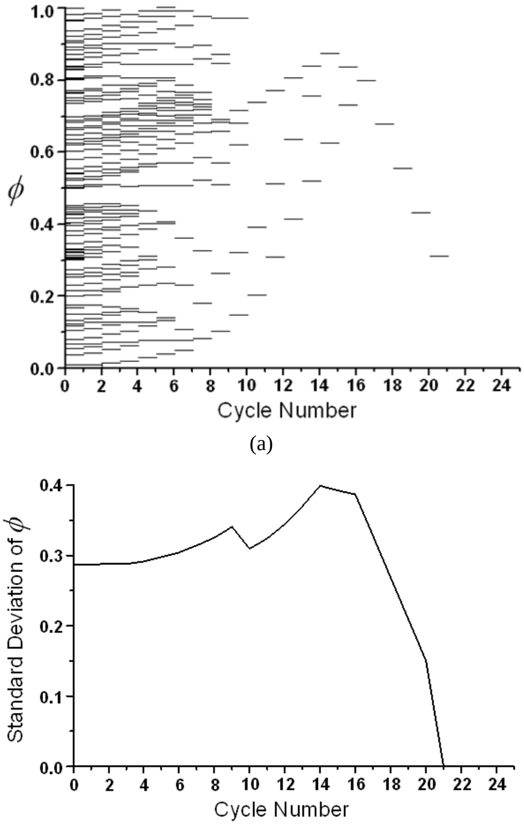
$$f(\mathrm{E})=a_1\varepsilon_1+a_2\varepsilon_2+\cdots+a_N\varepsilon_N$$

where $a_k=\begin{cases}1 & \varepsilon_k\in S_1\\ 0 & \varepsilon_k\in S-S_1-S_2\\ -1 & \varepsilon_k\in S_2\end{cases}$,which indicates $f(\mathrm{E})$ is a hyperplane in $[a,b]^N$. Furthermore, the amount of such hyperplanes is less than $3^N/2$, not unlimited. Therefore, the set of E satisfying condition (12) has the a Lebesgue measure of zero.

With Theorem 2, we found the synchronization condition for a multi-oscillators system, and proved if the condition is satisfied the system will achieve synchronization. Then in Theorem 3, we proved the condition will be satisfied except for a set of coupling strengths with zero measure. Combining the two theorems, we proved the presented multi-oscillators system will achieve synchronization except for a set of coupling strengths with zero measures.

5. Numerical Simulation and Analysis

To validate that the synchronization can be established for the presented model and investigate how model parameters affect the synchronization process, we perform a numerical simulation of the model in a Java environment. Every simulation consists of an initialization stage and a simulation stage. In the former, the parameters of the model are initialized, which includes the number of oscillators (n), the period (T), oscillator phase (ϕ) and coupling strength (ε). Due to the limitation of computer simulation, T, ϕ and ε are all discretized to integers. Specifically, T is set to a large number (10000000), and ϕ is generated randomly between [0, T]. ε is generated randomly between [$\varepsilon\_base - \frac{1}{2}\varepsilon\_base*\varepsilon\_ratio$, $\varepsilon\_base+\frac{1}{2}\varepsilon\_\ base*\varepsilon\ \_ratio$], where ε_base , ε_ratio are "coupling strength base value" and "coupling strength interval ratio" respectively. The simulation stage consists of many cycles. During each cycle, the oscillator with the maximum phase is found first, and the system is forwarded to the firing instant of the oscillator. Then all the oscillators' phases are adjusted according to the coupling strength of the fired oscillator. Finally, all the fired oscillators are combined into a group with new coupling strength computed by (10). This cycle repeats until there is only one group left.



(a)

(b)

Figure 1. Phase and standard deviation of phase during the synchronization process with n = 100, ε_base = 0.005, ε_ratio = 0.1.

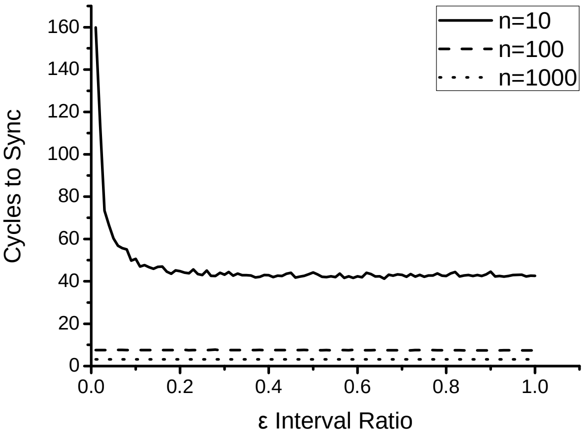


Figure 2. Cycle numbers to achieve synchronization versus ε_ratio with ε_base = 0.01 for n = 10, 100 and 1000.

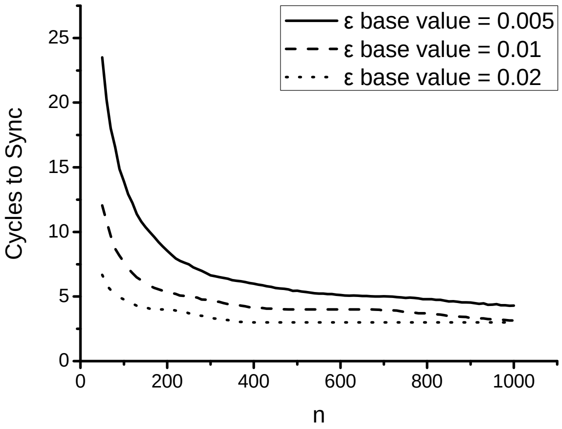


Figure 3. Cycle numbers to achieve synchronization versus n with ε_ratio = 0.1 for various ε_base. Because when n and ε_base are both very small, the cycle numbers until synchronization is going to be very large. To show the detail of all the plots, the figure is ploted from n = 50.

Additionally, to avoid the effect of arbitrary randomness of ϕ and ε, every simulation related to cycles to synchronization is done 1000 times, and the average of 800 values in the center range (eliminate the maximum 100 results and the minimum 100 results) is adopted as the final result.

We first simulate the synchronization process. The results are shown in Figure 1. Figure 1(a) shows the phases of oscillators at different cycles. Each dash in the figure represents the phase of a particular oscillator or oscillator group, and the phases are plotted every time when the phase of oscillator No.0 returns to 0. (As a result, there is always a dash at $\phi = 0$) We can see from the plot that as cycles continue, the number of dashes decrease, indicating that the oscillators gradually clump into groups. In the end, when there is only one group left, the oscillators achieve synchronization. Figure 1(b) shows the standard deviation of the oscillators' phases during the same process. From the figure, we can find that at the beginning the standard deviation generally increases as the evolution progresses, but each time when absorption happens the standard deviation decreases. Near the end, when there are only two groups, the standard deviation decreases dramatically, and finally reaches zero.

We then investigate how the parameters (n, ε_base and ε_ratio) affect the number of cycles needed to achieve synchronization. Figure 2 shows required cycle number to achieve synchronization as a function of ε_ratio with ε_base = 0.01 for n = 10, 100 and 1000. From Figure 2, we can see that when n is big enough the cycle number to synchronize does not change with ε_ratio. We now discuss the reason behind this phenomenon. Suppose i is an oscillator in a multi-oscillators system, then every time when *i* fires, its phase increases by $\sum_{k=1,k\neq i}^{n}\varepsilon_k$. In this simulation, although ε_ratio varies, the sum of all ε lies on ε_base. Furthermore when n is big enough, the sum $\sum_{k=1,k\neq i}^{n}\varepsilon_k$ will approximate the sum of all ε. Therefore, the cycles needed stay the same. We also notice that when n and ε_ratio are both small, more cycles are needed to synchronize. This is because, if ε_ratio is small the ε of all oscillators will be almost the same. Due to the linear dynamic, the deviation of all the oscillators' phases increases slowly, so more cycles are needed.

Following above analysis, we know that the ε_ratio will not affect the result if n is not a very small number. So we fix the value of ε_ratio to 0.1 and discuss how number of cycles needed to synchronize varies with different values of n and ε_base.

Figure 3 shows number of cycles required to achieve synchronization versus n for various ε_base. For a fixed ε_base, cycle numbers decrease with the increase of n. The reason is that the more oscillators there are in the system, the easier it is for the oscillators to absorb to synchronous firing groups. And the ε of a group is the sum of ε of all oscillators in that group, so it is equivalent to increase the ε of the oscillator. Therefore, system tends to synchronize earlier.

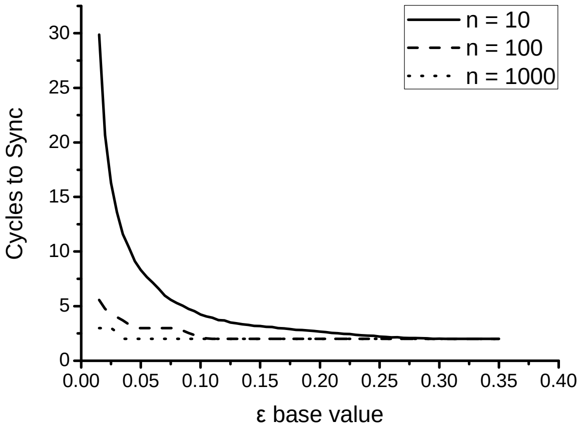


Figure 4. Cycle numbers to achieve synchronization versus ε_base with ε_ratio = 0.1 for various n. Because when n and ε_base are both too small, the cycle number to synchronized is too large. To show the detail of all the plots, the figure is plotted fromε_base = 0.015.

Figure 4 shows cycle numbers to achieve synchronization versus ε_base with $\varepsilon\_ratio = 0.1$ for various values of n. In the figure, we can find that for a certain number of oscillators, the larger ε is, the less cycles are needed to achieve synchronization. This is because the return map increases with the increase of ε, and the system tends to synchronize faster.

From the simulation result presented in this section, we can draw the following conclusion. First, as we discussed in Subsection 4.1, with the evolution of a muti-oscillator system, the oscillators in the system tend to clump together into synchronous firing groups. When there is only one group left, the system achieves synchronization. Second, the number of cycles to achieve synchronization depends on n and ε_base; a larger n or ε_base may lead to faster synchronization, and ε_ratio will have no effect on the number of cycles to synchronize, unless n is very small. To summarize, the simulations match well with our theoretical analysis.

6. Conclusions and Future Work

In this paper, we proposed a model for linear pulse-coupled oscillators system with different coupling strengths. We discussed the synchronization condition for both two- and multi-oscillators system, and proved that the proposed system can achieve synchronization for almost all conditions. Simulations of the model in a Java environment are also included, which validated the model and investigated how different parameters affect the synchronization.

As a swarm of fireflies, a WSN consists of a number of wireless sensor nodes that interact with each other via radio communications. Therefore, if the model presented in this paper is applied as a new approach for time synchronization in WSNs, the algorithm would be more scalable and robust. In the implement, the phase described in the model is represented by a counter, which moves monotonically towards a threshold T, corresponding to the period of the oscillator. When the counter reaches T, it jump back to zero and triggers an interrupt follow with a new cycle. In the interrupt handler, a packet containing the node's coupling strength ε is sent out, which will be used by other nodes to add to their own counter. In this manner, all the counters will be synchronized after a few cycles as what has been discussed in the simulation. However, there is also factors must be considered before this model can be adopted practically, including the message delay, the message collision, the network topology and so on. The implement of the model on a WSN testbed will be included in our future work.

7. Acknowledgement

The research presented in this paper was supported in part by National Natural Science Foundation of China (NSFC) under grants No.(60772070, 60873244, 60633060, 60831160526), in part by High-Tech Research and Development Program of China (863) under grants No.(2007AA12Z321, 2007AA01Z113), and in part by National Basic Research Program of China (973) under grant No.(2005CB321604, 2006CB303000). Authors also wish to acknowledge help of Sen Yu in writing the English version of this paper.

8. References

[1] K. G. Blair, "Luminous Insects," Nature, Vol. 96, No. 2406, pp. 411–415, 1915.

[2] C. A. Richmond, "Fireflies flashing in unison," Science, Vol. 71, No. 1847, pp. 537–538, 1930.

[3] C. S. Peskin, "Self-synchronization of the cardiac pacemaker," in Mathematical Aspects of Heart Physiology, New York University: New York. pp. 268–278, 1975.

[4] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," SIAM Journal on Applied Mathematics, Vol. 50, No. 6, pp. 1645–1662, 1990.

[5] Y. Kuramoto, "Self-entrainment of a population of coupled non-linear oscillators," in International Symposium on Mathematical Problems in Theoretical Physics, pp. 420–422, 1975.

[6] Y. Kuramoto, "Chemical oscillations, waves, and turbulence," Springer Series in Synergetics, Berlin, Springer-Verlag, pp. 164, 1984.

[7] S. H. Strogatz, "From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators," Physica D: Nonlinear Phenomena, Vol. 143, No.1–4, pp. 1–20, 2000.

[8] U. Ernst, K. Pawelzik, and T. Geisel, "Synchronization induced by temporal delays in pulse-coupled oscillators," Physical Review Letters, Vol. 74, No. 9, pp. 1570–1573, 1995.

[9] U. Ernst, K. Pawelzik, and T. Geisel, "Delay-induced multistable synchronization of biological oscillators," Physical Review E, Vol. 57, No. 2, pp. 2150–2162, 1998.

[10] O. Simeone, *et al.* "Distributed synchronization in wireless networks." IEEE Signal Processing Magazine, Vol. 25, No. 5, pp. 81–97, 2008.

[11] Y. W. Hong and A. Scaglione, "Time synchronization and reach-back communications with pulse-coupled oscillators for UWB wireless ad hoc networks," in Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies, 2003.

[12] Y. W. Hong, and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," IEEE Journal on Selected Areas in Communications, Vol. 23, No. 5, pp. 1085–1099, 2005.

[13] G. Werner-Allen, *et al.* "Firefly-inspired sensor network synchronicity with realistic radio effects," in Proceedings of the 3rd international conference on Embedded networked sensor systems. ACM: San Diego, California, USA. pp. 142–153, 2005.

Wireless Sensor Network, **2010, 2, 115-122**
doi:10.4236/wsn.2010.22016 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

K-Nearest Neighbor Based Missing Data Estimation Algorithm in Wireless Sensor Networks

Liqiang Pan, Jianzhong Li
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
Email: {*panlq, lijzh*}*@hit.edu.cn*

Received November 21, 2009; *revised November* 30, 2009; *accepted December* 4, 2009

Abstract

In wireless sensor networks, the missing of sensor data is inevitable due to the inherent characteristic of wireless sensor networks, and it causes many difficulties in various applications. To solve the problem, the missing data should be estimated as accurately as possible. In this paper, a k-nearest neighbor based missing data estimation algorithm is proposed based on the temporal and spatial correlation of sensor data. It adopts the linear regression model to describe the spatial correlation of sensor data among different sensor nodes, and utilizes the data information of multiple neighbor nodes to estimate the missing data jointly rather than independently, so that a stable and reliable estimation performance can be achieved. Experimental results on two real-world datasets show that the proposed algorithm can estimate the missing data accurately.

Keywords: Missing Data, Estimation, Wireless Sensor Networks

1. Introduction

The rapid development of wireless communication techniques, micro-electronics techniques and embedded computation techniques makes Wireless Sensor Networks (WSNs) being applied in many fields [1–4]. WSNs consist of many sensor nodes deployed in a special region where users are interested in, and each sensor node has some computing ability, storage ability and communication ability. Users issue queries to obtain information about the monitored region. Faced with the features of WSNs, many query processing algorithms have been proposed for various applications. However, all these query processing techniques are frustrated by a common problem, that is, the missing of sensor data.

Actually, the missing of sensor data is inevitable due to the inherent characteristic of WSNs. For example, the communication ability of sensor nodes is limited. Some sensor nodes may be isolated from the WSNs for a short or long time due to the influences of surrounding environment such as mountains and obstacles, which results that the sensor data of these nodes may be lost. In addition, the natural environment such as rain, thunder and lightning will influence the sensor nodes' communication quality either and make the communication links between sensor nodes connected and disconnected frequently. This will also result in the sensor data lost during the data transmission. Secondly, the power of sensor nodes is limited. When a sensor node's power is low, it usually works under an unstable state. This not only causes the unstable communication which may make the sensor data lost, but also makes the sensor data sampled be often useless abnormal data (e.g. the temperature of a room is 300℃). The abnormal data is looked as the missing data since it can never be used. When the power of a sensor node is exhausted, the sensor node cannot collect the data any more and the data cached in the storage which have not been sent back may also be lost. In addition, the size of sensor node is small and it is easy to be damaged, which may also result in the lost of sensor data. Due to the reasons given above, no matter how efficient and robust query processing algorithms are developed, the missing of the sensing data is inevitable.

The missing of sensor data will cause many difficulties in various applications. For example, in the data collection applications, the missing data will not only decrease the availability of sensing datasets, but also decrease the efficiency of WSNs greatly. In the research of forest environment [5], a WSN is deployed in the forest to collect the environment variables such as temperature, humidity, atmosphere pressure and sunlight etc. Based on the sensor data collected, biologists can study the forest microclimate, the dynamic tree respiration and growth models etc. However, the data collected by sensor nodes is raw data. Biologists need use some analysis tools on the amounts of raw data and then can get the analysis results and draw a conclusion. Unfortunately, the existing analysis tools which are adopted in these fields, such as support vector machines, principal component analysis and singular value decomposition etc., cannot process the datasets with missing data, and it is infeasible to modify the existing analysis tools for the datasets with missing data. Besides, it is also difficult to process the raw data artificially due to the amount of raw data being huge. So, how to deal with the datasets with missing data frustrates the biologists greatly.

If all the missing data is deleted, much original data information will be lost, which not only decreases the accuracy and the reliability of biologists' research, but also may lead to the wrong research results. In addition, deleting the missing data will also cause the waste of energy. This is because the non-missing data in the same tuple is valuable and believable. Collecting these data also cost much energy. Further more, from the perspective of temporal dimension, the state of the monitored objects at a certain moment can only be observed once, hence the missing data cannot be collected any more, it can only be estimated as accurately as possible.

Datasets [5,6] are two real-world datasets whose data is collected by the WSNs deployed in the Sonoma redwood trees and the Intel-Berkeley lab respectively. They show that there do exist vast missing data in the actual data collection. Since the missing of sensor data is inevitable and it causes many difficulties, developing the high quality missing data estimation algorithms is necessary and urgent. Unfortunately, there exist few works on investigating how to process the missing data efficiently in WSNs so far.

In this paper, a k-nearest neighbor based missing data estimation algorithm is proposed. It adopts linear regression model to describe the spatial correlation of sensor data among different sensor nodes and uses the multiple neighbor nodes' data jointly rather than independently to estimate the missing data. Hence, it can achieve a good estimation effect for the missing data, even for the sensor data of changing irregularly which appears often in WSNs. The performance of the algorithm proposed in this paper is evaluated through extensive experiments on two real-world datasets and compared with the other missing data estimation algorithm. The experiment results show that the proposed algorithm can estimate the missing data more accurately.

The remainder of this paper is organized as follows. In Section 2, an overview of related works is presented. In Section 3, we first give a formal definition of the missing data estimation problem, and then present the algorithm. Section 4 shows the experimental results, and Section 5 concludes the paper.

2. Related Work

Research on missing data estimation has been studied in some fields, such as artificial intelligence [7,8], bioinformatics [9,10–12], and data mining [13,14], but there are few works in WSNs. The works in those fields are not adapted for WSNs, since they do not take account of the features of sensor data being temporal and spatial correlated. The idea of k-nearest neighbor has been adopted in the bioinformatics to estimate the missing values of DNA microarrays [12]. However, the algorithm in [12] is trivial, since it only directly uses the weighted average of the other genes' corresponding data as the estimated values of the missing data. While in WSNs, the sensor data of different nodes is more likely to have some functional relationship rather than being similar in values simply. Thus, the algorithm in [12] is not adapted for estimating the missing sensor data.

Research on query processing in WSNs mainly focuses on processing continuous queries and approximate queries. Processing continuous queries mainly focuses on how to schedule the continuous queries optimally and how to collect the sensing data satisfying these queries energy-efficiently according to network topology and other system characteristics [3,15–18]. Processing approximate queries mainly focuses on how to utilize the temporal-spatial relationship of sensing data to construct appropriate mathematical models and how to use these models to answer the queries approximately, trying to lower communication cost [19–23]. To the best of our knowledge, there exist few works investigating how to process the missing data.

Although [24] and [25] seem to be similar with this paper, they focus on different problems from ours. We focus on how to estimate the missing data as accurately as possible, but [24] focuses on how to save the energy mostly when processing continuous queries. The accuracy of the estimated values is not mainly concerned in [24], and on the contrary, [24] will sacrifice the accuracy of the estimated values for saving energy in many cases. So the methods in [24] are not suitable for our problem. In [25], authors map the sensor network on a graph, and based on the graph theory, they focus on how to estimate the measurement values at arbitrary positions with the least sensor nodes, which is also different from ours. Besides, [25] assumes that the measurement values in the sensor network satisfy some spatial physical laws, and these physical laws can be modeled by the lumped-parameter models. However, WSNs are usually deployed in some unknown regions to execute monitoring task. The models which describe the measurement values of these unknown regions are difficult to be got in fact. So, the techniques in [25] are difficult to be used actually.

Based on the data mining techniques, literature [26,27] studied the estimation of the missing data in data streams. However, the algorithms in [26,27] have great limitations and cannot be used widely. For example, the algorithms in [26,27] can only deal with the discrete data, but not the continuous data. However, in many applications, the environment variables monitored by WSNs such as temperature, humidity, and atmosphere pressure etc. change continuously. In addition, the accuracy and the performance of the algorithms in [26,27] depend on the association rules support and confidence thresholds which need to be pre-specified by users. Since users are not familiar with the monitored environments usually and the vast raw data are difficult to be understood, users may not give the proper thresholds, which results that the accuracy and the performance of the algorithms decrease greatly. Further more, the algorithms in [26,27] estimate the missing data according to the frequent patterns which are pre-computed based on the existing data. If the patten containing the missing data does not appear in the frequent patterns, the missing data cannot be estimated by [26,27]. Compared with the algorithms in [26,27], the algorithm proposed in this paper can solve above problems well.

3. Algorithm Presentation

This paper investigates how to estimate the missing sensor data as accurately as possible. Before introducing the algorithm, we first give the problem definition.

Definition1: The sensor data collected by the sensor node N_i can be looked as a time series S_i=(<y_{i1},T_1>, … ,<y_{in},T_n>), where y_{ik} is the sensor data of N_i at time T_k. For $\forall T_k$, $k\in$ {1, … , n}, if the sensor data y_{ik} is missed, then computing its estimated value $\hat{y}_{ik}$ to minimize the expression $\left|\hat{y}_{ik} - y_{ik}\right|$ is called the missing data estimation problem.

In many applications, the environment variables monitored by the WSNs such as temperature and humidity change continuously. When some data of a sensor node is missed, a naive method for estimating the missing data is, based on the temporal correlation of sensor data, using the non-missing data whose collection time is near to the missing data to estimate them. However, this method works well only when the sensor data changes smoothly and the missing data appears in a short time period. In the other cases, this method may cause large estimation errors. This is because the sensor data in WSNs changes sharply and irregularly often in fact, especially the data sensed in the natural environment since too many uncertain factors, such as environment noise, will affect the variety of the sensor data. So, only depending on the temporal correlation of sensor data to estimate the missing data is not enough in many cases.

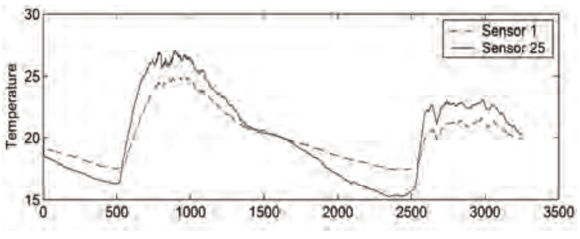


Figure 1. Temperature collected by two sensor nodes.

To estimate the missing data as accurately as possible, we should consider not only the temporal correlation of sensor data, but also the spatial correlation of sensor data. Motivated by this observation, we propose the Applying K-nearest neighbor Estimation (AKE) algorithm which estimates the missing data based on the spatial correlation more than the temporal correlation of sensor data.

As known, there are many sensor nodes deployed in a monitored region. The sensor data of these nodes has spatial correlations. That is, at a moment, the data sensed by the sensor nodes whose locations are nearby is similar or has some relationships. For example, Figure1 shows the temperature observed by two sensor nodes in two days [20]. From Figure 1, we can see that the data sensed by node 1 and node 25 has the similar variety curves. So, when some data of a sensor node is missed, we can estimate them by its neighbor nodes.

For convenience of the algorithm description, without loss of generality, we assume that only sensor node N_i has missing data, and N_i has m neighbor nodes totally, they are N_1, … , N_m respectively. We call the sensor node set consists of N_i ‘s all neighbors as N_i ‘s neighbor node set which noted as $Nb(i)$= { N_1, … , N_m }. For the node N_i, since it has multiple neighbor nodes and for $\forall N_j \in Nb(i)$, N_j has the spatial correlation with N_i, for decreasing the random error caused by a single node when estimating the missing data, AKE looks N_i and its all neighbor nodes as a whole, and utilizes N_i ‘s all neighbor nodes jointly rather than independently to estimate the missing data of N_i.

For $\forall N_j \in Nb(i)$, the functional relationship between the sensor data of N_i and N_j is unknown. Since the locations of the node N_i and N_j are close and an excitation will cause the similar responses on the sensor data of N_i and N_j, the relationship of N_i and N_j can be looked as linear approximately in a short time period. AKE adopts linear regression model to describe the spatial correlation of node N_i and N_j, *i.e.* for any time t, there has

$$y_{it} = \alpha + \beta \cdot y_{jt} + \mu_{jt} \quad (1)$$

where y_{it} is the sensor data of N_i at time t, and y_{jt} is the sensor data of N_j at the same time; α and β are the model coefficients, and μ_{jt} is the random error at time t. According to the theory of linear regression model, to estimate the missing data by utilizing Formula (1), we should first select h (h–1≥2) pairs of known sensor data< y_{in} , y_{jn} >, $n \neq t$, as the sample data, and then use the sample data to regress the coefficients in Formula (1). That is, compute $\hat{\alpha}$ and $\hat{\beta}$, which are the estimated values of α and β, based on the sample data according to least squares principle. When the sensor data of N_i at time t is missed, its estimated value computed by the node N_j can be expressed as:

$$\hat{y}_{it}^{(j)} = \hat{\alpha} + \hat{\beta} \cdot y_{jt} \quad (2)$$

where $\hat{y}_{it}^{(j)}$ represents the estimated value of y_{it}, which computed by node N_j. The deviation between the estimated value $\hat{y}_{it}^{(j)}$ and the real value y_{it} is called the residual of $\hat{y}_{it}^{(j)}$ and y_{it}, which is noted as $e_t^{(j)} = \hat{y}_{it}^{(j)} - y_{it}$. From the least squares principle, it is easy to know that $e_t^{(j)}$ has the minimal variance.

Based on the Formula (2), totally *m* estimated values can be got for a missing data y_{it}, since N_i has *m* neighbor nodes and according to each neighbor node N_j, $N_j \in Nb(i)$, an estimated value $\hat{y}_{it}^{(j)}$ can be computed. To decrease the random estimation error caused by a single neighbor node and improve the estimation system's reliability and stability, AKE uses the weighted average of the *m* estimated values computed by the *m* neighbor nodes as the final estimated value, *i.e.*

$$\hat{y}_{it} = \sum_{j=1}^{m} w_j \cdot \hat{y}_{it}^{(j)} \tag{3}$$

where w_j is the weight coefficient correspondingly, $0 < w_j < 1$ and $\sum_{j=1}^{m} w_j = 1$.

Theorem1: For the estimated values computed by the *m* neighbor nodes of N_i, assume that their corresponding residuals are $e_t^{(1)}$, $e_t^{(2)}$, … , $e_t^{(m)}$ respectively, and these residuals are independent and identically-distributed, then the variance of residual $e_t = \hat{y}_{it} - y_{it}$ is less than that of any $e_t^{(j)}$, j={1,2, … , m }.

Proof: According to the definition of the residuals, there have $\hat{y}_{it} = y_{it} + e_t$ and $\hat{y}_{it}^{(j)} = y_{it} + e_t^{(j)}$. Substitute them into Formula (3), we can get the relationship of e_t and $e_t^{(j)}$, that is, $e_t = \sum_{j=1}^{m} w_j \cdot e_t^{(j)}$. Since $e_t^{(1)}$, $e_t^{(2)}$, …, $e_t^{(m)}$ are independent and identically-distributed, without loss of generality, we assume the variance of $e_t^{(j)}$, j={1,2, … , *m* }, is *DX*. Then, from the properties of variance, we can deduce that the variance of e_t is $\sum_{j=1}^{m} (w_j)^2 \cdot DX$. Obviously, $\sum_{j=1}^{m} (w_j)^2 < 1$ since $0 < w_j < 1$ and $\sum_{j=1}^{m} w_j = 1$. Accordingly, $\sum_{j=1}^{m} (w_j)^2 \cdot DX < DX$.

Next, we discuss the weight assignment in Formula (3). Since many factors will affect the spatial correlations among the sensor nodes, the accuracy of the estimated value computed by different neighbor nodes may be different. Intuitively, a more accurate estimated value should be assigned a larger weight. Considering that, given a set of sample data, the sample determination coefficient $R^2(0 \le R^2 \le 1)$ can reflect the goodness of regression equation fitting the sample data. The more the value of R^2 is, the better the regression equation fits the sample data, which indicates that the estimated values computed by the regression equation will be more accurate. Thus, we can assign the weight according to the sample determination coefficient R^2. For the regression equation $\hat{y}_{it}^{(j)} = \hat{\alpha} + \hat{\beta} \cdot y_{jt}$, assume the sample data consists of *h* pairs of sensor data, then the sample determination coefficient corresponding to this regression equation can be expressed as

$$R_{(j)}^2 = \sum_{n=1}^{h} \frac{(\hat{y}_{in}^{(j)} - \overline{y}_i)^2}{(y_{in} - \overline{y}_i)^2} \tag{4}$$

where $\overline{y}_i$ is the sample mean of node N_i. Accordingly, the weight coefficient corresponding to the estimated value $\hat{y}_{it}^{(j)}$ can be defined as

$$w_j = \frac{R_{(j)}^2}{\sum_{k=1}^{m} R_{(k)}^2} \tag{5}$$

Based on the Formula (5), AKE can assign the appropriate weights to the corresponding estimated values which computed according to different neighbor nodes. Obviously, a more accurate $\hat{y}_{it}^{(j)}$ will contribute more to the final estimated value.

The computational complexity of AKE consists of two components mainly. One is that of computing the coefficients of the regression equation for each neighbor node. Another is that of computing the sample determination coefficient R^2 for each regression equation and then computing the estimated values according to Formula (3). From the theory of linear regression model, it is easy to know that the cost of computing the coefficients for each regression equation is $O(h)$, and *h* is usually an empirical constant. So, the cost of computing the coefficients for all *m* regression equation is $O(m)$. From Formula (4), we can know that the cost of computing $R_{(j)}^2$ is also $O(h)$. Thus, the cost of computing the sample determination coefficient for *m* regression equations and then estimating the missing data based on Formula (3) is also $O(m)$. Due to computing the coefficients of regression equation and computing the sample determination coefficient of regression equation are two individual steps and executed by AKE sequentially, the computational complexity of AKE is $O(m)$, where *m* is the number of N_i 's neighbor nodes.

Since AKE is based on the sensor data spatial correlation to estimate the missing data and linear model is adopted by the algorithm, it will perform best when the sensor data of different nodes is linear correlated absolutely. Although Figure 1 shows that the correlation of sensor data may not be linear sometimes, it does not matter too much. This is because the linear model can approximate the real data correlation well in a short time period, and hence when the sample size is not too much, AKE will perform well even when the sensor data is not linear correlated rigidly.

4. Experiment Results

The algorithm proposed in this paper is implemented by

Java, and evaluated over two real-world datasets whose data is collected by the WSNs indoors and outdoors respectively. One dataset is Intel-lab dataset [6], which is a trace of readings from 54 sensor nodes deployed in the Intel Research Berkeley lab. These sensor nodes collected light, humidity, temperature and voltage readings once every 30 seconds. Another dataset is Redwood dataset [5], which is a trace of readings from 72 Mica2dot sensor nodes deployed throughout two 67 meters high giant redwood trees in a grove. These sensor nodes collected humidity, temperature and voltage readings once every 5 minutes.

To evaluate the performance of the algorithm, we make the algorithm estimate the non-missing data in datasets, and compare the estimated values with their corresponding real data. Before the algorithm is executed, we repair the datasets first since there is many missing data. First of all, we select some fragments of datasets as candidate test dataset. These fragments contain as little missing data as possible. Then, we replace the missing data in the fragments with the average of the non-missing data nearby and to get the test datasets without missing data. Next, we label some data in test datasets as the missing data randomly, and make the algorithm estimate these dummy missing data. Due to the problem focused by this paper is how to estimate the missing data as accurately as possible, we use the accuracy of the estimated values as the evaluation criteria of the algorithm. Specifically, we use Root Mean Square Error (RMSE):

$$RMSE = \sqrt{mean[(\hat{y}_{it} - y_{it})^2]}$$

where y_{it} is the known value which is labeled as missing data, $\hat{y}_{it}$ is the estimated value of y_{it}, and *mean* represents computing the average for all the data labeled as missing value.

we compare the effectiveness of the algorithm proposed in this paper against three algorithms:

LIN method: This is a temporal correlation based missing data estimation method which is based on the linear interpolation model. For the missing data y_{it}, the estimated value $\hat{y}_{it}$ given by method LIN can be expressed as

$$\hat{y}_{it} = y_{iu} + \frac{y_{iv} - y_{iu}}{T_v - T_u}(t - T_u)$$

where y_{iu} and y_{iv} are non-missing data whose collecting moments are near to time *t*.

KNN method: This is a naive spatial correlation based missing data estimation method. For the missing data y_{it}, KNN estimates it with the weighted average of all neighbor nodes' data. *i.e.* $\hat{y}_{it} = \sum_{k=1}^{m} w_k \cdot y_{kt}$, where y_{kt} is the data of $N_k \in Nb(i)$, w_k is the normalized weight coefficient which represents the similarity of the node N_i and N_k. We use KNN as a baseline to show the effectiveness of the algorithm proposed in this paper.

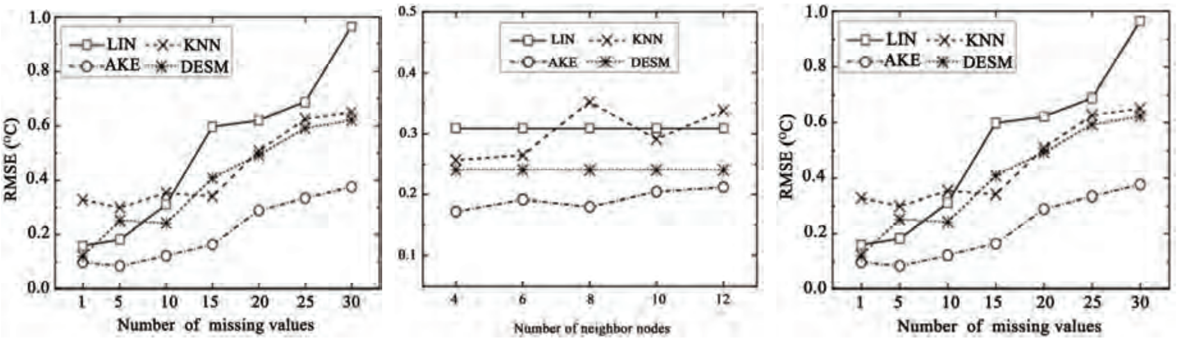
DESM method [24]: This method computes the missing data based on the temporal-spatial correlation. For the missing data y_{it}, the estimated value $\hat{y}_{it}$ given by method DESM can be expressed as $\hat{y}_{it} = (1-\alpha)\hat{y}_{i(t-1)} + (\alpha)\hat{z}$, where $\hat{z}$ is the estimated value of y_{it} computed based on node N_j, $N_j \in Nb(i)$, and α is the Pearson correlation coefficient between N_i and N_j.

Since the data sampling interval, the number of neighbor nodes, and the number of missing data are the main factors which affect the effectiveness of the missing data estimation algorithm, we use them as the experiment parameters. In the experiments, the data sampling interval varies from 1 to 30 minutes, and its default value is 15 minutes. The number of neighbor nodes varies from 4 to 12, and its default value is 8. The number of the missing data varies from 1 to 30, and its default value is 10. In all experiments, while changing a parameter, all other parameters are set as their default values. Specifically, due to the data used in the experiments is collected by the real WSN and the locations of sensor nodes in the real WSN are changeless, the number of neighbor nodes is in logical. In fact, varying the number of neighbor nodes is equivalent to assuming the sensor node has different sensing radius, so that the number of a node's neighbor nodes is alterable.

4.1. Intel-Lab Dataset

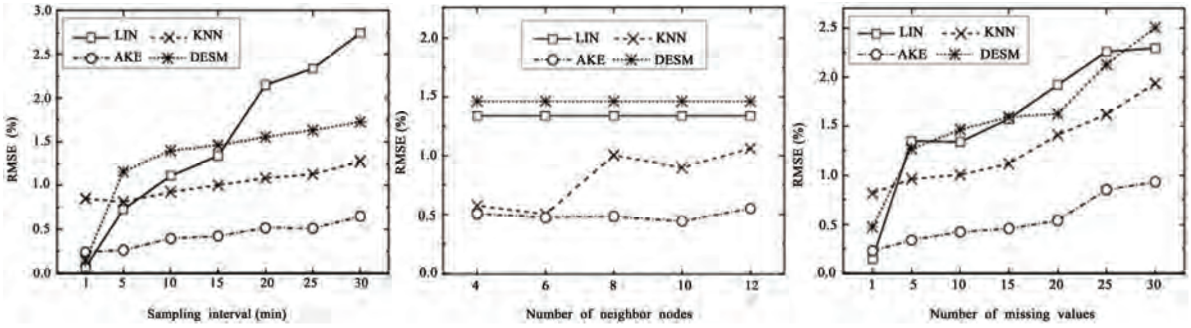
Figure 2 and Figure 3 show the experimental results of the algorithms on temperature and humidity data of the Intel-lab dataset respectively. Figure 2(a) shows that the estimation errors of the algorithms increase when prolonging the sensor node's sampling time interval. This is because all these algorithms estimate the missing data based on the temporal correlation more or less. The increasing of data sampling interval will decrease the temporal correlation of sensor data, which results the algorithms' estimation errors increased, since the sensor data may change greatly with a long time interval. Due to algorithm LIN estimating the missing data according to the temporal correlation absolutely, its estimation error increases most when sampling time interval is enlarged. While, DESM, KNN and AKE estimate the missing data based on the spatial correlation more than temporal correlation, so their estimation errors increase less. Specifically, due to AKE adopts the regression model and uses the multiple neighbor nodes to estimate the missing data jointly, its estimation error increases least.

Figure 2(b) shows that the estimation errors of the algorithm KNN and AKE increase slightly with the number of neighbor nodes increasing. This is because KNN and AKE estimating the missing data are based on the multiple neighbor nodes. Due to the data used in the experiments is collected by the real WSN and the locations



(a) (b) (c)

Figure 2. ***RMSE*** **of the algorithms on temperature data of Intel-lab dataset. (a)** ***RMSE*** **vs. sampling interval; (b)** ***RMSE*** **vs. ＃of neighbor nodes; (c)** ***RMSE*** **vs. ＃of missing data.**



(a) (b) (c)

Figure 3. ***RMSE*** **of the algorithms on humidity data of Intel-lab dataset. (a)** ***RMSE*** **vs. sampling interval; (b)** ***RMSE*** **vs. ＃of neighbor nodes; (c)** ***RMSE*** **vs. ＃of missing data.**

of sensor nodes in the real WSN are changeless, some nodes farer in distance will be involved into missing data estimation when increasing the number of neighbor nodes in the experiments. Since the farer the distance between the sensor nodes is, the lower the spatial correlation of sensor nodes is, using the nodes farer in distance into the estimation equation will decrease the accuracy of the estimated values. From Figure 2(b), we can see that the estimation errors of AKE are always smaller than those of KNN under different number of neighbor nodes. This is because AKE describes the functional relationship of different sensor nodes' data by regression model and estimates the missing data based on the functional relationship of sensor data rather than using the data of neighbor nodes simply which is adopted by KNN method. So, AKE can estimate the missing data more accurately than KNN.

From Figure 2(b), we can also see that the estimation error of LIN and DESM is independent of the number of neighbor nodes. This is because LIN estimates the missing data only using the data of itself and no neighbor nodes data is involved. Similarly, since only one of the neighbor nodes is used by DESM to estimate the missing data, varying the neighbor nodes number has no impact on the estimation error of DESM.

Figure 2(c) shows that the estimation errors of the algorithms increase with the number of missing data increasing. The reason is that much missing data will decrease the temporal correlation between the missing data and the non-missing data, which results the algorithms' estimation errors increased. Due to LIN estimates the missing data according to the temporal correlation absolutely, its estimation error increases most. While, AKE is based on the spatial correlation more than the temporal correlation, so its estimation error increases less than that of LIN.

Figure 3 shows the experimental results of the algorithms on the humidity data, and the similar results can be got. Being different from Figure 2, the estimation errors of the algorithms on the humidity data are larger than that on the temperature data. This is because the correlation of humidity data is lower than that of temperature data since it is more apt to be affected by some environment factors.

From Figure 2 and Figure 3, we can see that no matter on the temperature data or the humidity data, the estimation accuracy of AKE is always better than that of DESM and KNN for all parameters. This is because AKE estimates the missing data not only utilizing the neighbor nodes jointly, but also exploiting the functional relationship of sensor data. So, the estimation performance of AKE is the most stable. In addition, we can alsosee that, with the increasing of the sampling time interval and the number of missing data, the estimation effect of AKE is much better than that of the other algorithms. This is also because the same reasons.

4.2. Redwood Dataset

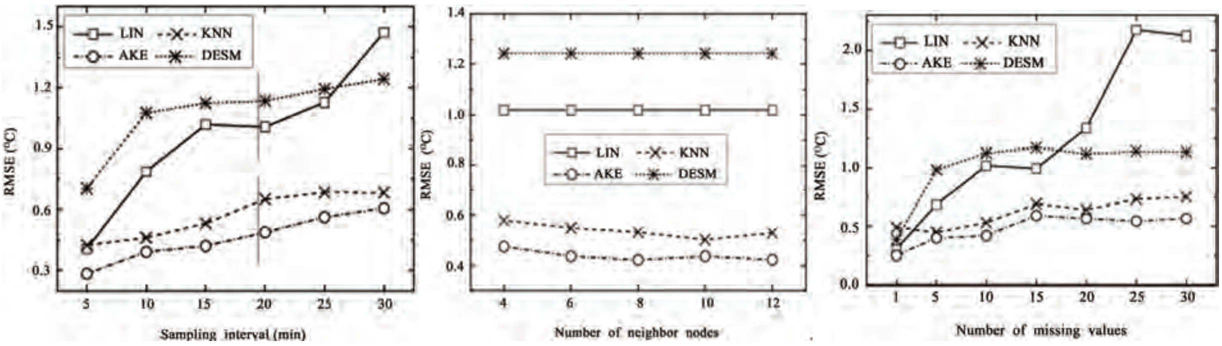
Figure 4 and Figure 5 show the experimental results of the algorithms on temperature and humidity data of the Redwood dataset respectively. From these two figures, we can see the similar experimental results with those of the Intel-lab dataset. The difference is that, on the Redwood dataset, the performance of LIN decreases more greatly when the sampling time interval or the number of missing data increases. This is because the data of the Redwood dataset is collected by the WSN deployed outdoors. The data of outdoors changes more sharply and irregularly, which makes the temporal correlation of the sensor data be lower. Thus, the estimation performance of LIN is worse on the Redwood dataset. Comparatively, due to AKE is based on the spatial correlation more than the temporal correlation, its performance remains relative stable.

From Figure 4 and Figure 5, we note that even KNN which is a naive spatial correlation based missing data estimation algorithm always outperforms LIN for all parameters, especially on humidity data. Thus, we can conclude that, for the data of changing non-smoothly, the spatial correlation based missing data estimation algorithms will perform better.

From Figure 4 and Figure 5, we can also see that the performance gap between AKE and KNN is not too much on the Redwood dataset, especially on humidity data. This is mainly because the sensor data of outdoors changes more sharply and irregularly, the sensor data is in a low correlation. This decreases the advantage of the regression equation, and hence shrinks the performance gap between AKE and KNN. However, no matter in what cases, we can see that AKE always performs the best.

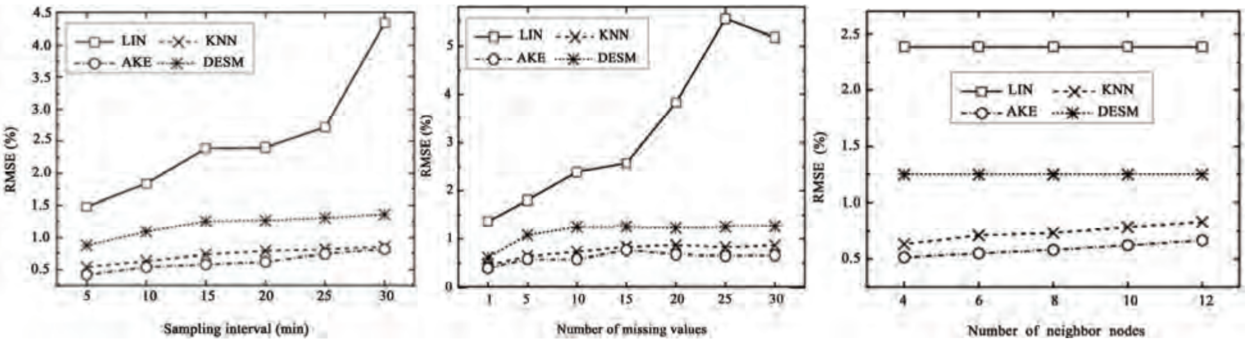
5. Conclusions

Missing data causes many difficulties in various applications of WSNs. Whereas, it is inevitable due to the inherent characteristic of WSNs. To solve the problem, the best way is to estimate the missing data as accurately as possible. In this paper, a k-nearest neighbor based missing data estimation algorithm, called AKE, is proposed. The algorithm is based on the spatial correlation more than the temporal correlation of sensor data, and estimates the missing data utilizing multiple neighbor nodes



(a) (b) (c)

Figure 4. *RMSE* of the algorithms on temperature data of redwood dataset. (a) *RMSE* vs. sampling interval; (b) *RMSE* vs. # of neighbor nodes; (c) *RMSE* vs. # of missing data.



(a) (b) (c)

Figure 5. *RMSE* of the algorithms on humidity data of redwood dataset. (a) *RMSE* vs. sampling interval; (b) *RMSE* vs. # of neighbor nodes; (c) *RMSE* vs. # of missing data.

jointly rather than independently. So, the estimation performance of the algorithm is stable and reliable. In addition, the algorithm estimates the missing data by exploiting the functional relationship of sensor data rather than using the sensor data directly, so, the estimated values computed by AKE are more accurate. Experimental results on two real-world datasets show that the algorithm proposed in this paper performs well both for the data indoors and the data outdoors.

6. Acknowledgments

This work is partially supported by the Key Program of the National Natural Science Foundation of China under Grant No.60533110, the National Grand Fundamental Research 973 Program of China under Grant No.2006 CB303005, the National Natural Science Foundation of China under Grant No.60773063 and No.60703012, and the NSFC-RGC of China under Grant No. 60831160525.

7. References

[1] D. E. Cullar, D. Estrin, and M. Strvastava, "Overview of sensor networks," IEEE Computer, Vol. 37, No. 8, pp. 41–49, 2004.

[2] W. F. Fung, D. Sun, and J. Gehrke, "Cougar: the network is the database," In SIGMOD Conference, pp. 621, 2002.

[3] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks [C]," In SIGMOD. San Diego, California, 2003.

[4] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," In SIGMOD Record, Vol. 31, No. 3, pp. 9–18, 2002.

[5] G. Tolle, "Sonoma redwoods data," 2005. http://www.cs.berkeley.edu/~get/sonoma/.

[6] S. Madden, "Intel Berkeley research lab data," 2003. http://berkeley.intel-research.net/labdata.

[7] X. Zhu, S. Zhang, J. Zhang, and C. Zhang, "Cost-sensitive imputing missing values with ordering," In AAAI. Vancouver, Canada, pp. 1922–1923, 2007.

[8] N. A. Setiawan, P. A. Venkatachalam, and A. F. M. Hani, "Missing attribute values prediction based on artificial neural network and rough set theory," In BMEI. Sanya, Hainan, China, pp. 306–310, 2008.

[9] M. S. B. Sehgal, I. Gondal, L. Dooley, and R. L. Coppel, "Ameliorative missing value imputation for robust biological knowledge inference," Journal of Biomedical Informatics, Vol. 41, No. 4, pp. 499–514, 2008.

[10] M. S. B. Sehgal, I. Gondal, and L. Dooley, "Collateral missing value imputation: A new robust missing value estimation algorithm for microarray data," Bioinformatics, Vol. 21, No. 10, pp. 2417–2423, 2005.

[11] H. Kim, G. H. Golub, and H. Park., "Missing value estimation for dna microarray gene expression data: local least squares imputation[J]," Bioinformatics, Vol. 22, No. 11, pp. 1410–1411, 2006.

[12] O. G. Troyanskaya, M. Cantor, G. Sherlock, P. O. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimationmethods for dna microarrays," Bioinformatics, Vol. 17, No. 6, pp. 520–525, 2001.

[13] C. Zhang, X. Zhu, J. Zhang, Y. Qin, and S. Zhang, "Gbkii: An imputation method for missing values," In PAKDD. Nanjing, China, pp. 1080–1087, 2007.

[14] S. Zhang, J. Zhang, X. Zhu, Y. Qin, and C. Zhang, "Missing value imputation based on data clustering," Transactions on Computational Science, Vol. 1, No. 1, pp. 128–138, 2008.

[15] A. Manjhi, S. Nath, and P. B. Gibbons, "Tributaries and deltas: efficient and robust aggregation in sensor network streams," In SIGMOD Conference. Baltimore, Maryland, pp. 287–298, 2005.

[16] A. Silberstein, K. Munagala, and J. Yang, "Energy-efficient monitoring of extreme values in sensor networks," In SIGMOD Conference. Chicago, Illinois, pp. 169–180, 2006.

[17] D. J. Abadi, S. Madden, and W. Lindner, "Reed: robust, efficient filtering and event detection in sensor networks," In VLDB, Trondheim, Norway, pp. 769–780, 2005.

[18] X. Yang, H. B. Lim, M. T. Ozsu, and K. L. Tan. "In-network execution of monitoring queries in sensor networks," In SIGMOD Conference, Beijing, China, pp. 521–532, 2007.

[19] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," In ICDE. Boston, MA, pp. 449–460, 2004.

[20] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," In VLDB, Toronto, Canada, pp. 588–599, 2004.

[21] A. Deshpande, C. Guestrin, W. Hong, and S. Madden, "Exploiting correlated attributes in axquisitional query processing," In ICDE, Tokyo, Japan, pp. 143–154, 2005.

[22] D. Chu, A. Deshpand, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," In ICDE. Atlanta, pp. 48, 2006.

[23] A. Silberstein, R. Braynard, C. S. Ellis, K. Munagala, and J. Yang, "A sampling-based approach to optimizing top-k queries in sensor networks," In ICDE. Atlanta, pp. 68, 2006.

[24] Y. Li, C. Ai, W. P. Deshmukh, and Y. Wu, "Data estimation in sensor networks using physical and statistical methodologies," In ICDCS, Beijing, China, pp. 538–545, 2008.

[25] H. Zhang, J. M. F. Moura, and B. H. Krogh. "Estimation in sensor networks: A graph approach," In IPSN, Los Angeles, California, pp. 203–209, 2005.

[26] M. Halatchev and L. Gruenwald. "Estimating missing values in related sensor data streams," In COMAD, Hyderabad, India, pp. 83–94, 2005.

[27] N. Jiang and L. Gruenwald, "Estimating missing data in data streams," In DASFAA, Bangkok, Thailand, pp. 981–987, 2007.

Wireless Sensor Network, **2010, 2, 123-128**
doi:10.4236/wsn.2010.22017 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Recharging Sensor Nodes Using Implicit Actor Coordination in Wireless Sensor Actor Networks

Mohsen Sharifi, Saeed Sedighian, Maryam Kamali
School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
E-mail: {*msharifi, sedighian*}*@iust.ac.ir, m_kamali@comp.iust.ac.ir*

Received November 25, 2009; *revised December* 9, 2009; *accepted December* 14, 2009

Abstract

Wireless sensor actor networks are composed of sensor and actor nodes wherein sensor nodes outnumber resource-rich actor nodes. Sensor nodes gather information and send them to a central node (sink) and/or to actors for proper actions. The short lifetime of energy-constrained sensor nodes can endanger the proper operation of the whole network when they run out of power and partition the network. Energy harvesting as well as minimizing sensor energy consumption had already been studied. We propose a different approach for recharging sensor nodes by mobile actor nodes that use only local information. Sensor nodes send their energy status along with their sensed information to actors in their coverage. Based on this energy information, actors coordinate implicitly to decide on the timings and the ordering of recharges of low energy sensor nodes. Coordination between actors is achieved by swarm intelligence and the replenishment continues during local learning of actor nodes. The number of actors required to keep up such networks is identified through simulation using *VisualSense*. It is shown that defining the appropriate number of actor nodes is critical to the success of recharging strategies in prolonging the network lifetime.

Keywords: Wireless Sensor Actor Networks, Coordination, Energy Harvesting, Swarm Intelligence

1. Introduction

Wireless sensor and actor networks (WSANs) are made of two types of nodes called sensor nodes and actors. Sensor nodes are tiny, low-cost, low-power devices with limited sensing, computation, and wireless communication capabilities. Actors are usually resource-rich nodes with higher processing capabilities, higher transmission powers and longer life. In WSANs a large number of sensor nodes are randomly deployed in a target area, may be in the order of hundreds or thousands, to perform a coordinated sensing task. Such a dense deployment is usually not necessary for actor nodes because actors are sophisticated devices with higher capabilities that can act on wider areas. Actors collect and process sensor data and perform actions on the environment based on their gathered information.

Common applications of WSANs include all types of surveillance, target tracking, attack detection, medical sensing, and environment monitoring. Actors in many applications should coordinate together to maximize their overall task performance by sharing and processing the sensors' data, making decisions and taking appropriate actions [1].

One of the greatest concerns in WSANs is energy especially for sensor nodes. A sensor node without energy cannot do its duties unless the source of energy is recharged or changed. Since such networks are usually deployed in large scales and for a long period of time, human intervention for replenishment of energy is not feasible.

The rest of paper is organized as follows. Section 2 presents notable related work. Section 3 describes our energy harvesting approach in detail. Section 4 argues the validity of the approach and presents simulation results. Section 5 concludes the paper.

2. Related Work

Many researches on extending the lifetime of WSANs have focused on minimizing energy usage in different layers of WSANs like in data aggregation to decrease data traffic [2], in energy-efficient networking protocols [3], in managing sleep cycles [4] and in using low power MAC [5].

There are other variants to tackle energy problem that consider different ways of recharging or changing the energy sources (mostly, batteries) of sensor nodes. A method to improve the battery lifetime of sensor nodes is to supplement the battery supply with environmental energy. Environmental energy that is usually harvested to generate electricity includes solar, wind, water and thermal energies.

The main sources of energy for use by sensor networks are solar, mechanical and thermal energies. Solar power is the current matured energy among other forms of energy harvesting and is more attractive in outdoor applications. A solar energy harvesting module for use by Crossbow/Berkeley motes has been developed by Lin *et al*. [6] in the Helimote project.

Vibration and mechanical energy are generated by movements of objects. Traffic sensors [7] are powered by the short duration vibration when a vehicle passes over the sensor.

Thermal energy harvesting uses temperature differences to generate electricity. Thermo generators (TEGs) [8] harvest energy from the human body and are used by devices with direct contact to the human body.

Moreover, some other works have focused on energy replenishment in sensor networks by robots [9]. Robots based on solar cell are used to recharge sensor nodes and to water plants. Another energy harvesting study [10] with focus on mobile nodes has also researched the feasibility of a system wherein mobile nodes have the ability to move in search of energy and delivery of energy to static nodes.

3. Energy Harvesting Through Actor Nodes

In this paper we propose a new approach for recharging sensor nodes by actor nodes to prolong the lifetime of the network. We assume that sensor nodes are static and actors, which are charged by solar cells, are mobile nodes that charge sensor nodes and do the actuating tasks too.

To replenish the network energy, actor nodes learn the usage energy model of the network. They then coordinate to select appropriate sensors to service and move towards their selected sensor nodes to charge them.

Whenever actors request energy related information of nearby sensor nodes, these sensor nodes send their power capacity, their current power, their power usage ratio, their location and their interest to the requesting actor. This information helps actors to find their service area for recharging sensors. Power capacity shows the maximum energy that a node can have, and the current power shows the remaining energy. Interest is a parameter that shows the tendency of sensor nodes to actor nodes. Whenever a sensor node receives a message from an actor node, the interest of the sensor node is increased to that actor node. In addition, when an actor charges a sensor node, the sensor node's interest to that actor will be highest. When power decreases in sensor nodes, the interests to actor nodes also decrease. In fact, actor nodes learn the usage energy model of the network by power information that is gathered from sensor nodes. When an actor learns the usage energy model of sensor nodes, it figures out the sensor nodes to target for recharge. According to the achieved model and by considering the sensors' parameters, the actor selects an appropriate sensor node for recharging and then moves towards that sensor and charges it.

Figure 1 shows the pseudo code for recharging sensors by actors. When an actor receives a message from a sensor node, the actor selects the part of message that relates to energy information of the sensor. EnergyList is a list that holds energy information of sensor nodes. Then SelectSensor method chooses the best sensor node that needs to be charged. A sensor is selected for charge by

```
//Preposition: actor has gathered energy information of nearby sensor nodes, implying that the state of actor is getting energy info

DoRechare(ActorID)
EnergyList ← GetEnergyInformationOfSensors();
//It returns a List of <SensorID, CurrentPower, PowerRatio, InterestToActor, Location>
SelectedSensor = SelectSensorForCharge(EnegyList);
MoveToSensor(SelectedSensor.Location);
//actor state: moving
ChargeSensor(SelectedSensor);
//actor state: charging
SetFullSensorInterestToActor (SelectedSensor, ActorID);
Return;
//actor state: getting energy info

SelectSensor(EnergyList)
For each node Є EnergyList
   If (node.InterestToActor == ActorId)
        node.compareValue ←
                   (node.currentpower / node.powerRatio +0.1))
   end if
end for
return FindMinValue(EnergyList.compareValue)
```

Figure 1. A pseudo code for recharging sensor nodes by actor nodes.

this method based on the energy information of sensor nodes in the EnergyList. Afterwards, the location of the selected sensor node is passed to MoveToSensor and the actor moves towards the sensor and charges the selected sensor node.

During the time actors are learning the energy usage model of sensors, the network is divided into parts. This division is formed by energy information that is sent by sensors.

If the network architecture is automated, there is no central controller (sink) and actor nodes process all incoming data and initiate appropriate actions [11]. In such a network, events might happen everywhere uniformly and the usage of power for each sensor node is considered approximately the same as for other nodes. In this scenario, the sensor node deployment has a random uniform distribution; the divided parts are about the same size.

However, if the network architecture is considered semi-automated, the sink in a WSAN coordinates network activities [11] and nodes that are closer to the sink lose more power because they are involved more in communication than nodes farther from the sink. So in semi-automated networks, the usage of power in a sensor differs from power usage in other sensor nodes, resulting in different sizes of network partitions. Because sensor nodes closer to the sink lose their power faster, actors have to recharge them more frequently. Therefore, the service area of actors near the sink is smaller in size.

4. Experimental Validation

The proposed approach was simulated by *VisualSense* [12] with different number of actor nodes and 100 sensor nodes. Actors and sensor nodes were randomly placed in an area of 1000m × 1000m. The transmission ranges for each actor node and for each sensor node were 100m and 50m, respectively. The model was used in different scenarios to evaluate the performance of the model. We used a short range for actor nodes because our algorithm is really localized.

In Figure 2, there are 3 actor nodes and 80 sensor nodes scattered randomly in the area. Also there is a sink at the bottom right corner of the area. Actually, the network is assumed semi-automated. As it is shown, the area that actor #1 has created is smaller than the area of actor #2 and actor #3. In this scenario, since energy of sensor nodes near the sink decreases by higher ratio, actor #1 has to act in a smaller area to be able to supply enough energy to sensor nodes. As in this scenario, there are limited actor nodes with a lot of sensor nodes, so some sensor nodes die out. But because there are a lot of sensor nodes, sensing coverage does not decrease quickly.

The learning steps of actors for the above scenario are shown in Figure 4. As it is depicted in the third part of Figure 4, sensor nodes at the left top corner need not to be charged because they are not involved in interface communication to other nodes; this means that they only use power when they need to send their information.

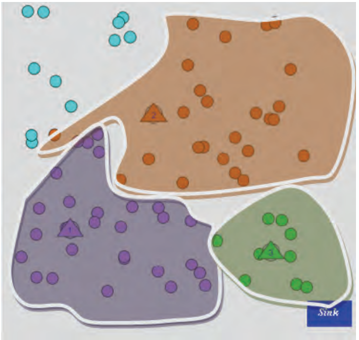


Figure 2. Sensor nodes charged by actor nodes in a network wherein the sink is placed at the right bottom corner (circles represent sensor nodes and triangles represent actor nodes).

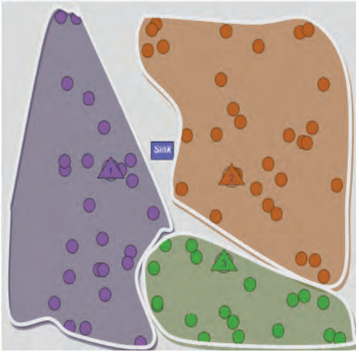


Figure 3. Charging sensors by actor nodes in a network with the sink placed at the centre.

Figure 3 shows the same scenario as the previous scenario except that the sink is placed in the middle of the area. As it is shown in Figure 3, network service areas are different from those shown in Figure 2. Furthermore, Figure 5 shows the evolutionary progress of the service area formation as a result of application of learning process.

Figure 6 shows a scenario for an automated architecture containing the same number of sensor and actor nodes as in the last two scenarios. Figure 7 shows the learning process in this scenario and illustrates how some sensor nodes die out because of limited number of actor nodes.

In our proposed approach, we have an important challenge for the number of actors. Some important parameters that affect the number of actors needed for the network are the number of sensor nodes, the area, average recharge ratio, local recharge ratio, speed of actors, charge time and the network architecture. If the number of actors does not match the network requirements, the network sensing coverage will decrease, leading to a dead network.

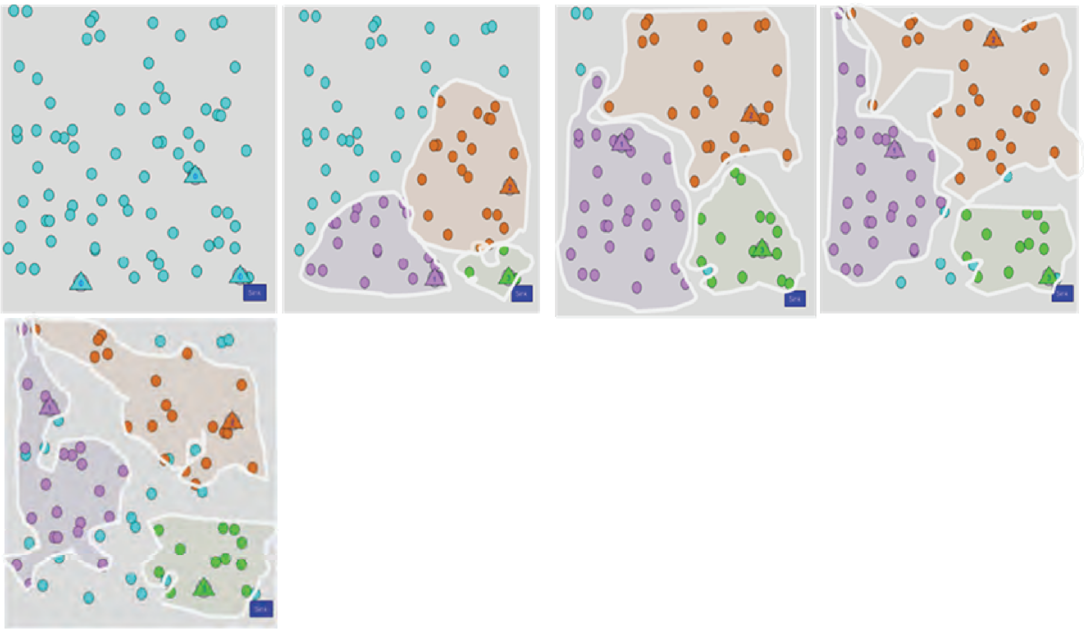


Figure 4. Learning process in a semi-automated network with the sink placed at the right bottom corner.

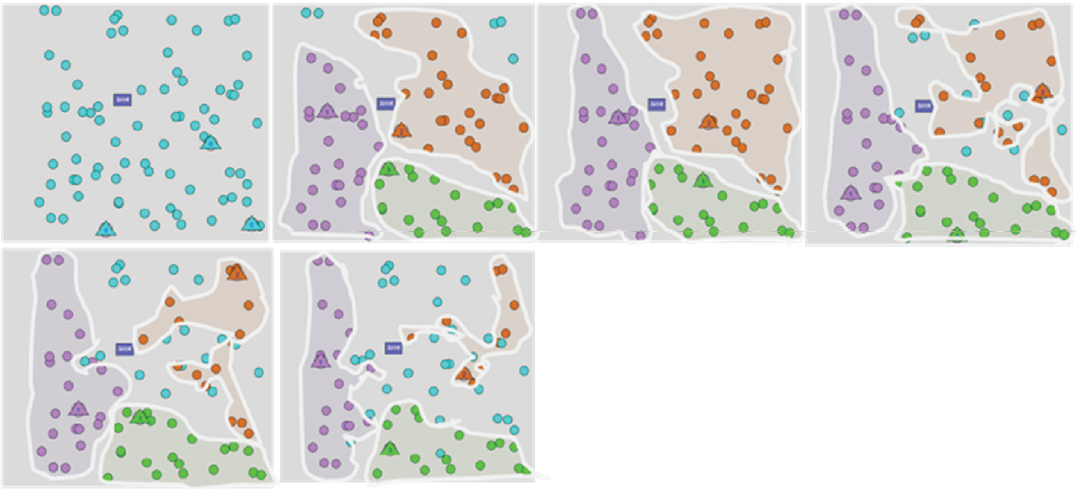


Figure 5. Learning process in a semi-automated network with the sink placed at the centre.

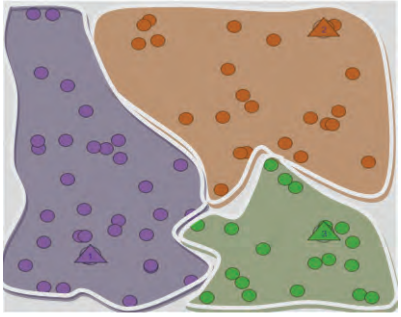


Figure 6. Charging sensors by actor nodes in a network without sink.

Figure 8 shows the result of the number of actors on the sensing network coverage in a network with an

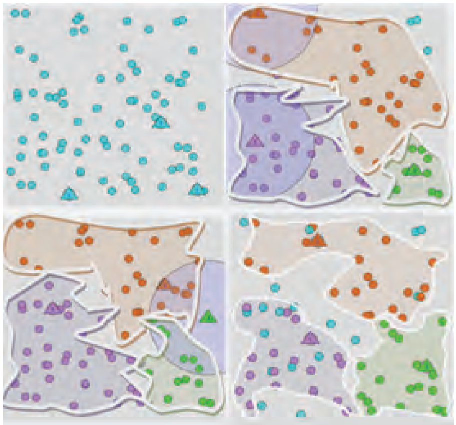


Figure 7. Learning process in an automated network.

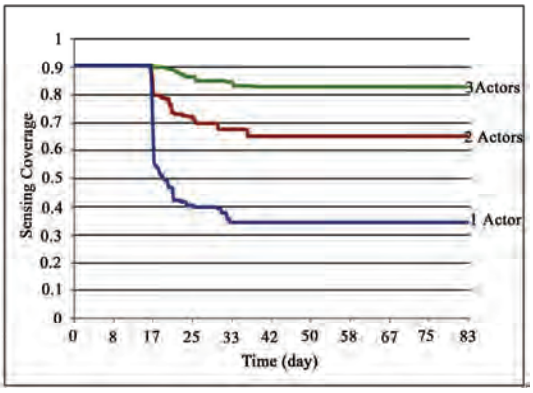


Figure 8. Sensing network coverage in an automated network based on the number of actors (76 sensor nodes).

automated architecture. If we consider the sensing coverage under 50% as a dead network, the network will die with less than 2 actor nodes; otherwise the network will stay alive although some sensor nodes might have died.

Figure 9 and Figure 10 show the impact of the number of actors and semi-automated architecture of the network on network coverage. Moreover, these figures show that the placement of the sink in the centre changes the sensing coverage. It is shown that in a network with semi-automated architecture, wherein the sink is placed in the centre of the network, even 2 actor nodes cannot provide enough sensing coverage; the sensing coverage decreases to less than 50% and the network is considered as a dead network.

5. Conclusions and Future Work

In this paper we introduced a new approach for energy replenishment of sensor nodes by mobile actor nodes in wireless sensor and actor networks. The replenishment starts when local learning of actor nodes is finished. To show the significance of the number of actor nodes withcharging duty on the lifetime of the network, some scenarios were simulated. Simulations showed that a network with 76 sensor nodes scattered randomly in a 1000m×1000m area needs at least 3 actor nodes to stay alive. Although some sensor nodes may die because actor nodes cannot charge them before they are completely discharged, the network still stays alive with a more limited coverage.

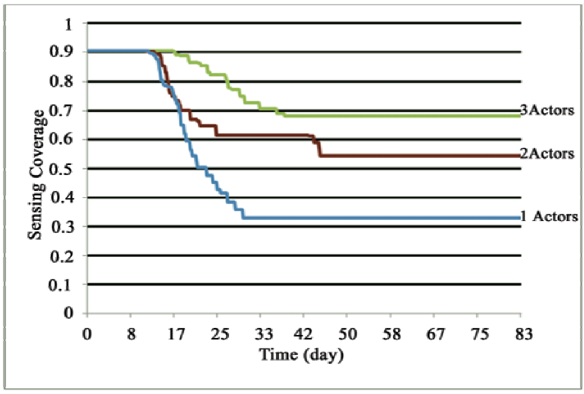


Figure 9. Sensing network coverage in a semi-automated network with the sink placed at the bottom right corner (76 sensor nodes).

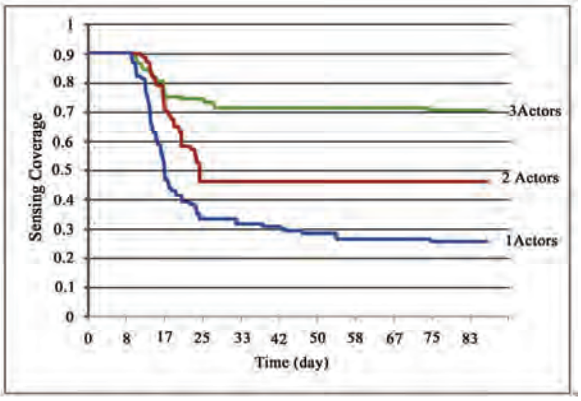


Figure 10. Sensing network coverage in a semi-automated network with the sink placed at the centre (76 sensor nodes).

We are currently studying the case where actor nodes share their learned energy model in order to make more optimized decisions for recharging sensor nodes that prolong the network lifetime (i.e., explicit coordination). When actor nodes learn the energy model of the whole network in addition to their own local learning, they can cooperate to service sensor nodes. This way, the number of sensor nodes recharged can be optimized and the number of dead sensor nodes will decrease especially in networks with semi-automated architecture. We also know that explicit coordination may need more energy consumption in actor nodes, so our desire is a trade-off between implicit and explicit coordination.

Another future work that can increase the performance is to recharge sensor nodes based on their required energy and not fully. In our approach, actor nodes charge sensor nodes completely while they can charge them just according to their real needs that can be extracted from their ratio power, power capacity, current power and time. This way, the time actor nodes spend on each sensor node decreases and they can charge more sensor nodes. The number of movements of actor nodes increases though because sensor nodes require more frequent recharges.

5. References

[1] B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multirobot coordination," IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, pp. 758–768, 2002.

[2] C. Q. Hua and T. S. P. Yum, “Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks,” IEEE-ACM Transactions on Networking, Vol. 16, No. 4, pp. 892–903, 2008.

[3] S. Zhou, R. P. Liu and Y. J. Guo, “Energy efficient networking protocols for wireless sensor networks,” Proceedings of IEEE International Conference on Industrial Informatics, Singapore, pp. 1006–1011, 2006.

[4] M. I. Shukur, L. S. Chyan, and V. V. Yap, “Wireless sensor networks: delay guarantee and energy efficient MAC protocols,” World Academy of Science, Engineering and Technology, Vol. 50, 2009.

[5] M. H. F. Ghazvini, M. Vahabi, M. F. A. Rasid, R. S. A. R. Abdullah, and W. M. N. M. W. Musa, “Low energy consumption MAC protocol for wireless sensor networks,” Proceedings of Second International Conference on Sensor Technologies and Applications, France, pp. 49–54, 2008.

[6] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan and M. Srivastava, “Heliomote: enabling long-lived sensor networks through solar energy harvesting,” Proceedings of 3rd International Conference on Embedded Networked Sensor Systems, California, USA, pp. 309, 2005.

[7] K. Vijayaraghavan and R. Rajamani, “Active control based energy harvesting for battery-less wireless traffic sensors: theory and experiments,” Proceedings of American Control Conference, Washington, USA, pp. 4579–4584, 2008.

[8] L. Mateu, C. Codrea, N. Lucas, M. Pollak, and P. Spies, “Energy harvesting for wireless communication systems using thermogenerators,” Proceeding of the XXI Conference on Design of Circuits and Integrated Systems (DCIS), Barcelona, Spain, 2006.

[9] A. LaMarca, D. Koizumi, M. Lease, S. Sigurdsson, G. Borriello, W. Brunette, K. Sikorski, and D. Fox, “Plantcare: An investigation in practical ubiquitous systems,” Lecture Notes in Computer Science, Vol. 2498, pp. 316–332, 2002.

[10] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, “Studying the feasibility of energy harvesting in a mobile sensor network,” Proceeding of IEEE International Conference on Robotics and Automation, Taiwan, pp. 19–24, 2003.

[11] I. F. Akyildiz and I. H. Kasimoglu, “Wireless sensor and actor networks: research challenges,” Ad Hoc Networks, Vol. 2, No. 4, pp. 351–367, 2004.

[12] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, Y. Zhao, C. Brooks, N. V. Krishnan, S. Neuendorffer, C. Zhong, and R. Zhou, “Visualsense: visual modeling for wireless and sensor network systems,” Technical Memorandum UCB/ ERL M04/08, Electronics Research Laboratory, College of Engineering, University of California, 2004.

***Wireless Sensor Network*, 2010, 2, 129-140**
doi:10.4236/wsn.2010.22018 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Scientific
Research

Finding the Optimal Percentage of Cluster Heads from a New and Complete Mathematical Model on LEACH

A. B. M. Alim Al Islam[1], Chowdhury Sayeed Hyder[2], Humayun Kabir[3], Mahmuda Naznin[4]

[1]*School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA*
[2]*Department of Computer Science and Engineering, Michigan State University, Michigan, USA*
[3,4]*Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh*
E-mail: *abmalima@purdue.edu, hydercho@msu.edu,* {*mhkabir, mahmudanaznin*}*@cse.buet.ac.bd*

Received November 29, 2009; *revised December* 18, 2009; *accepted December* 21, 2009

Abstract

Network lifetime is one of the important metrics that indicate the performance of a sensor network. Different techniques are used to elongate network lifetime. Among them, clustering is one of the popular techniques. LEACH (Low-Energy Adaptive Clustering Hierarchy) is one of the most widely cited clustering solutions due to its simplicity and effectiveness. LEACH has several parameters that can be tuned to get better performance. Percentage of cluster heads is one such important parameter which affects the network lifetime significantly. At present it is hard to find the optimum value for the percentage of cluster head parameter due to the absence of a complete mathematical model on LEACH. A complete mathematical model on LEACH can be used to tune other LEACH parameters in order to get better performance. In this paper, we formulate a new and complete mathematical model on LEACH. From this new mathematical model, we compute the value for the optimal percentage of cluster heads in order to increase the network lifetime. Simulation results verify both the correctness of our mathematical model and the effectiveness of computing the optimal percentage of cluster heads to increase the network lifetime.

Categories and Subject Descriptors: **[Sensor Networks]** – Energy and resource management – Clustering.

Keywords: Wireless Senor Network, Clustering, Mathematical Model Network Lifetime, Energy Consumption Rate, Renewal Reward Process

1. Introduction

With the advent of new technology and low production costs, wireless sensor networks (WSN) prove to be useful in myriad of diversified applications, although its original development was motivated by military applications, such as battlefield surveillance. Most of the WSN applications involve monitoring, tracking, or controlling. Habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring are few examples of such WSN applications. In a typical WSN application, sensor nodes are scattered in a region from where they collect data to achieve certain goals. Data collection may be continuous, periodic or event based process.

Irrespective of data collection process different kinds of management, such as power management, dynamic topology (due to node failure) management, self-configuration management, resource management, and security management are necessary for WSN. Power management deals with the optimum energy usage in order to increase network lifetime. Dynamic topology management dynamically adjusts the topology in case of the death of an existing node or the arrival of a new node. Self configuration management enables the nodes to tune its parameters on the fly. Resource management takes the role to ensure effective resource (CPU and memory) sharing among multiple tasks. Security management guarantees protection against any theft or intrusion in the network.

The importance of power management in WSN lies in the fact that the sensor nodes come with limited powered battery pre-installed. Moreover, the batteries cannot be replaced in the sensor nodes once they are in operation. For these reasons, the algorithms and protocols used in WSN have to be energy efficient. Different techniques are used to achieve energy efficiency like clustering, data compression, dynamic power management etc. Among these, clustering techniques have got wider popularity.

In clustering technique, deployed sensor nodes form some groups called clusters. Each cluster has only one cluster head node, which has the sole responsibility of communicating with the base station. Other nodes in a cluster are called followers. They collect and pass data to the corresponding cluster head. The cluster head of a cluster aggregates its own data with that of its followers. Aggregated data is then sent to one or more sinks. Compared to a sink, a cluster head generally remains closer to the followers. For this reason, it takes less energy to transmit data to the cluster head instead of transmitting directly to the sink. This allows the sensor nodes to conserve more energy and live longer in WSN. There are many clustering techniques [1–7], and [8] available in the literature. LEACH [1] is one of the simple but popular clustering techniques used in WSN. LEACH rotates the cluster head role very effectively among the sensor nodes of a cluster based only on some locally available information. Some modified LEACH [9–12] have also been proposed. LEACH has some parameters that can be tuned to make it more effective. The percentage of cluster heads, which is an input to LEACH algorithm that indicates the overall percentage of sensor nodes that can become cluster head at any time, plays an important role to achieve optimal energy consumption goal in a WSN. However, it is difficult to get the optimum value for this parameter due to the absence of a complete mathematical model on LEACH.

In the mathematical model of [11], total energy consumption in the sensor network is calculated against the transmission of only one frame. In the mathematical model in [13] and [14] total energy consumption in the sensor network is calculated during a single round. Energy consumption during a single round or a frame transmission does not reflect the true energy consumption of the sensor nodes in their lifetime. The long run or expected rate of energy consumption truly reflects the energy consumption of the sensor nodes in their lifetime as the lifetime is inversely related to the long run rate of energy consumption. However, none of the previous LEACH mathematical models [11,13], and [14] has computed long run rate of energy consumption by the sensor nodes, which made them incomplete and not useful. Therefore, it is necessary to derive a mathematical model based on long run rate of energy consumption in the sensor nodes in order to make it complete and useful to compute the tunable parameters, such as the percentage of cluster head. In this paper, we formulate a complete mathematical model based on long run rate of energy consumption in the sensor nodes that uses LEACH clustering algorithm. We use this new mathematical model to find the optimal value of percentage of cluster heads and to tune other LEACH parameters.

In the next section, we briefly describe some related works. We analyze the behavior of a random LEACH node by simulation in Section 3. We devise a complete mathematical model to realize this behavior in Section 4. In Section 5, we prove the correctness of our proposed mathematical model. In Section 6 and Section 7, we conclude the paper and shed some lights on our future work respectively.

2. Related Works

LEACH [1] introduced a simple mechanism for localized coordination and control for cluster set-up and operation. It also introduces the randomized rotation of the cluster heads and the corresponding clusters. LEACH is a self-organizing, and adaptive clustering protocol [1]. It dynamically creates clusters in order to distribute the energy load evenly among all of the sensor nodes. This algorithm needs time synchronization. Cluster heads are randomly rotated during each time interval. The resultant cluster heads directly communicate with the base station.

In LEACH, the lifetime of the network is divided into some discrete, disjoint time intervals. Each interval is again subdivided into some subintervals as shown in Figure 1. LEACH algorithm uses the percentage of cluster head parameter to determine the number of subintervals in an interval. The number of subintervals is the inverse of the value of the above parameter and this value is set a priori to the algorithm. Each subinterval begins with an advertisement phase followed by a cluster set up phase. In the advertisement phase, each node independently decides whether to become a cluster head or not and advertise this information to its neighbors. In the cluster set-up phase, the clusters are organized based on the decisions made in the advertisement phase. Then a steady-state phase follows. In this phase, the followers, *i.e.*, the sensor nodes except cluster heads, sends data to the corresponding cluster head. The cluster heads accumulate and compress the received data with its own data. Cluster heads finally send the compressed data to the base station or sink. In order to minimize cluster establishment overhead, the duration of steady-state phase is kept longer than that of cluster set-up phase.

At the very beginning of an advertisement phase, each node decides whether it wants to become a cluster head for the current round. This decision depends on the value of the percentage of cluster heads as well as the number of times the node has already become cluster head. A node *n* chooses a random number between 0 and 1 and computes

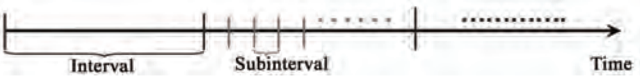


Figure 1. Discrete and disjoint intervals in the whole network lifetime; discrete and disjoint subintervals in an interval.

a threshold $T(n)$ as follows:

$$T(n)=\begin{cases}\dfrac{P}{1-P\times\left(r \bmod \dfrac{1}{P}\right)} & if\ n\in G\\ 0 & otherwise\end{cases}$$

where, P = the percentage of nodes that can become cluster heads at any time (e.g. $P = 0.05$);

$1/P$ = the number of subintervals in an interval;

r = the current subinterval;

G = the set of nodes that have not been cluster heads yet in the current interval.

Comparing the random number with this threshold, a node can be either a cluster head or a follower in any one of $1/P$ subintervals of an interval. If the picked random number is less than the threshold $T(n)$, the node decides to become a cluster head. Otherwise, it decides to become a follower. At the first subinterval of an interval ($r = 0$), each node has a probability P to become a cluster head. The nodes that were cluster heads in the first subinterval cannot be cluster heads in the next ($1/P – 1$) subintervals of the same interval. Thus the probability that the remaining nodes are becoming cluster heads is increasing after each round. After the completion of $1/P$ subintervals, a new interval will start and all the nodes are again equally eligible to become cluster head.

Each node that has chosen itself as a cluster head in the current subinterval, broadcasts an advertisement message to the rest of the nodes in its neighborhood. The non-cluster-head nodes will choose the cluster to which it will belong in this subinterval. This decision is based on the received signal strength of the advertised messages from the nearby cluster head nodes. Assuming symmetric propagation channels, the cluster head whose advertisements have been heard with the largest signal strength will be selected by a non-cluster-head sensor node as its cluster head. A cluster head is chosen randomly in case of a tie.

This algorithm introduced a fairly simple strategy which is more efficient than the direct transmission. LEACH algorithm is also more effective than that of the minimum-transmission-energy (MTE) protocol, where minimum energy route is chosen to transmit data from the sensor nodes to the sink. LEACH algorithm uses the desired percentage of cluster heads as an input parameter. However, it is difficult to find the value of this parameter, for which LEACH will ensure optimum energy consumption. When the number of live sensor nodes becomes very small the number of prospective cluster heads, which is equal to the multiple of the number of live sensor nodes and the desired percentage of cluster heads, will also become very small and sometime it may become even less than one. For example, if the initial number of sensor nodes is 100 and the desired percentage of heads P is 0.05 then the initial number of prospective heads is 5 (=100×0.05). However, with the same P when the number of live nodes becomes less than 20 the number of prospective heads becomes less than one. Under this condition, none of the live sensor nodes can become a cluster head in most of the subintervals by choosing a random number less than the current threshold. In other words, there will be no cluster head available to the sensor nodes to which they can become followers. Rather, all the live sensors will force themselves to become a one member cluster head. In this particular case, the resultant cluster setting will behave like a setting which does not have clustering at all. For this reason, no energy efficiency will be gained.

Despite of these limitations, LEACH is still a widely accepted clustering protocol and a number of its variants have already been proposed for further enhancement in [9–11], and [12]. SEP [9], a LEACH variant, modifies the equation of the threshold, $T(n)$. Deterministic Cluster Head Selection [10], another variant of LEACH also modifies the threshold to accommodate the heterogeneity of residual energy based on some heuristics. LEACH-C, proposed by the original LEACH authors in [11], is a centralized technique which selects the cluster heads based on their positions. It considers uniform distribution of the cluster heads based on their positions and the average residual energy in the network. Adaptive Cluster Head Selection [12], a distributed clustering technique based on LEACH, considers the positions for uniform distribution of cluster heads throughout the network. The availability of many variations on LEACH also indicates its popularity. Actually, LEACH is very popular clustering technique because of its simplicity and effectiveness.

There are some incomplete mathematical models available on LEACH. In [11], a mathematical model is proposed to compute the total energy dissipation in the sensor network for the transmission of a frame. By taking the derivative of the total energy it finds the optimum number of clusters, k_{opt} as—

$$k_{opt}=\frac{\sqrt{N}}{\sqrt{2\pi}}\sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}\frac{M}{{d_{BS}}^2}$$

where, N is the total number of sensor nodes, M is the dimension of the sensor area, d_{BS} is the distance between cluster head and base station, ε_{fs} and ε_{mp} are the amplifier energies.

In [13], a mathematical model is proposed to compute the total energy consumption in the sensor network during a single round. By taking the derivative of the total energy, it also finds the optimum number of clusters, k_{opt} as—

$$k_{opt}=\frac{\sqrt{N}}{\sqrt{\pi}}\sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}\frac{M}{{d_{BS}}^2}$$

In [14], a mathematical model is proposed to calculate the total energy consumption in the sensor network during a single round. It also finds the desired optimum cluster head probability, p_{opt} as—

$$p_{opt} = \frac{1}{2}\sqrt{\frac{\varepsilon_{fs}}{\lambda\left(\varepsilon_{mp} d_{BS}{}^{4} - E_{elec} - E_{DA}\right)}}$$

where, λ is the intensity of homogeneous spatial Poisson process that indicates the sensor node density, E_{elec} is the electronic energy required for coding, modulation, filtering etc. and E_{DA} is the energy required for data aggregation.

However, the lifetime of a sensor node is the inverse of its long run rate energy consumption. Therefore, in order to achieve elongated network lifetime of a sensor node, the long run rate of energy consumption must be more significant than the total energy consumption in an interval or for one frame transmission. Unfortunately, none of the mathematical models mentioned above consider the long run rate of energy consumption by a sensor node in the network. Moreover, none of these models consider the situation, in which all the sensor nodes in the network, can pick a random number higher than its respective threshold and become one member cluster head all together. For these reasons, all the previous mathematical models remain incomplete and less useful. We deduce a complete mathematical model based on the long run rate of energy consumption and considering the above mentioned situation in Section 4. In order to deduce the mathematical model, we need to observe the inherent behavior of a random LEACH node in terms of long run energy consumption.

3. Observed Behavior of A Random Leach Node

In this section, we analyze the behavior of a random LEACH node in terms of long run energy consumption by simulation in order to find a trend in the behavior. In our simulation, expected energy consumption rate is calculated against different percentage of cluster heads.

A visual C++ program is developed for the simulation. We use following network setting as shown in the simulation runs:

- The dimension of sensor area is 200×200.
- Total number of heterogeneous sensor nodes in the network is 100.
- Initial energies of the sensor nodes are uniformly distributed between 1J and 5J.
- The sensor nodes are uniformly distributed over the sensor area.
- The base station is located at position (1500, 100).

We show the network setting in Figure 2. We plot the energy consumption rate by a randomly chosen LEACH node against the percentage of heads in Figure 3 from the results of our simulation runs.

According to the graph in Figure 3,

1) Energy consumption rate initially decreases very sharply with the increase of the percentage of cluster heads and

2) There is an optimal point for which energy consumption rate is the lowest. After the optimal point, the energy consumption rate increases with the increase of the percentage of cluster heads. In our simulation runs, this optimal point is (0.057, 0.003433).

The pattern of the graph matches with that of the normalized energy dissipation versus the percentage of cluster heads in figure of [1]. We formulate a mathematical model complying with this behavior of a LEACH node in the next section. In Section 5, we compare the trends observed in this section with that from our mathematical model in order to proof that our mathematical model is correct and explain the reasons of these trends.

4. Proposed Mathematical Model for Leach

The primary reason behind the clustering technique is to reduce the rate of energy consumption so that the life of a wireless network elongates. Popular clustering techniques leach [1] and its variants [9] and [10] can achieve this goal if their parameters are set suitably. As of today, some heuristics are used to choose the values for leach parameters. A complete mathematical model can serve better than the heuristics to achieve an optimal rate of energy consumption in WSN. Moreover, a mathematical model can provide the ways to tune other application specific

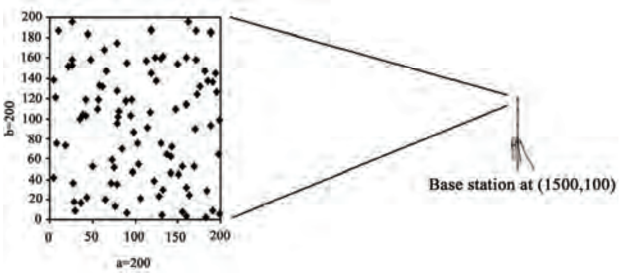


Figure 2. Uniformly distributed sensor network with a distant base station.

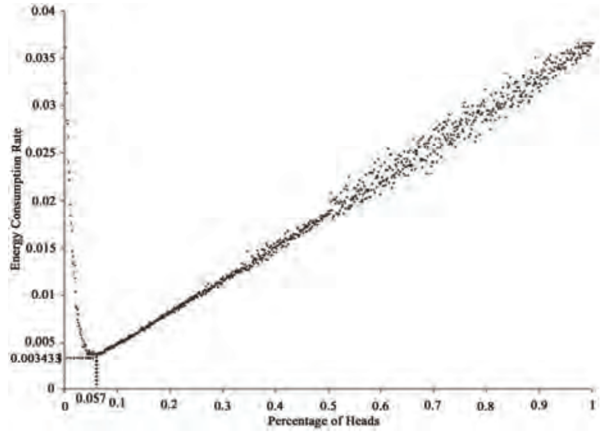


Figure 3. Energy consumption rate against different percentage of heads for a randomly chosen leach node.

parameters. We derive a complete mathematical model of energy consumption rate of LEACH in this section.

4.1. Preliminaries

We use some basic assumptions about the sensor nodes and the network settings while developing our mathematical model for LEACH. We also use some base models in the formulation of our mathematical model. After describing the basic assumptions, we describe the base models.

1) *Assumptions*

In order to make our mathematical model fully aligned with LEACH algorithm [1], we use following assumptions about the sensor nodes and network settings:

- Nodes do not have any location information. This assumption makes our model suitable for the applications (for example, environmental monitoring) that use randomly deployed sensor nodes.
- All nodes can directly reach the BS. This assumption ensures applicability of our model for single-hop clustering [7].
- The propagation channel is symmetric. Symmetry of propagation channel implies that energy required to transmit a message from node A to node B is the same as energy required to transmit a message from node B to node A for a given Signal to Noise Ratio (SNR).

2) *Base Models*

Heinzelman proposed an energy model namely first order radio model for energy consumption in a wireless network in [15]. Like other research works [1], [9], and [10], we use this first order radio model to compute the expected energy consumption rate in sensor networks.

Energy consumption, due to the reception and the transmission of data in a sensor network, is a stochastic process. Therefore, we use Renewal Reward Process [16], [17], a widely known stochastic process, to capture the nature of energy consumption due to data transmission and reception by a sensor node. In the following subsections, we briefly describe these two base models.

3) *Energy Model: Heinzelman First Order Radio Model*

A receiving sensor node consumes the energy while receiving and processing a message. The amount of this energy is proportional to the number of bits in the message under processing. For example, if the message contains k bit and the energy per bit is E_{elec} Joules, then the energy used to run the circuitry will be ($E_{elec}\times k$) Joules. Therefore, the energy consumed by a receiving node to receive and process a k-bits message is,

$$E_{R_X}(k) = (E_{elec} \times k) \tag{1}$$

The energy needs to send k bit message over a distance d is ($\in_{amp}$ x $k\times d^{\lambda}$) Joules, where $\in_{amp}$ is the energy constant for the radio transmission and λ is the path loss exponent. While transmitting, a sensor node needs energy to process as well as to send the message. Therefore, the total energy consumed by a transmitting node to send a k-bits message over distance d is,

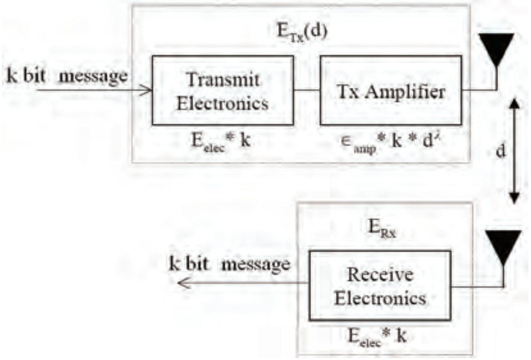


Figure 4. Heinzelman first order radio model.

$$E_{T_X}(k,d) = (E_{elec} \times k) + (\in_{amp} \times k \times d^{\lambda}) \tag{2}$$

We consider Heinzelman's first order model (*i.e.* $\lambda = 2$) for transmission of data in low noise environment. This model is shown in Figure 4. We consider $\lambda = 4$ for high noise environment.

4) *Renewal Reward Process*

A Renewal Process is special counting process $N(t)$ which counts the number of events up to time t and the inter-arrival times of the events are independent and identically distributed (*iid*) random variables. The expected value of inter-arrival times is in between zero and infinity. A Renewal Reward Process [16,17] is a Renewal Process such that there are some rewards for each of the inter-arrival times. These rewards are also independent and identically distributed (*iid*) random variables. If X_i is the i^{th} inter-arrival time and R_i is reward for the inter-arrival time X_i, the total reward earned up to time t will be:

$$R(t) = \sum_{i=1}^{N(t)} R_i \tag{3}$$

According to Renewal Reward Theorem, the rate of reward will be:

$$\lim_{t\to\infty} \frac{R(t)}{t} = \frac{E(R)}{E(X)} \tag{4}$$

This means that the rate of reward is equal to the ratio between the expected reward in a single inter arrival time $E(R)$ and the expected inter-arrival time $E(X)$ in the long run. The theorem can be proved based on Strong Law of Large Numbers [18] and is out of scope of the this paper. In stochastic process, the inter arrival time is also called a cycle.

4.2. Proposed Mathematical Model

As the part of our mathematical analysis, we calculate the expected energy consumption rate following the renewal-reward process. We consider the energy consumed by the sensor as the reward. Then the long run rate of reward will essentially be the long run rate of energy consumption. However, to map this problem with renewal-reward process perfectly, we have to define cycle in such a way that both the reward and the cycle can be treated as *iid* random variables.

According to LEACH algorithm, in the first subinterval of an interval each live sensor node will have some non zero probability to become cluster head. However, in the other subintervals a sensor node has zero probability to become cluster head, if it became a cluster head in the first or any other previous subinterval. It must be a follower in all other subsequent subintervals in the same interval. We define a cycle be the difference between two consecutive subintervals in which a sensor node becomes cluster head. Cluster establishment is probabilistically done in consecutive subintervals. Hence, the cycle or inter-arrival time is an integer number and *iid* random variable. Energy consumption in each interval is also an *iid* random variable. These definitions of cycle and reward map our problem to a renewal-reward process perfectly. Thus, the long run rate of reward Equation (4) gives the expected energy consumption rate. We need to compute $E(R)$ and $E(X)$ to derive the energy consumption rate. We define following parameters for this purpose—

1) P is the desired percentage of cluster heads,
2) s is the number of subintervals in an interval, therefore $s = 1/P$,
3) P_h is the probability of becoming cluster head of a follower node at the start of any subinterval,
4) P_h' is the probability of becoming cluster head of a cluster head node at the start of a subinterval in the next interval,
5) Φ_0 is the probability of becoming cluster head of a sensor node at the start of any subinterval,
6) T_h is the currently considered threshold value.
7) N is the total number of sensor nodes in the network.
8) $a \times b$ is the two dimensions of rectangular sensor area.

a) *Calculation of E(X)*

We compute expected cycle length, $E(X)$, of Equation (4) in this section. At the beginning of each subinterval new cluster heads are selected and new clusters are generated. Each sensor will generate a random number between 0 and 1 and compares it to a predefined threshold value T_h. If the random number is less than the threshold, the sensor node becomes cluster head. Otherwise, the sensor node acts as a follower. We can show that the transitions between two states (heads: h and follower: f) of a sensor node while changing the subinterval in an interval by following matrix:

$$\begin{array}{c c} & \begin{array}{cc} h & f \end{array} \\ \begin{array}{c} h \\ f \end{array} & \begin{pmatrix} 0 & 1 \\ P_h & 1-P_h \end{pmatrix} \end{array}$$

If the interval is changed then the probability of becoming head while changing the subinterval will be the same irrespective of the previous state. Therefore, we can show the transitions between two states of a sensor node while changing the subinterval as well as the interval by following matrix:

$$\begin{array}{c c} & \begin{array}{cc} h & f \end{array} \\ \begin{array}{c} h \\ f \end{array} & \begin{pmatrix} P_h' & 1-P_h' \\ P_h & 1-P_h \end{pmatrix} \end{array}$$

Above behaviors of sensor nodes in LEACH can be shown by the transition diagrams in Figure 5.

A sensor node can become cluster head at the start of the first subinterval of a new interval based on the picked random number and the threshold. This decision does not depend on whether it was cluster head or follower in the last subinterval of previous interval. In this case, the probability of a follower to become a cluster head and the probability of a cluster head to remain cluster head are same, *i.e.*, $P_h' = P_{h.}$.

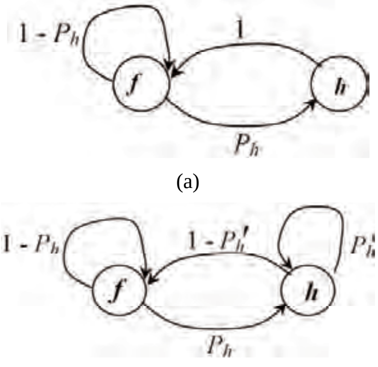
The number of subintervals in an interval is s. Therefore, a sensor node remains in the same interval up to (s-1) subinterval transitions and moves to the next interval only at the last subinterval transition. From this observation, we can say that the probability of remaining in the same interval is equal to (s-1)/s and the probability of changing the interval is equal to 1/s.

We combine these probabilities with their corresponding transition matrices in order to capture the whole scenario.

$$\frac{s-1}{s}\begin{pmatrix} 0 & 1 \\ P_h & 1-P_h \end{pmatrix} + \frac{1}{s}\begin{pmatrix} P_h & 1-P_h \\ P_h & 1-P_h \end{pmatrix} = \begin{pmatrix} \frac{P_h}{s} & \frac{s-P_h}{s} \\ P_h & 1-P_h \end{pmatrix} \quad (5)$$

Hence, the combined transition matrix becomes as follows:

1) The probability of a cluster head to remain cluster head, P_{hh} at the start of any subinterval is P_h/s.
2) The probability of a follower to become a cluster head, P_{hf} at the start of any subinterval is P_h.



(a)

(b)

Figure 5. State transition of a node while (a) Changing subinterval without changing Interval; (b) Changing subinterval as well as the interval.

3) The probability of a cluster head to become a follower, P_{fh} at the start of any subinterval is 1-P_h/s.
4) The probability of a follower to remain follower, P_{ff} at the start of any subinterval is 1-P_h.

Now, we can compute the probability of becoming a cluster head, Φ_0, at the start of any subinterval by summing up the first two values P_{hh} and P_{hf} as follows—

$$\Phi_0 = \left(\frac{P_h}{s}\right) + P_h = (s+1) \times \left(\frac{P_h}{s}\right) \tag{6}$$

We can say that the expected value of the cycle is reciprocal of the probability of becoming cluster head, Φ_0, at the start of any subinterval, *i. e.*—

$$E(X) = \frac{1}{\Phi_0} = \frac{s}{(s+1) \times P_h} \tag{7}$$

In order to compute Ph we need to consider following two conditions—

1) A node can be a cluster head if the picked random number is lower than the threshold. In LEACH, the threshold is maintained in a way such that the mean value of the threshold becomes the percentage of sensor nodes to be elected as the cluster heads in the network. Hence, the probability of becoming cluster head in this way is equal to the said percentage, *i.e.*, $P_{h1} = P$.

2) If none of the nodes pick the random number less than that of the threshold, all nodes act as one-member cluster head. The probability of becoming one-member cluster head in this way is, $P_{h2} = (1 - P)^N$.

Therefore, the ultimate probability of becoming a cluster head, P_h while changing subinterval in an interval will be $P + (1 - P)^N$. Hence, expected cycle length $E(X)$ can be calculated by substituting P_h from Equation (7).

b) *Calculation of E*(*R*)

We compute expected reward (energy consumption), $E(R)$, of Equation (4) in this section. Energy consumption by a sensor node as a cluster head differs from that of a sensor node as a follower. Let—

1) H be the amount of energy consumed by a cluster head in a single subinterval and
2) F be the amount of energy consumed by a follower in a single subinterval.

In a cycle, the expected number of subintervals in which a sensor node remains a follower is ($E(X)$ – 1) and the expected number of subintervals in which a sensor node remains a cluster head is 1. Therefore, the amount of energy consumed by a sensor node in a single cycle is—

$$E(R) = (E(X) - 1) \times E(F) + E(H) \tag{8}$$

Here, $E(F)$ and $E(H)$ are the expected values of energy consumed by a follower and a cluster head, respectively, in a single subinterval. $E(X)$ has already been calculated in previous sub-section. We need to calculate $E(F)$ and $E(H)$ in order to find $E(R)$ of Equation (8).

1) *Calculation of E*(*F*)

We can compute the expected value of energy, $E(F)$, consumed by a follower in a single subinterval using Heinzelman first order radio model [15]. Being a follower, a sensor node consumes energy only for transmitting. According to Heinzelman First Order Radio Model, the total energy to transmit a *k*-bit message by a follower over distance *X* is—

$$E(F|X = x) = (E_{elec} \times k) + (\in_{amp\_F} \times k \times x^2) \tag{9}$$

If *f(x)* is the distribution function of the distance *X* of a follower to its nearest cluster head, the energy consumption by a follower will be—

$$\begin{aligned} E(F) &= \int E(F|Distance\ \ to\ the\ nearest\ \ cluster\ \ head = x) f(x)dx \\ &= \int ((E_{elec} \times k) + (\in_{amp\_F} \times k \times x^2)) f(x)dx \\ &= (E_{elec} \times k) + (\in_{amp\_F} \times k) \int x^2 f(x)dx \end{aligned} \tag{10}$$

Now, we calculate the distribution function of the distance, *f(x)*—

$$f(x) = \frac{d}{dx} P(X \le x) \tag{11}$$

There might be several cluster heads at the nearest distance. Therefore,

$$\begin{aligned} P(X \le x) &= P(at\ least\ 1\ cluster\ head\ is\, at\ distance\ of\ \ x\ or\ less\ than\ x) \\ &= 1 - P(no\ cluster\ head\ is\, inside\ the\ area\ with\ radius\ x) \\ &= 1 - P(all\ cluster\ heads\ are\ outside\ the\ area\ with\ radius\ x) \end{aligned} \tag{12}$$

Now, if the number of cluster heads is N_c then,

$$\begin{aligned} & P(all\ cluster\, heads\, are\, outside\, the\ area\ with\ radius\ x) \\ & = \sum_{n=1}^{N} P(all\ cluster\, heads\, are\, outside\, \pi x^2 | N_c = n) P(N_c = n) \\ & = \sum_{n=1}^{N} \binom{N}{n} \left(\frac{s+1}{s} P_h\right)^n \left(1 - \frac{s+1}{s} P_h\right)^{N-n} \left(1 - \frac{\pi x^2}{ab}\right)^n \end{aligned} \tag{13}$$

Therefore, we can calculate the distribution function of the distance, *f(x)* as follows—

$$\begin{aligned} f(x) &= \frac{d}{dx} P(X \le x) \\ &= \sum_{n=1}^{N} \binom{N}{n} \left(\frac{s+1}{s} P_h\right)^n \left(1 - \frac{s+1}{s} P_h\right)^{N-n} \frac{2n\pi x}{ab} \left(1 - \frac{\pi x^2}{ab}\right) \\ &= \sum_{n=1}^{N} g(n) \times x \times \left(1 - \frac{\pi x^2}{ab}\right)^{n-1} \end{aligned} \tag{14 }$$

where,

$$g(n) = \binom{N}{n} \left(\frac{s+1}{s} P_h\right)^n \left(1 - \frac{s+1}{s} P_h\right)^{N-n} \frac{2n\pi}{ab}$$

Let,

$$\begin{aligned} I &= \int x^2 \times f(x)dx \\ &= \sum_{n=1}^{N} \int g(n) \times x^3 \times \left(1 - \frac{\pi x^2}{ab}\right)^{n-1} dx \end{aligned} \tag{15}$$

After solving the integration at the right side of the above equation, we get—

$$I = \frac{1}{2}\left(\frac{ab}{\pi}\right)^2 \sum_{n=1}^{N} g(n)\left(\frac{y^{n+1}}{n+1} - \frac{y^n}{n}\right) \tag{16}$$

where,

$$y = 1 - \frac{\pi x^2}{ab}$$

The subtracted value of y indicates the proportion between two areas of Figure 6. Here, the first area is the area inside the circle with the center at the sensor node under consideration and the radius equal to the distance from the sensor node to its nearest cluster head. The second area is the total area covered by all the sensor nodes. If the cluster head position coincides with that of the node, we get the lower limit of x and $\frac{\pi x^2}{ab}$ equal to zero. In this case, y value becomes 1. If the cluster head is positioned at a position such that the first area fully covers the second area, we get the higher limit of x and $\frac{\pi x^2}{ab}$ equal to 1. In this case, y value becomes 0.

Therefore, the integrated value with the limits of y is—

$$[I]_1^0 = \frac{1}{2}\left(\frac{ab}{\pi}\right)^2 \sum_{n=1}^{N} g(n)\left(\frac{1}{n} - \frac{1}{n+1}\right) \tag{17}$$

Combining Equations (10) and (17), we get the expected value of energy consumption of a sensor node as a follower in a subinterval as follows—

$$E(F) = (E_{elec}\, k) + (\in_{amp\_F}\, k)\,\frac{1}{2}\left(\frac{ab}{\pi}\right)^2 \sum_{n=1}^{N} g(n)\left(\frac{1}{n} - \frac{1}{n+1}\right) \tag{18}$$

2) *Calculation of E(H)*

We can compute the expected value of energy, $E(H)$, consumed by a cluster head in a single subinterval using Equation (2). The cluster head aggregates and compresses the data from its followers with its own data before sending them to the base station. Therefore, the actual number of bits sent by a cluster head is less than the summation of the numbers of the bits of all the messages those it handles. Let, γ be the compression ratio. If there are N_f followers and each sensor node generates k bit message, the energy consumption by a cluster head will be:

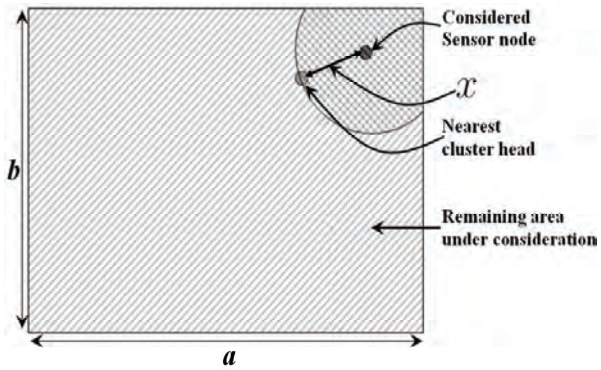


Figure 6. *y* is the ratio between two areas. First one is the remaining area under consideration and the second one is the total area under consideration.

$$E(H|N_f) = (2N_f+1)k(E_{elec}+E_{DA}) + (N_f+1)\in_{amp\_H} k\gamma {d_{BS}}^{\lambda} \tag{19}$$

where, d_{BS} is the distance between the cluster head and base station. Now,

$$\begin{aligned} E(H) &= \sum_{i=0}^{N-1} E(H|N_f=i)\,P(N_f=i) \\ &= \sum_{i=0}^{N-1}\left[\left\{(2i+1)k(E_{elec}+E_{DA}) + (i+1)\in_{amp\_H} k\gamma {d_{BS}}^{\lambda}\right\}\times P(N_f=i)\right] \end{aligned} \tag{20}$$

Since N_c is the total number of cluster heads we can write,

$$P(N_f=i) = \sum_{n=1}^{N-i} P(N_f=i|\, N_C=n)\,P(N_C=n) \tag{21}$$

Here,

$$P(N_C=n) = \binom{N}{n}\Phi_0^n(1-\Phi_0)^{N-n} \tag{22}$$

and

$$P(N_f=i|\, N_C=n) = \binom{N-n}{i}P(A)^i(1-P(A))^{N-n-i} \tag{23}$$

Here, A is an event that ensures the considered cluster head is the nearest cluster head to a follower. If the location of the cluster head is (x_h, y_h) and the location of the follower is (x, y), we can write

$$P(A) = \frac{\pi r^2}{ab}p_a\left(1-\frac{\pi r^2}{ab}p_a\right)^{n-1} \tag{24}$$

where, $r = \sqrt{((x - x_h)^2 + (y - y_h)^2)}$ and p_a is the percentage of the circular area (centered at the follower and with radius r) falls within the area covered by the sensor network. Let,

$P(A)=h(r, n)$. Now, combining Equations (21–23), we get—

$$P(N_f=i) = \sum_{n=1}^{N-i}\left[\binom{N-n}{i}h(r,n)^i(1-h(r,n))^{N-n-i}\binom{N}{n}\Phi_0^n(1-\Phi_0)^{N-n}\right] \tag{25}$$

Combining Equations (6), (20) and (25), we get—

$$\begin{aligned} E(H) \;=\; &\sum_{i=0}^{N-1}\Bigg[\left\{(2i+1)k(E_{elec}+E_{DA}) + (i+1)\in_{amp\_H} k\gamma {d_{BS}}^{\lambda}\right\} \\ &*\sum_{n=1}^{N-i}\left\{\binom{N-n}{i}h(r,n)^i(1-h(r,n))^{N-n-i}\binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n\left(1-\frac{s+1}{s}P_h\right)^{N-n}\right\}\Bigg] \end{aligned} \tag{26}$$

c) *Energy Consumption Rate*

Combining Equations (7), (8), (18) and (26),, we can get energy consumption rate as follows—

$$\begin{aligned}\lim_{t\to\infty}\frac{R(t)}{t} &= \frac{E(R)}{E(X)}\\ &=\left(1-\frac{s+1}{s}P_h\right)\times\left[\left(E_{elec}k\right)+\left(\in_{amp\_F}k\right)\frac{1}{2}\left(\frac{ab}{\pi}\right)^2\sum_{n=1}^{N}g(n)\left(\frac{1}{n}-\frac{1}{n+1}\right)\right]\\ &\quad+\frac{s+1}{s}P_h\sum_{i=0}^{N-1}\left[p(i)\times\sum_{n=1}^{N-i}q(i,n)\right]\end{aligned} \qquad (27)$$

where,

$$p(i) = (2i+1)k\left(E_{elec}+E_{DA}\right) + (i+1)\in_{amp\_H} k\,\gamma\, d_{BS}{}^{\lambda}$$

and,

$$q(i,n)=\binom{N-n}{i}h(r,n)^i\left(1-h(r,n)\right)^{N-n-i}\binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n\left(1-\frac{s+1}{s}P_h\right)^{N-}$$

Equation (27) concludes the formulation of our mathematical model. This equation evaluates the expected energy consumption rate in a wireless sensor net- work. The optimal number of cluster heads can be determined using this equation by plotting various expected energy consumption rate against different percentage of cluster heads. The percentage of cluster heads at the lowest expected energy consumption rate gives the optimal percentage of cluster heads. In the next section, we prove the correctness of our analytical model by simulation results and also find the optimal percentage of cluster heads.

5. Simulation Results and Validation of Analytical Model

Like [11,13,14], we cannot find the optimal value of percentage of cluster heads by taking derivative of the equation of long run rate of energy consumption as there is no closed form of the derivative of Equation (27). Therefore, to verify the correctness of our mathematical model, we conduct simulation runs with the previously described network settings and parameters. We use the following values of the parameters in the simulation runs to be aligned with the simulations of the research works in [1,10,19]—

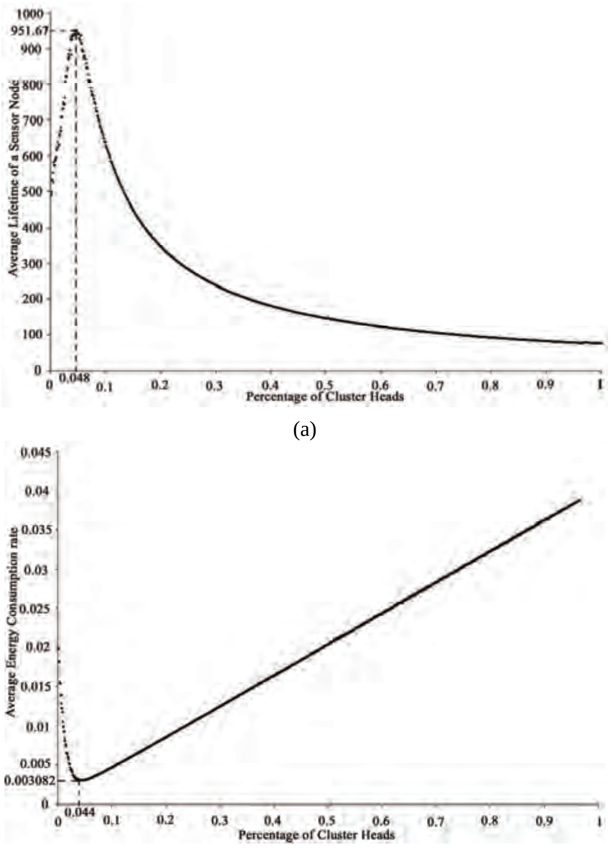
1) The amount of energy per bit to run sensor node circuitry, E_{elec} is 5 x 10^{-8};
2) Both the values of energy constants, $\in_{amp\_F}$ and $\in_{amp\_H}$ for radio transmission, is 1 x 10^{-10};
3) A sensor node generates 0 to 50 units data in an interval;
4) Each data unit contains 8 bits data;
5) The probability that a message successfully arrives at its destination is 90%. Therefore, the communications are low noise and thus λ=2.

We first calculate the lifetime of a sensor node as the number of subintervals it remains alive. We also calculate the energy consumption rate of a sensor node as the ratio between total energy and lifetime. Then, we calculate average lifetime and average energy consumption rate of the sensor nodes in the network from ten simulation runs and plot them in Figure 7(a) and Figure 7(b) accordingly. Figure 7(b) has the similar pattern that has been found in Figure 3. The optimal point is (0.048, 951.67) in Figure 7(a) and (0.044, 0.003082) in Figure 7(b). Apparently, the optimal percentage of cluster heads in both the graphs should be equal. However, in the simulation runs total energy of a sensor node is its initial energy which is uniformly distributed between 1J and 5J. Therefore, total energies of the sensor nodes are different. As a result, optimal percentages of cluster heads in Figure 7(a) and Figure 7(b) are also different. These two percentages of cluster heads (0.044 and 0.048) roughly indicate the range in which the actual optimal percentage of cluster heads should lie.

Before validating our mathematical model against the graphs found from simulation runs, we briefly illustrate the reason behind the trend in the graph of average energy consumption rate against the percentage of cluster heads in Figure 3 and Figure 7(b). The percentage of cluster heads P reflects the expected value of the threshold. When the value of P is close to 0, the probability of the random numbers picked by the sensor nodes in the network to be less than the threshold becomes very low. In this situation, most of the sensor nodes will frequently become one member cluster head. Therefore, the expected energy consumption rate becomes high. As the value of P increases, the probability of frequently becoming one member cluster head for the sensor nodes decreases, *i.e.*, the probability of becoming follower increases and the expected energy consumption rate decreases. Though the probability of becoming a one member cluster head decreases with the increase in P after a certain point this does not help that much to reduce the ultimate energy consumption rate. After that point the increase in the percentage of cluster heads increases the probability of the sensor nodes to become regular cluster head. This is superseding the energy savings by not becoming one member cluster head. For this reason, as the value of the percentage of cluster heads increases the ultimate energy consumption rate increases.

We plot the energy consumption rate against different percentage of cluster heads P using Equation (27) of our mathematical model for the same network settings and parameters in Figure 8(a). For high noise data communication between cluster head and base station λ=4 is applicable. The resultant graph with λ=4 is also shown in Figure 8(b). In both the cases, the resultant graphs have the same trend which perfectly matches the trend found in Figure 7(b) by simulation. Moreover, the optimal point is (0.044, 0.003082) in Figure 7(b) and is (0.045, 0.003733) in Figure 8(b). These two optimal points match very closely.

The value of the probability of becoming cluster head of a sensor node at the start of any subinterval Φ_0 must not exceed 1 and the value of Φ_0 can be computed from by Equation (6). According to Equation (6), if the value of P



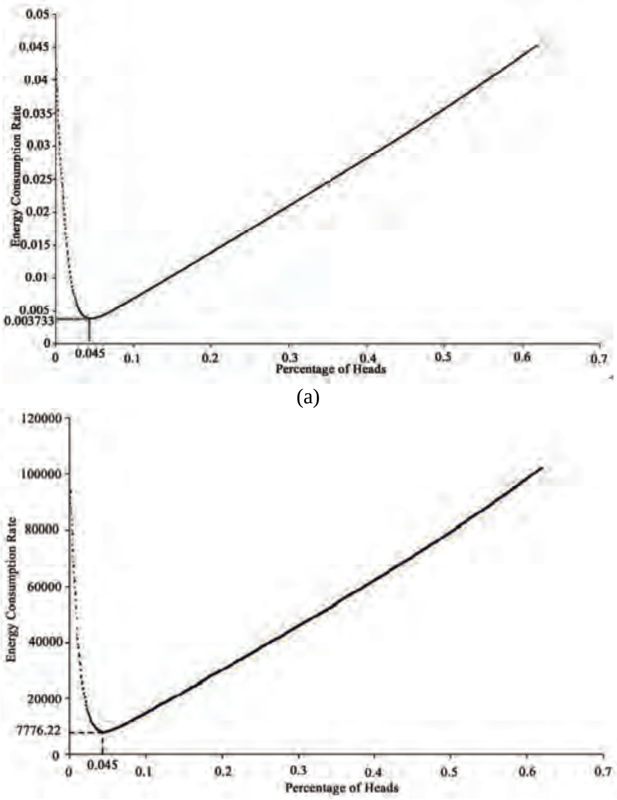
(a)

(b)

Figure 7. Average Behaviors for Different Percentage of Heads over the Whole Network. (a) Average Lifetime against Different Percentage of Cluster Heads; (b) Average Energy Consumption Rate against Different Percentage of Cluster Heads.

exceeds 0.61 then the value of Φ_0 will exceed 1. In order to avoid this, we plot the graph against the percentage of cluster heads P up to 0.61. The graph obtained from our mathematical model also shows the same pattern that has been found in Figure 3 and Figure 7(b). Energy consumption rate also decreases very sharply at the beginning with the increase of the percentage of cluster heads. The value of percentage of cluster heads at the optimal energy consumption rate is 0.045 both in Figure 8(a) and Figure 8(b). The value of percentage of cluster heads at the optimal energy consumption rate is 0.044 in Figure 7(b). These two values are almost equal. Moreover, the optimal percentage of cluster heads in Figure 8(a) and Figure 8(b) is in the range [0.044, 0.048], which was found from the two graphs in Figure 7. The energy consumption rate also increases with the increase of the percentage of cluster heads beyond the optimal point in both Figure 7(b) and Figure 8. These results clearly validate the correctness of our mathematical model.

We also find the optimal percentage of cluster heads from the previous mathematical models of LEACH. We take the value of the distance between a cluster head and the base station, dBS to be 1400 assuming a typical cluster head at the center of the sensor area. We get the values of optimal percentage of cluster heads as 0.00041, 0.00058 and 0.0000051from [11,13,14] accordingly. From Figure 7(b), it is obvious that all of these values cause larger energy consumption than that causes by the value found from our mathematical model. We also compare these values against the optimal percentage of cluster heads found from the mathematical model in terms of network stability period and network lifetime in Figure 9 and Figure 10



(a)

Figure 8. Energy Consumption Rate against Different Percentage of Heads according to the Mathematical Model of LEACH (b). (a) λ = 2; (b) λ = 4.

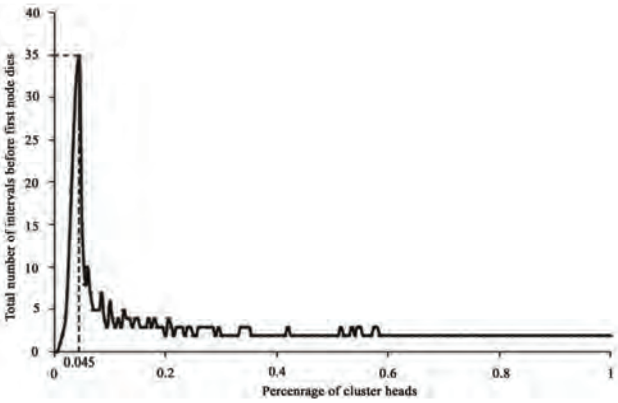


Figure 9. Network stability period against different percentage of heads in LEACH.

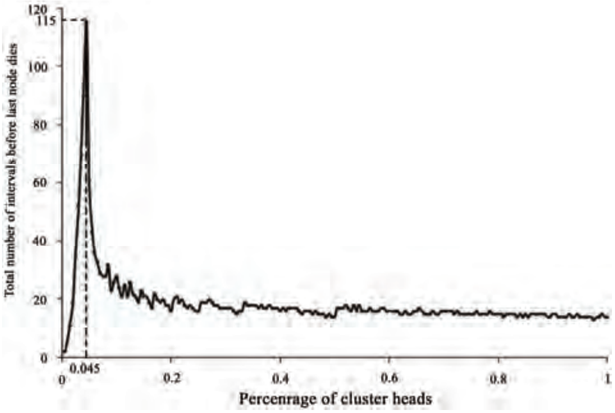


Figure 10. Network lifetime against different percentage of heads in LEACH.

accordingly. Network stability period is the number of intervals passed before the death of the first node and network lifetime is the number of intervals passed before the death of the last node. We plot the network stability period against different percentage of cluster heads in Figure 9. Here, we find the maximum network stability period at the point (0.045, 35). We also plot the network lifetime against different percentage of cluster heads in Figure 10 and find the maximum network lifetime at the point (0.045, 115). In both cases, the optimal percentage of cluster heads found from our mathematical model perfectly matches to the optimal points. However, all the optimal percentages of cluster heads found from the previous mathematical models give significantly lower values for both stability period and network lifetime.

6. Conclusions

In this paper, we devised a mathematical model on LEACH protocol, a widely accepted clustering protocol for WSN. For the verification of our mathematical model, we have simulated a WSN with a random setting. We have applied our mathematical model on the same network setting. We have plotted two graphs of energy consumption rate versus the percentage of heads, one for each of the above cases. The conformity between these two graphs ensures the correctness of our mathematical model. We have also showed that we can find the optimal percentage of cluster heads, for which the energy consumption rate would be the lowest, from the graph of the mathematical model.

7. Future Works

In this paper, we have proposed a mathematical model on the original LEACH, which does not consider the heterogeneity of the sensor nodes in terms of the residual energy and some other problems mentioned in Section 2. In our future work, we will modify basic LEACH algorithm and our mathematical model in order to incorporate the above heterogeneity of the sensor nodes and to overcome problem exists in the original LEACH algorithm.

8. References

[1] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocol for wireless microsensor networks,” In Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, Vol. 2, pp. 10, January2000.

[2] S. Lindsey and C. S. Raghavendra, “PEGASIS: power-efficient gathering in sensor information systems,” Aerospace Conference Proceedings, Vol. 3, pp. 1125–1130, 2002.

[3] L. Li, S. Dong, and X. Wen, “An energy efficient clustering routing algorithm for wireless sensor networks,” The Journal of China Universities of Posts and Telecommunications, Vol. 3, No. 13, pp. 71–75, September 2006.

[4] Y. Sangho, H. Junyoung, C. Yookun, and H. Jiman, “PEACH: power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks,” Com- puter communications, Vol. 30, pp. 2842–2852, 2007.

[5] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, “Optimal energy aware clustering in sensor networks,” SENSORS Journal, Vol. 7, No. 2, pp. 258–269, 2002.

[6] H. Chan and A. Perrig, “ACE: an emergent algorithm for highly uniform cluster formatio,” Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Vol. 2920, pp. 154–171, February 2004.

[7] O. Younis and S. Fahmy, “Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach,” Proceedings of IEEE Infocom, Vol. 1, pp. 640, 2004.

[8] J. Y. Cheng, S. J. Ruan, R. G. Cheng, and T. T. Hsu, “PADCP: poweraware dynamic clustering protocol for wireless sensor network,” International Conference on Wireless and Optical Communications Networks, pp. 6, 13 April 2006.

[9] G. Smaragdakis, I. Matta, and A. Bestavros, “SEP: a stable election protocol for clustered heterogeneous wireless sensor networks,” In the Proceedings of the International Workshop on SANPA, 2004.

[10] M. Haase and D. Timmermann, “Low energy adaptive clustering hierarchy with deterministic cluster-head selection,” 4th International Workshop on Mobile and Wireless Communications Network, pp. 368–372, 2002.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, pp. 660–670, October 2002.

[12] C. Nam, H. Jeong, and D. Shin, “The adaptive cluster head selection in wireless sensor networks,” IEEE International Workshop on Semantic Computing and Applications, pp. 147–149, 2008.

[13] J. C. Choi and C. W. Lee, “Energy modeling for the cluster-based sensor networks,” In Proceedings of the Sixth IEEE International Conference on Computer and Information Technology, pp. 218, September 20–22 2006.

[14] S. Selvakennedy and S. Sinnappan, “An energy-efficient clustering algorithm for multihop data gathering in wireless sensor networks,” Journal of Computers, pp. 1, April 2006.

[15] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocol for wireless microsensor networks,” In Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, January 2000.

[16] E. Popova and H. C. Wu, “Renewal reward processes with fuzzy rewards and their applications to T-age replacement policies,” European Journal of Operational Research, Vol. 117, pp. 606–617, 1999.

[17] R. Zhao and B. Liu, “Renewal process with fuzzy interarrival times and rewards,” International Journal of Uncertainty, Fuzziness and Knowledge based Systems, Vol. 11, pp. 573–586, 2003.

[18] H. D. Brunk, “The strong law of large numbers,” Duke Mathematical Journal, Vol. 15, pp. 181–195, 1948.

[19] http://www.xbow.com/Products/productdetails.aspx?sid=156

Wireless Sensor Network, **2010, 2, 141-147**
doi:10.4236/wsn.2010.22019 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Coordination for Networks of Dynamic Agents with Time-Varying Delays

Hongwang Yu[1], Baoshan Zhang[1], Yufan Zheng[2]
[1]*School of Mathematics and Statistic, Nanjing Audit University, Nanjing, China*
[2]*Department of Mathematics, Shanghai University, Shangha, China*
E-mail: {*yuhongwang, zbs*}*@nau.edu.cn*, {*yfzheng, yuhw*05820107}*@shu.edu.cn*
Received November 21, 2009; *revised December* 5, 2009; *accepted December* 7, 2009

Abstract

This paper is devoted to the study of the coordinate stability in undirected networks of dynamical agents with time-varying transmission delay. Neighbor-based rules are adopted to realize local control strategies for these continuous-time autonomous agents. Sufficient and necessary conditions in terms of linear matrix inequalities (LMIs) are given to guarantee the coordination of dynamical agents. Numerical simulations are given and demonstrate that our theoretical results are effective.

Keywords: Multi-Agent System, Time-Varying Delay, Consensus Protocol, Laplacian

1. Introduction

The coordinate stability of multi-agent systems is an important research topic in engineering applications, including moving in formation for fleets of unmanned aerial vehicles (UAVs), satellite clusters and mobile sensor networks. In the last few years, it has attracted more attention in diverse fields in physics, biophysics, systems biology, applied mathematics, mechanics, computer science and control theory.

In the multi-agent systems, agents are usually coupled and interconnected with some simple rules including a proposed first-order/two-order dynamical model and nearest-neighbor rules. Using graph theory, Jadbabaie *et al.* [1] provided a theoretical explanation for the consensus behavior of dynamical multi-agents. The extended results under some more relaxable conditions are studied in [2]. Saber *et al.* investigated a systematical framework of consensus problem under a variety of assumptions on the network topology (fixed or switching), presence or lack of communication delays, and directed or undirected network information flow [3,4].

In networks of the dynamic agents, time-varying delays may arise naturally, e.g., because of the moving of the agents, the congestion of the communication channels, the asymmetry of interactions, and the finite transmission speed due to the physical characteristics of the medium transmitting the information. The different consensus protocols have been investigated in [4–6], where the communication delay is a fixed constant. The average-consensus problem of agents under continuous-time networks with both switching topology and time-delay is studied in [7,8], where the dynamics order of each agent is one. A leader-following consensus problem for multiple agent with communication transmission time delays is discussed in [9], where the dynamics of each agent is second order.

Motivated by [7] and [9], we study the coordinated stability of multi-agent systems where the dynamics of each agent is second order in this paper. The communication transmission time delays of multi-agent systems are varying and the interconnection graph of the agents is undirected. The method used in this paper is partly motivated by the work of [10,11].

This paper is organized as follows. In Section 2, we recall some properties of graph and give the problem formulation. Coordinated stability analysis of the agents under network is given in Section 3. Section 4 gives a simulation example. Section 5 is a conclusion.

2. Preliminaries

By $G=(V,E,A)$, we denote an undirected graph with an weighted adjacency matrix $A=[a_{ij}]$, where $V=\{p_1,p_2,\ldots,p_M\}$ is the set of nodes, $E\subseteq V\times V$ is the set of edges. The node indexes belong to a finite index set $\underline{M}\triangleq\{1,2,\ldots,M\}$. An edge of G is denoted by $e_{ij}=(p_j,p_i)$ for some $i,j\in\underline{M}$. The adjacency elements a_{ij} are defined in following way: $e_{ij}\in E \Leftrightarrow a_{ij}>0$

and $e_{ij} \notin E \Leftrightarrow a_{ij} = 0$. Moreover, we assume $a_{ii} = 0$ for all $i \in \underline{M}$. The set of neighbors of node p_i is denoted by $N_i = \{p_j \in V \mid (p_i, p_j) \in E\}$.

A diagonal matrix $D = \{d_1, \ldots, d_M\} \in R^{M \times M}$ is a degree matrix of G, whose diagonal elements $d_i = \sum_{j=1}^{M} a_{ij}$ for $i \in \underline{M}$. Then the Laplacian of the weighted graph G is defined as $L = D - A$. A graph is called connected if there exists a path between any two distinct vertices of the graph.

Lemma 1 The graph G with the Laplacian L is connected if and only if rank(L)=M-1 and all eigenvalues of L are of positive real numbers except that only one eigenvalue is zero with eigenvector $1_M = (1, \ldots, 1)^T$.

In this paper, we consider a network of dynamical agents defined by a connected graph $G=(V, E, A)$. The node set V consists of dynamical agents $p_i, i \in \underline{M}$. The dynamics of $p_i, i \in \underline{M}$ are identical and described as follows.

$$\begin{aligned} \dot{x}_i &= v_i \\ m_i \dot{v}_i &= k v_i + u_i \\ y_i &= F \begin{pmatrix} x_i \\ v_i \end{pmatrix} \end{aligned} \tag{1}$$

where $x_i \in R^n$ is the location vector of agent p_i, $v_i \in R^n$ represents its velocity vector of the *i*-th agent, $u_i \in R^n$ is its coupling inputs and $m_i \in R^n$ is its mass. The control gain k is designed later. The output map indicates that the state information of dynamical agents is measured by, for example, some remote sensor and transmitted to other agents in network [10].

Due to time-delay in communicated network, the control protocol of the dynamical agent p_i is a neighbor-based linear control law in the form that

$$u_i = \sum_{p_j \in N_i} a_{ij} (y_j(t - \tau_{ij}(t)) - y_i(t - \tau_{ij}(t))) \tag{2}$$

where N_i is the set of neighbors of agent p_i and a_{ij} are adjacency elements of A. The $\tau_{ij}(t) \geq 0$, denoting the communication transmission time-delay from agent p_j to agent p_i. In the following, we assume that time-varying delays in (2) satisfy

$$0 \leq \tau_{ij}(t) \leq d, \quad \dot{\tau}_{ij}(t) \leq h \tag{3}$$

or

$$0 \leq \tau_{ij}(t) \leq d \tag{4}$$

For $t \geq 0$. That is to say, nothing has been known about the derivative of $\tau_{ij}(t)$, where d and h are positive constant numbers.

To focus our study in a main stream, we simply assume that $m_i = 1$, the observation matrix $F = [I_{n \times n} \ 0_{n \times n}]$ and $\tau_{ij}(t) = \tau(t)$ for all $i, j \in \underline{M}$ in this paper. We shall give the conditions, under which the network of dynamical agents (\ref{dyn0}) achieve asymptotical consensus stability meaning that there exists a fixed position (equilibrium) $x^* \in R$ such that for $i \in \underline{M}$

$$\begin{aligned} &\lim_{t \to \infty} x_i(t) = x^* \otimes 1_n \\ &\lim_{t \to \infty} v_i(t) = 0_{n \times 1} \end{aligned} \tag{5}$$

3. Coordination of Dynamic Agents with Time-Varying Delay

We study the collective behavior of dynamical agents under a class of communicated networks. The collective behavior of dynamical agents in network can be described by $x(t) = (x_1^T(t), \ldots, x_M^T(t))^T \in R^{Mn}$, $v(t) = (v_1^T(t), \ldots, v_M^T(t))^T$ and its communication topology is characterized by a connected graph G. By $x(0) = (x_1^T(0), \ldots, x_M^T(0))^T$, $v(0) = (v_1^T(0), \ldots, v_M^T(0))^T$, we denote the initial locations and the initial velocities of the agents, respectively.

3.1. Description of Dynamic Systems

Under control protocol (2) with $\tau_{ij}(t) = \tau(t)$ for all $i, j \in \underline{M}$, the dynamical equations of each agent of multi-agent systems are written by

$$\dot{x}_i(t) = A x_i(t) + B \sum_{p_j \in N_i} a_{ij} (x_j(t - \tau(t)) - x_i(t - \tau(t))) \tag{6}$$

where $x_i(t) = (x_i^T(t), v_i^T(t))^T$, $i \in \underline{M}$

$$A = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ 0_{n \times n} & k I_{n \times n} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} \\ I_{n \times n} & 0_{n \times n} \end{bmatrix}.$$

Furthermore, let $x(t) = (x_1^T(t), \ldots, x_M^T(t))^T$, then the dynamic network is of the following form

$$\dot{x}(t) = (I_M \otimes A) x(t) - (L \otimes B) x(t - \tau(t)) \tag{7}$$

where *L* is the Laplacian associated with the connected graph *G*. Moreover, we have the following result, which is similar to the dynamic systems without time- delay [10].

Lemma 2 The dynamics of System 7 is stabilized if and only if M systems

$$\dot{h}_i(t) = A h_i(t) - \lambda_i B h_i(t - \tau(t)) \tag{8}$$

are globally asymptotical stable, where $\lambda_i, i \in \underline{M}$ are the nonnegative eigenvalues of *L*.

Proof Since the Laplacian L of undirected graph G is real symmetric matrix, there exists an orthogonal matrix W such that

$$W^T L W = \Lambda = \begin{bmatrix} l_1 & 0 & \cdots & 0 \\ 0 & l_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_M \end{bmatrix}$$

where $l_i, i \in \underline{M}$ are the nonnegative real eigenvalues of Laplacian L. By the transform $h = (W \otimes I_{2n\times 2n})x$, we may obtain

$$\begin{aligned} \dot{h} &= (W \otimes I_{2n\times 2n})[(I_M \otimes A)x(t) - (L \otimes B)x(t - t(t))] \\ &= (I_M \otimes A)h(t) - \Lambda h(t - t(t)) \end{aligned}$$

which implies that the dynamics of System 7 is stabilized if and only if M Systems 8 are globally asymptotical stable.

3.2. Main Results

First, by means of linear matrix inequality (LMI), we study consensus stability of dynamic Systems 8 with certain communication transmission time-varying delay $t(t)$.

Theorem 1 The dynamic equations of (8) that the eigenvalues of Laplacian L are zero, *i.e.*, $l_s = 0$ for some $s \in \underline{M}$, achieves globally asymptotical stable if the control gain $k < 0$.

Moreover, let $h_s(t) = \begin{pmatrix} h_{1s}(t) \\ h_{2s}(t) \end{pmatrix} \otimes 1_n$, then

$$\lim_{t\to\infty} h_s(t) = \begin{pmatrix} x^* \\ 0 \end{pmatrix} \otimes 1_n, \qquad (9)$$

where $x^* = h_{s1}(0) - \frac{1}{k} h_{s2}(0)$.

Proof Consider the dynamic equations of (8) with $l_s = 0$ for some $s \in \underline{M}$, it is easy to obtain their expressions as the following

$\dot{h}_s(t) = A h_s(t)$.

Denoting $h_s(t) = \begin{pmatrix} h_{1s}(t) \\ h_{2s}(t) \end{pmatrix} \otimes 1_n$, we have

$$\dot{h}_{s1}(t) = h_{s2}(t), \qquad \dot{h}_{s2}(t) = k h_{s2}(t).$$

Then

$$h_{s2}(t) = e^{kt} h_{s2}(0), \quad h_{s2}(t) = \frac{e^{kt} - 1}{k} h_{s2}(0) + h_{s1}(0).$$

Since $k < 0$, one gets

$$\lim_{t\to\infty} h_{s1}(t) = h_{s1}(0) - \frac{1}{k} h_{s2}(0), \quad \lim_{t\to\infty} h_{s2}(t) = 0.$$

which leads to the result of Theorem 1.

In order to prove our main result relevant to the dynamic Systems 8 with communication transmission time-varying delay, we recite the following lemma [7].

Lemma 3 For any real differentiable vector function $z(t) \in R^n$ and any $n \times n$ symmetric positive definite matrix Γ, one has the following inequality

$$\begin{aligned} &[z(t) - z(t - t(t))]^T \cdot \Gamma \cdot [z(t) - z(t - t(t))] \\ &\le d \int_{t - t(t)}^{t} \dot{z}(s)^T \Gamma \dot{z}(s) ds \end{aligned}$$

where $t(t)$ satisfies $0 \le t(t) \le d$.

Theorem 2 Assume that the control gain $k < 0$ and the communication transmission time-varying delay satisfies (3). If there exist symmetric positive definite matrices $P_i, Q_i, R_i \in R^{2n\times 2n}$ such that the following conditions hold:

$$\begin{bmatrix} \Phi_{i1} & \Phi_{i2} \\ \Phi_{i2}^T & \Phi_{i3} \end{bmatrix} < 0, \qquad \Phi_{i3} > 0 \qquad (10)$$

where

$$\begin{aligned} \Phi_{i1} &= -\Omega_i + hQ_i + d(A - l_iB)^T R_i (A - l_iB), \\ \Phi_{i2} &= l_i P_i B + (1-h)Q_i + d\, l_i (A - l_iB)^T R_i B, \\ \Phi_{i3} &= \frac{1}{d} R_i + (1-h)Q_i - d\, l_i^2 B^T R_i B, \\ \Omega_i &= -[(A - l_iB)^T P_i + P_i(A - l_iB)]. \end{aligned} \qquad (11)$$

with properly choosing $d \ge 0$, $h \ge 0$. Then the origin of the i-th dynamic System 8 is asymptotical stable equilibrium point if and only if the communication networked topology G is connected.

Proof (Sufficiency) Since the undirected communication networked topology G is connected, the eigenvalues $l_i, i = 2, \cdots, M$ of Laplacian L are positive numbers in addition to $l_1 = 0$ from **Lemma 1**. Consider the characteristic polynomial of $A - l_iB, i = 2, \cdots, M$

$$p_{A - l_iB}(s) = \det(sI - (A - l_iB)) = (s^2 - ks + l_i)^n.$$

Since $k < 0$ and $l_i > 0$, it is easy to be verify that $A - l_iB$ is Hurwitz. Then there exists a symmetric positive definite matrix P_i such that

$\Omega_i = -[(A - l_iB)^T P_i + P_i(A - l_iB)]$ is positive definite matrix. So (10) is always feasible for appropriate positive scalars h and d.

Take a Lyapunov function for the i-th dynamic System 8 as follows:

$$\begin{aligned} V_i(t) &= h_i^T(t) P_i h_i(t) + \int_{t - t(t)}^{t} h_i^T(s) Q_i h_i(s) ds \\ &+ \int_{-d}^{0} (s + d) \dot{h}_i^T(t+s) R_i \dot{h}_i(t+s) ds \end{aligned}$$

Rewrite the i-th dynamic System 8 as the following equivalent form

$$\dot{h}_i(t) = (A - l_iB)h_i(t) + l_iBz_i(t) \qquad (12)$$

where $z_i(t) = h_i(t) - h_i(t - t(t))$. Along the trajectory of the solution of System 12, we have

$$\begin{aligned}\dot{V}_i(t) &= h_i^T(t)[(A-l_iB)^TP_i + P_i(A-l_iB)]h_i(t)\\ &+ l_i[h_i^T(t)P_iBz_i(t) + z_i^T(t)B^TP_ih_i(t)]\\ &+ h_i^T(t)Q_ih_i(t) - (1-\dot{t}(t))h_i^T(t-t(t))Q_ih_i(t-t(t))\\ &+ d\dot{h}_i^T(t)R_i\dot{h}_i(t) - \int_{-d}^{0}\dot{h}_i^T(t+s)R_i\dot{h}_i(t+s)ds.\end{aligned}$$

With the condition (3) and **Lemma 2**, we have

$$\begin{aligned}\dot{V}_i(t) &\le h_i^T(t)[(A-l_iB)^TP_i + P_i(A-l_iB) + hQ_i\\ &+ d(A-l_iB)^TR_i(A-l_iB)]h_i(t) + h_i^T(t)[l_iP_iB\\ &+ (1-h)Q_i + d\,l_i(A-l_iB)^TR_iB]z_i(t) + z_i^T(t)\\ &\cdot[l_iB^TP_i + (1-h)Q_i + d\,l_iB^TR_i(A-l_iB)]\cdot h_i(t)\\ &- z_i^T(t)[(1-h)Q_i - d\,l_i^2B^TR_iB + d^{-1}R_i]z_i(t)\\ &= \begin{pmatrix}h_i^T(t) & z_i^T(t)\end{pmatrix}\begin{bmatrix}\Phi_{i1} & \Phi_{i2}\\ \Phi_{i2}^T & -\Phi_{i3}\end{bmatrix}\begin{pmatrix}h_i(t)\\ z_i(t)\end{pmatrix}\end{aligned}$$

where Φ_{ij}, $j = 1,2,3$ are defined in (10). Therefore, there exists a positive constant b_i such that

$$\dot{V}_i(t) \le -b_i\left\|\begin{matrix}h_i(t)\\ z_i(t)\end{matrix}\right\| \le -b_i\left\|h_i(t)\right\|.$$

This implies that the i-th dynamic System of 8 achieve asymptotical stable for $0 \le t(t) \le d$ and $0 \le \dot{t}(t) \le h\cdot$

(Necessary) Since the origin of the dynamic Systems 8 is asymptotical stable equilibrium point, the eigenvalue of $A - l_iB$ have negative real-parts except that at most n eigenvalues are zero. Considering the Laplacian L and the characteristic polynomial of $A - l_iB$, one may get $l_1 = 0$ and the eigenvalues l_i, $i = 2,\cdots,M$ of Laplacian L are positive numbers. By **Lemma 1**, we may get the communication networked topology is connected.

Due to the reversible orthogonal transform, the M dynamic Systems 8 are equivalent to the System 7. So we get the same result of stability for the System 7.

Theorem 3 Assume that the graph G is connected, the control gain $k < 0$ and the communication transmission time-varying delay satisfies (3). If there exist symmetric positive definite matrices $P_i, Q_i, R_i \in R^{2n\times 2n}$ such that the following linear matrix inequalities hold

$$\begin{bmatrix}\Phi_{21} & \Phi_{22} & \cdots & 0_{2n\times 2n} & 0_{2n\times 2n}\\ \Phi_{22}^T & -\Phi_{23} & \cdots & 0_{2n\times 2n} & 0_{2n\times 2n}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0_{2n\times 2n} & 0_{2n\times 2n} & \cdots & \Phi_{M1} & \Phi_{M2}\\ 0_{2n\times 2n} & 0_{2n\times 2n} & \cdots & \Phi_{M2}^T & -\Phi_{M3}\end{bmatrix} < 0,$$

$$\Phi_{i3} > 0 \qquad (13)$$

with properly choosing positive scalars h and d, where Φ_{ij} ($i \in \{2,\cdots,M\}$ and $j = 1,2,3$) are defined in (11). Then the dynamic System 7 achieves globally asymptotical consensus stability if and only if the communication networked topology G is connected.

Moreover,

$$\lim_{t\to\infty} x(t) = 1_M \otimes \begin{pmatrix}x^*\\ 0\end{pmatrix} \otimes 1_n, \qquad (14)$$

where $x^* = \dfrac{1}{M}\sum_{i=1}^{M}[x_i(0) - \dfrac{1}{k}v_i(0)] \otimes 1_n\cdot$

Proof By **Lemma 1**, the networked topology G is connected if and only if the real eigenvalues of Laplacian L with $0 = l_1 < l_2 \le l_3 \le \cdots \le l_M$. Under the given conditions, the M dynamic Systems 8 achieve globally asymptotical stable if and only if the communication networked topology G is connected from Theorem 1 and Theorem 2.

Since the transform $h = (W \otimes I_{2n\times 2n})x$ is reversible orthogonal, the M dynamic Systems 8 are equivalent to System 7. Hence, for appropriate positive scalars h and d, one can conclude that the System 7 achieves globally asymptotically consensus stability if and only if the communication networked topology G is connected. And the allowable h and d can be obtained by the feasible linear matrix Inequality 13.

As the origin of the dynamic Systems of 8 for $i \in \{2,\cdots,M\}$ is asymptotically stable equilibrium, we have $\begin{pmatrix}h_2^T(t) & \cdots & h_M^T(t)\end{pmatrix} \to 0_{1\times 2n(M-1)}$.

Due to the fact that one eigenvalue of Laplacian is zero with eigenvector $1_M = (1,\cdots,1)^T$, one may get

$$W = \frac{1}{\sqrt{M}}\begin{bmatrix}1 & 1_{M-1}^T\\ 1_{M-1} & W_1\end{bmatrix}$$

with $W_1 1_{M-1} = -1_{M-1}$, $W_1^TW_1 = MI_{M-1} - 1_{M-1}1_{M-1}^T$.

By Lemma \ref{lem1} and **Lemma 2**, one gets

$$\begin{aligned}\begin{pmatrix}h_2^T, & \cdots, & h_M^T\end{pmatrix}^T &= \frac{1}{\sqrt{M}}(1_{M-1} \otimes I_{2n\otimes 2n})x_1\\ &+ \frac{1}{\sqrt{M}}(W_1^T \otimes I_{2n\otimes 2n})\begin{pmatrix}x_2^T, & \cdots, & x_M^T\end{pmatrix}^T\end{aligned}$$

Then, we can obtain that

$$x_i = \bar{x} = \frac{1}{M}\sum_{m=1}^{M}x_m, \quad i \in \{2,\cdots,M\}.$$

One can get $h_1 = \dfrac{1}{\sqrt{M}}\sum_{i=1}^{M}x_i$ by **Lemma 1**. Then from Theorem 1, it is hold that $\lim_{t\to\infty}h_1(t) = x^*\begin{pmatrix}1_n\\ 0\end{pmatrix}$, with

$$x^* = h_{11}(0) - \frac{1}{k}h_{12}(0) = \frac{1}{\sqrt{M}}\sum_{i=1}^{M}[x_i(0) - \frac{1}{k}v_i(0)].$$

Therefore, we can obtain

$$\lim_{t\to\infty} x_i(t) = \frac{1}{M}\sum_{i=1}^{M}[x_i(0) - \frac{1}{k}v_i(0)] \otimes 1_n,$$

$$\lim_{t\to\infty} v_i(t) = 0_{n\times 1}$$

The proof of the Theorem is completed.

Remark For any $h \ge 0$, the maximal allowable d guaranteeing average consensus in Theorem 2 and/or Theorem 3 can be obtained from the following optimization problem:

Maximize d

s.t. $0 \le h < 1,\ P_i > 0,\ Q_i > 0,\ R_i > 0$ and (13).

This optimization problem can be solved by using the GEVP solver in Matlab's Control Systems Toolbox [5].

Considering the matrices in the linear matrix Inequality 13 are continuous for $l_i > 0$, $i \in \{2, \cdots, M\}$, we may obtain the following corollary for estimation of conservative upper bound h and d.

Corollary 1. Assume that the control gain $k < 0$ and the communication time-varying delay satisfies (3). If there exist symmetric positive definite matrices $P_i, Q_i, R_i \in R^{2n\times 2n}$ such that the following linear matrix inequalities hold

$$\begin{bmatrix} \Phi_{21} & \Phi_{22} & 0_{2n\times 2n} & 0_{2n\times 2n} \\ \Phi_{22}^T & -\Phi_{23} & 0_{2n\times 2n} & 0_{2n\times 2n} \\ 0_{2n\times 2n} & 0_{2n\times 2n} & \Phi_{M1} & \Phi_{M2} \\ 0_{2n\times 2n} & 0_{2n\times 2n} & \Phi_{M2}^T & -\Phi_{M3} \end{bmatrix} < 0,$$

$$\Phi_{23} > 0 \qquad \Phi_{M3} > 0 \tag{15}$$

with properly choosing $d \ge 0$ and $h \ge 0$, where Φ_{2j}, Φ_{Mj} ($j = 1,2,3$) are defined in (13). Then the dynamic System 7 achieves globally asymptotical consensus stability if and only if the communication networked topology G is connected.

When time-varying delays satisfy (4), that is to say, nothing has been known about the derivative of $\tau(t)$. For Systems 8, one may construct the following Lyapunov function as

$$W_i(t) = h_i^T(t)S_i h_i(t) + \int_{-d}^{0}\int_{t+\theta}^{t} \dot{h}_i^T(s)R_i\dot{h}_i(s)dsd\theta.$$

Similar to the proof of Theorem 2 and Theorem 3, it is easy to get the following results and we may omit their proof here.

Theorem 4 Assume that the control gain $k < 0$ and the communication transmission time-varying delay satisfies (4). If there exist symmetric positive definite matrices $S_i, T_i \in R^{2n\times 2n}$ such that the following conditions hold:

$$\begin{bmatrix} \Psi_{i1} & \Psi_{i2} \\ \Psi_{i2}^T & \Psi_{i3} \end{bmatrix} < 0, \qquad \Psi_{i3} > 0 \tag{16}$$

where

$$\Psi_{i1} = -\Gamma_i + d(A - l_iB)^T T_i(A - l_iB),$$

$$\Psi_{i2} = l_iS_iB + d\,l_i(A - l_iB)^T T_iB,$$

$$\Psi_{i3} = \frac{1}{d}T_i - d\,l_i^2B^TT_iB,$$

$$\Gamma_i = -[(A - l_iB)^T S_i + S_i(A - l_iB)]. \tag{17}$$

with properly choosing $d \ge 0$. Then the origin of the i-th dynamic System of 8 is asymptotical stable equilibrium point if and only if the communication networked topology G is connected.

Theorem 5 Assume that the graph G is connected, the control gain $k < 0$ and the communication transmission time-varying delay satisfies (4). If there exist symmetric positive definite matrices $S_i, T_i \in R^{2n\times 2n}$ such that the following linear matrix inequalities hold

$$\begin{bmatrix} \Psi_{21} & \Psi_{22} & \cdots & 0_{2n\times 2n} & 0_{2n\times 2n} \\ \Psi_{22}^T & -\Psi_{23} & \cdots & 0_{2n\times 2n} & 0_{2n\times 2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{2n\times 2n} & 0_{2n\times 2n} & \cdots & \Psi_{M1} & \Psi_{M2} \\ 0_{2n\times 2n} & 0_{2n\times 2n} & \cdots & \Psi_{M2}^T & -\Psi_{M3} \end{bmatrix} < 0,$$

$$\Psi_{i3} > 0 \tag{18}$$

with properly choosing positive scalars d, where Ψ_{ij} ($i \in \{2, \cdots, M\}$ and $j = 1,2,3$) are defined in (17).

Then the dynamic System 7 achieves globally asymptotical consensus stability if and only if the communication networked topology G is connected.

Moreover,

$$\lim_{t\to\infty} x(t) = 1_M \otimes \begin{pmatrix} x^* \\ 0 \end{pmatrix} \otimes 1_n,$$

where $x^* = \dfrac{1}{M}\displaystyle\sum_{i=1}^{M}[x_i(0) - \frac{1}{k}v_i(0)] \otimes 1_n$.

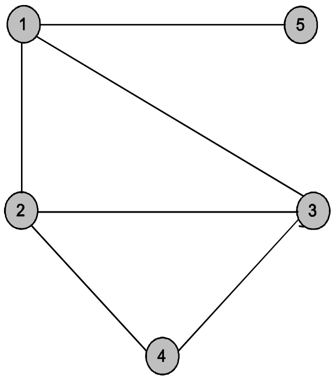


Figure 1 Undirected connected graph G with five nodes.

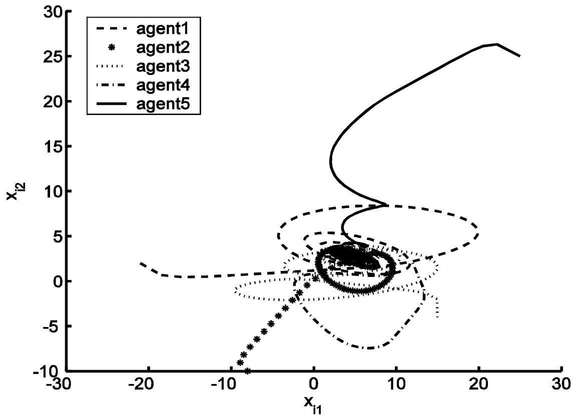


Figure 2. State trajectories of the agents in G.

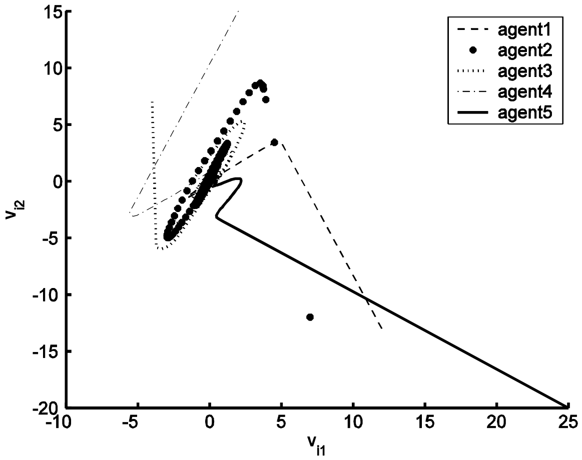


Figure 3. Velocity trajectories of the agents in G.

4. Simulations

Numerical simulations will be given to illustrate the theoretical results obtained in the previous section. Consider five dynamic agents under network described in Figure 1.

Here we consider the dynamical equations (\ref{dyn0}) with $n=2$. By employing the LMI Toolbox in Matlab, one gets that the maximum time-delay bound is $d=2.1152$ when $h=0$, *i.e.* the value of time-delay is fixed. When $h=0.5$, the maximum delay bound is $d=1.2799$. And we may get the corresponding feasible solutions in the following.

$$P_1=\begin{bmatrix}1.4978I_2 & 0.2519I_2\\0.2519I_2 & 0.9299I_2\end{bmatrix}\; P_2=\begin{bmatrix}1.3969I_2 & 0.2175I_2\\0.2175I_2 & 0.6666I_2\end{bmatrix}$$

$$P_3=\begin{bmatrix}1.2385I_2 & 0.0985I_2\\0.0985I_2 & 0.2196I_2\end{bmatrix}\; P_4=\begin{bmatrix}1.6896I_2 & 0.5056I_2\\0.2056I_2 & 0.2557I_2\end{bmatrix}$$

$$Q_1=\begin{bmatrix}0.1784I_2 & 0.0334I_2\\0.0334I_2 & 0.6915I_2\end{bmatrix}\; Q_2=\begin{bmatrix}0.3766I_2 & 0.0488I_2\\0.0488I_2 & 0.6925I_2\end{bmatrix}$$

$$R_3=\begin{bmatrix}2.0632I_2 & 0.1171I_2\\0.1171I_2 & 0.0099I_2\end{bmatrix}\; R_4=\begin{bmatrix}2.4174I_2 & 0.1864I_2\\0.1864I_2 & 0.0471I_2\end{bmatrix}$$

$$Q_3=\begin{bmatrix}0.2463I_2 & -0.0197I_2\\-0.0197I_2 & 0.3858I_2\end{bmatrix}\; Q_4=\begin{bmatrix}0.8508I_2 & -0.1547I_2\\-0.1547I_2 & 0.0829I_2\end{bmatrix}$$

$$R_1=\begin{bmatrix}1.4655I_2 & -0.0197I_2\\-0.0197I_2 & 0.1397I_2\end{bmatrix}\; R_2=\begin{bmatrix}1.9895I_2 & -0.0281I_2\\-0.0281I_2 & 0.0714I_2\end{bmatrix}$$

The agents have initial conditions $x_1(q)=(-21\quad 2)^T$, $x_2(q)=(-8\quad -10)^T$, $x_3(q)=(-15\quad 4)^T$, $x_4(q)=(12\quad 2)^T$, $x_5(q)=(25\quad 25)^T$, $v_1(q)=(12\quad -13)^T$, $v_2(q)=(-12\quad 5)^T$, $v_3(q)=(7\quad 18)^T$, $v_4(q)=(15\quad -25)^T$, $v_5(q)=(-20\quad 15)^T$ for $q\in[-1,\ 0]$. The eigenvalues of the Laplacian matrix are $l_1=0$, $l_2=0.8299$, $l_3=2.6889$, $l_4=4$, $l_5=4.4812$. Figure 2 and Figure 3 show the state and velocity trajectories of the multi-agent systems with time-varying delay $t(t)=0.2|\sin t|$.

5. Conclusions

In this paper, we discuss the coordinate stability of multi-agent systems where the agent is described by double-integrator with time-varying transmission delay in their communicated network. Two different time-varying delays are considered for dynamical systems. We firstly decompose the multi-agent systems into M dynamical systems by certain transformation of state space under the condition of undirected connected communication network. By the methods of linear matrix inequality (LMI), we study each dynamical system with time-varying delay and show that the agents of multi-agent systems can achieve globally asymptotical consensus stability. Meanwhile, the upper bound parameters of time-varying delay can be estimated by checking solutions of LMI. Numerical simulation results are provided and demonstrate the effectiveness of our theoretical results.

6. Acknowledgment

This work is supported by National Nature Science Foundation of China under Grant 60674046, the Nanjing Audit University Scientific Research Start-up Fund for High-level Talents and the Theory and Application of Differential Equations Foundation of the Nanjing Audit University.

7. References

[1] A. Jadbabaie, J. Lin, and S. A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," IEEE Transactions on Automatic Control, Vol. 48, No. 6, pp. 988–1000, June 2003.

[2] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," IEEE Transactions on Automatic Control, Vol. 50, No. 5, pp. 655–661, May 2005.

[3] R. O. Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, Vol. 49, No. 9, pp. 1520–1533, September 2004.

[4] R. O. Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multiagent systems," Proceedings of the IEEE, Vol. 95, No. 1, pp. 215–240, January 2007.

[5] L. Moreau, "Stability of multiagent systems with time-dependent communication," IEEE Transactions on Automatic Control, Vol. 50, No. 2, pp. 169–182, February 2005.

[6] D. Lee and W. M. Spong, "Agreement with non-uniform information delays," Proceedings of the 2006 American Control Conference, pp. 756–761, 2006.

[7] P. Lin and Y. Jia, "Average consensus in networks of multi-agents with both switching topology and coupling time-delay," Physica A, Vol. 387, pp. 303–313, 2008.

[8] Y. Sun, L. Wang, and G. Xie, "Average consensus in directed networks of dynamic agents with time-varying communication delays," Proceedings of the 45th IEEE Conference on Decision and Control, pp. 3393–3398, 2006.

[9] J. Hu and Y. Hong, "Leader-following coordination of multi-agent systems with coupling time delays," Physica A, Vol. 374, pp. 853–863, 2007.

[10] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," IEEE Transactions on Automatic Control, Vol. 49, No. 9, pp. 1465–1476, 2004.

[11] G. Xie and L. Wang, "Information flow and cooperative control of vehicle formations," International Journal of Robust and Nonlinear Control, Vol. 17, pp. 941–959, 2007.

[12] L. Moreau, "Stability of continuous-time distributed consensus algorithms," In Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, pp. 3998–4003, 2004.

[13] Dongjun Lee and Mark W. Spong, "Agreement with non-uniform information delays", In Proceedings of the 2006 American Control Conference, Minnesota, USA, June 14–16, 2006.

[14] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus using delayed information", In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, pp. 3375–3380, 2006.

[15] N. Biggs, "Algebraic, graph theory," Cambridge: Cambridge University Press, 1994.

[16] H. Yu, Y. Zheng, "Global behavior of dynamical agents in directed network," Proceedings of the 26th Chinese Control Conference, pp. 557–561, 2007.

[17] A. Isidori, "Nonlinear control systems II," London: Communications and Control Engineering Series, Springer-Verlag, 1999.

[18] R. A. Horn and C. R. Johnson, "Matrix analysis," Cambridge: Cambridge University Press, 1987.

Wireless Sensor Network, **2010, 2, 148-160**
doi:10.4236/wsn.2010.22020 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

A Novel Approach for Finding a Shortest Path in a Mixed Fuzzy Network

Ali Tajdin[1], Iraj Mahdavi[1*], Nezam Mahdavi-Amiri[2], Bahram Sadeghpour-Gildeh[3], Reza Hassanzadeh[1]
[1]*Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran*
[2]*Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran*
[3]*Department of Mathematics and Statistics, Mazandaran University, Babolsar, Iran*
E-mail: *irajarash@rediffmail.com*

Received October 11, 2009; *revised November* 13, 2009; *accepted November* 14, 2009

Abstract

We present a novel approach for computing a shortest path in a mixed fuzzy network, network having various fuzzy arc lengths. First, we develop a new technique for the addition of various fuzzy numbers in a path using a -cuts. Then, we present a dynamic programming method for finding a shortest path in the network. For this, we apply a recently proposed distance function for comparison of fuzzy numbers. Four examples are worked out to illustrate the applicability of the proposed approach as compared to two other methods in the literature as well as demonstrate the novel feature offered by our algorithm to find a fuzzy shortest path in mixed fuzzy networks with various settings for the fuzzy arc lengths.

Keywords: Fuzzy Numbers, a -Cut; Shortest Path, Dynamic Programming

1. Introduction

Determination of shortest distance and shortest path between two vertices is one of the most fundamental problems in graph theory. Let $G = (V, E)$ be a graph with V as the set of vertices and E as the set of edges. A *path* between two vertices is an alternating sequence of vertices and edges starting and ending with vertices, and no vertices or no edges appear more than once in the sequence. The *length* of a path is the sum of the weights of the edges on the path. There may exist more than one path between a pair of vertices. The *shortest path problem* is to find the path with minimum length between a specified pair of vertices. In classical graph theory, the weight of each edge is taken as a crisp real number. But, practically weight of each edge of the network may not be a fixed real number and it may well be imprecise.

The shortest path problem involves addition and comparison of the edge weights. Since, the addition and comparison between two interval numbers or between two triangular fuzzy numbers are not alike those between two precise real numbers, it is necessary to discuss them at first. Interval arithmetic was developed in Moore [1]. The case of optimization with interval valued and fuzzy constraints were discussed in Delgado *et al*., Ishibuchi and Tanaka, Sengupta, and Shaocheng [2–5]. Various ranking methods for interval numbers were introduced by several researchers. An extensive survey of the order relations along with a new proposal are given by Sengupta and Pal [6]. There are also ranking methods for fuzzy numbers available in the literature. Dubois and Prade [7] introduced a ranking of fuzzy numbers in the setting of possibility theory, and Chen [8] ranked fuzzy numbers using maximizing and minimizing sets. Ranking of fuzzy numbers was also studied by Bortolan and Degani, Cheng, and Delgado *et al.* [2,9,10].

Fuzzy graph problems were addressed in Blue *et al.* and Koczy, Klein, Li *et al.*, Lin and Chen, Okada and Gen [11–17] paid special attention to fuzzy shortest paths. In a recent development, Okada and Soper [18] proposed an algorithm to find the shortest path in a network with fuzzy edge weights. The algorithm gives a family of non-dominated shortest paths for a specified satisfaction level, but it does provide any guideline to the decision-maker to choose the best amongst the paths according to his/her view; *i.e.*, optimistic, pessimistic, etc.

The shortest path (SP) problem has received lots of attention from researchers in the past decades, because it is important to many applications such as communication, transportation, scheduling and routing. In a network, the arc length may represent time or cost. Conventionally, it is assumed to be crisp. However, it is difficult for decision makers to specify the arc lengths. For example, using the same modem to transmit the data from node *a* to *b* in a network, the data transmission time may not be the same every time. Therefore, in real world, the arc length could be uncertain. Fuzzy set theory, as proposed by Zadeh [19], is frequently utilized to deal with uncertainty. Zadeh presented the possibility theory using membership functions to describe uncertainties.

Considering a directed network that is composed of a finite set of nodes and a set of directed arcs, we denote each arc by an ordered pair *(i, j),* where *i* and *j* are two different nodes. The arc length represents the distance needed to traverse (*i, j*) from node *i* to *j*. We denote it by *l(i, j)* or *L(i, j),* when it is crisp or fuzzy, respectively. We call *L(i, j)* as "fuzzy arc length".

The shortest path problem is the following: given a weighted, directed graph and two special vertices *s* and *t*, compute the weight of the shortest path between *s* and *t*. The shortest path problem is one of the most fundamental network optimization problems. This problem comes up in practice and arises as a subproblem in many network optimization algorithms. Algorithms for this problem have been studied for a long time [20–22]. However, advances in the method and theory of shortest path algorithms are still being made [23–25].

In the network we consider here, the lengths of the arcs are assumed to represent transportation times or costs rather than geographical distances. As time or cost fluctuate with traffic conditions, payload and so on, it is not practical to represent the arcs as crisp values. Thus, it is appropriate to utilize fuzzy numbers based on fuzzy set theory. In proposing an algorithm for solving the problem, we are first faced with the comparison or ordering of fuzzy numbers, a task not considered to be routine. For this reason, fuzzy shortest path problems have rarely been studied despite their potential application to many problems [18,26]. The problem turns out to be even more complicated in our more general case of allowing various fuzzy arc lengths.

Here, we propose a new approach and an algorithm to find a shortest path in a mixed network having various fuzzy arc lengths. The remainder of the paper is organized as follows. In Section 2, basic concepts and definitions are given. A dynamic programming algorithm for finding a fuzzy shortest path in a network is presented in Section 3. There, we make use of a-cuts for computing approximations for the addition of two different types of fuzzy numbers and apply a distance function for the comparison of fuzzy numbers. Comparative examples are given in Section 4. Section 5 works out an example to show the novel feature of our algorithm to find fuzzy shortest paths in mixed fuzzy networks with various settings for the fuzzy arc lengths. We conclude in Section 6.

2. Concepts and Definitions

We start with basic definitions of some well-known fuzzy numbers.

Definition 1. An *LR* fuzzy number is represented by $\tilde{A} = (m, a, b)_{LR}$, with the membership function, $m_{\tilde{a}}(x)$, defined by the expression,

$$m_{\tilde{a}}(x) = \begin{cases} L\left[\dfrac{m-x}{a}\right] & x \le m \\ R\left[\dfrac{x-m}{b}\right] & x \ge m, \end{cases}$$

where *L* and *R* are non-increasing functions from R^+ to [0,1], *L*(0)=*R*(0)=1, *m* is the center, a is the left spread and b is the right spread.

Note that if *L(x)=R(x)*=1-*x* with 0<*x*<1, then *x* is a triangular fuzzy number and is represented by the triplet $\tilde{a} = (a_1, a_2, a_3)$, with the membership function, $m_{\tilde{a}}(x)$, defined by

$$m_{\tilde{a}}(x) = \begin{cases} 0 & x \le a_1 \\ \dfrac{x-a_1}{a_2-a_1} & a_1 < x \le a_2 \\ \dfrac{a_3-x}{a_3-a_2} & a_2 < x \le a_3 \\ 0 & x > a_3. \end{cases}$$

A triangular fuzzy number is shown in Figure 1.

Definition 2. A trapezoidal fuzzy number $\tilde{a}$ is shown by $\tilde{a} = (a_1, a_2, a_3, a_4)$, with the membership function as follows:

$$m_{\tilde{a}}(x) = \begin{cases} 0 & x \le a_1 \\ \dfrac{x-a_1}{a_2-a_1} & a_1 \le x \le a_2 \\ 1 & a_2 \le x \le a_3 \\ \dfrac{a_4-x}{a_4-a_3} & a_3 \le x \le a_4 \\ 0 & a_4 \le x. \end{cases}$$

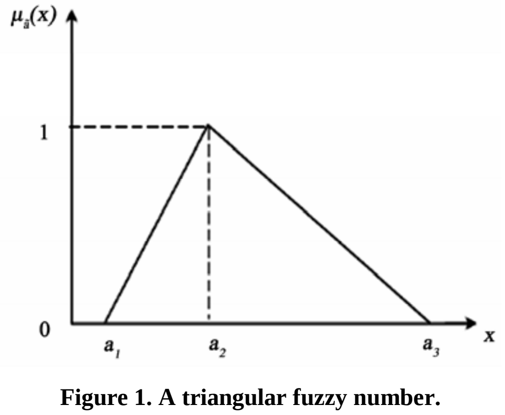


Figure 1. A triangular fuzzy number.

A general trapezoidal fuzzy number is shown in Figure 2. It is apparent that a triangular fuzzy number is a special trapezoidal fuzzy number with $a_2 = a_3$.

Definition 3. If $L(x)=R(x)=$ e^{-x^2}, with $x \in \Re$, then x is a normal fuzzy number that is shown by (m, s) and the corresponding membership function is defined to be:

$$m_{\tilde{a}}(x) = e^{-(\frac{x-m}{s})^2}, \quad x \in \Re,$$

where m is the mean and s is the standard deviation. A normal fuzzy number is shown in Figure 3.

Definition 4. The a -cut and strong a -cut for a fuzzy set $\tilde{A}$ are shown by $\tilde{A}_a$ and $\tilde{A}_a^+$, respectively, and for $a \in [0,1]$ are defined to be:

$$\tilde{A}_a = \{x \mid m_{\tilde{A}}(x) \ge a, x \in X\},$$

$$\tilde{A}_a^+ = \{x \mid m_{\tilde{A}}(x) > a, x \in X\},$$

where X is the universal set.

Upper and lower bounds for any a -cut $(\tilde{A}_a)$ are shown by $\sup \tilde{A}_a$ and $\inf \tilde{A}_a$, respectively. Here, we assume that the upper and lower bounds of a -cuts are finite values and for simplicity we show $\sup \tilde{A}_a$ by A_a^+ and $\inf \tilde{A}_a$ by A_a^-.

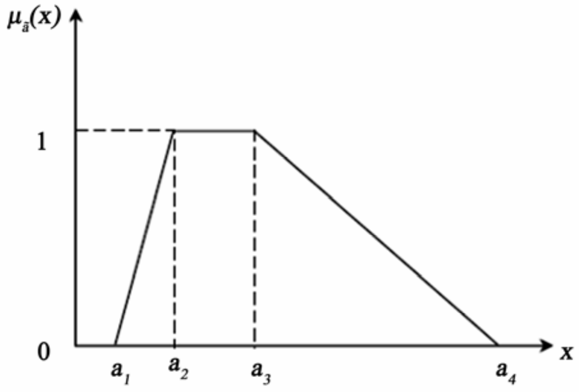


Figure 2. A trapezoidal fuzzy number.

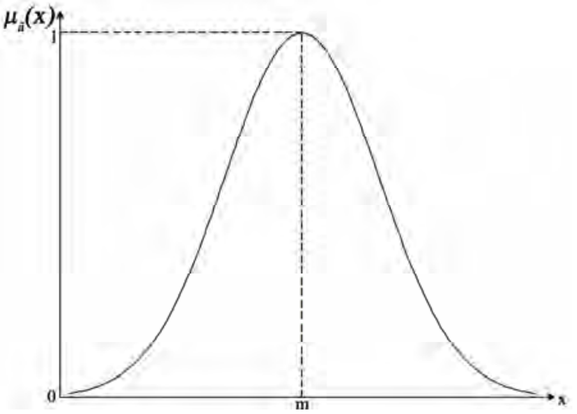


Figure 3. A normal fuzzy number.

2.1. Computing a -Cuts for Fuzzy Numbers

For an LR fuzzy number with L and R as invertible functions, the a -cuts are:

$$a = L[\frac{m-x}{b}] \Rightarrow \frac{m-x}{b} = L^{-1}(a) \Rightarrow x = m - bL^{-1}(a),$$

$$a = R[\frac{x-m}{g}] \Rightarrow \frac{x-m}{g} = R^{-1}(a) \Rightarrow x = m + gR^{-1}(a).$$

For specific L and R functions, the following cases are discussed. Let $\tilde{a} = (a_1, a_2, a_3, a_4)$ be a trapezoidal fuzzy number. The a -cut for $\tilde{a}$, $(\tilde{a}_a)$, is computed as:

$$m_{\tilde{a}}(x) = \begin{cases} 0 & x \le a_1 \\ \frac{x-a_1}{a_2-a_1} & a_1 \le x \le a_2 \\ 1 & a_2 \le x \le a_3 \\ \frac{a_4-x}{a_4-a_3} & a_3 \le x \le a_4 \\ 0 & a_4 \le x \end{cases}$$

$$\Rightarrow \begin{matrix} a = \frac{x-a_1}{a_2-a_1} \Rightarrow & x = (a_2-a_1)a + a_1 \\ a = \frac{a_4-x}{a_4-a_3} \Rightarrow & x = a_4 - (a_4-a_3)a \end{matrix}$$

$$\Rightarrow \tilde{a}_a = \begin{cases} \tilde{a}_a^+ = a_4 - (a_4-a_3)a \\ \\ \tilde{a}_a^- = (a_2-a_1)a + a_1 \end{cases}, \quad 0 \le a \le 1. \qquad (1)$$

Note that the a -cut for triangular fuzzy numbers is simply obtained by using the above equations considering $a_2 = a_3$:

$$\tilde{a}_a = \begin{cases} \tilde{a}_a^+ = a_3 - (a_3-a_2)a \\ \tilde{a}_a^- = (a_2-a_1)a_a + a_1 \end{cases}, \quad 0 \le a \le 1. \qquad (2)$$

For $\tilde{a} = (m, s)$, a normal fuzzy number, $\tilde{a}_a$, is computed as:

$$a = e^{-(\frac{m-x}{s})2} \Rightarrow \sqrt{-\ln(a)} = \frac{m-x}{s}$$

$$a = e^{-(\frac{x-m}{s})2} \Rightarrow \sqrt{-\ln(a)} = \frac{x-m}{s}$$

$$\Rightarrow \tilde{a}_a = \begin{cases} \tilde{a}_a^L = m - s\sqrt{-\ln a} \\ \\ \tilde{a}_a^R = m + s\sqrt{-\ln a} \end{cases}, \quad 0 < a \le 1. \qquad (3)$$

2.2. Fuzzy Sum Operators

Here, we propose an approach for summing various fuzzy numbers approximately using a -cuts. The approximation is based on fitting an appropriate model for the sum using a -cuts of the addition by a set of a_i values. Let us divide the a -interval [0,1] into n equal subintervals by letting $a_0 = 0$, $a_i = a_{i-1} + \Delta a_i$, *i=1,...,n* with $\Delta a_i = \frac{1}{n}$, *i=1,...,n*. This way, we have a set of *n+1* equidistant a_i points. For addition of two different fuzzy numbers, we add the set of corresponding a_i-cut points of the two numbers to yield the a_i-cuts of the sum as an approximation for the fuzzy addition.

3. An Algorithm for Fuzzy Shortest Path in a Network

3.1. Distance between Fuzzy Numbers

Knowing that we can obtain a good approximation of the addition of various fuzzy numbers by use of a -cuts, we compute the distance between two fuzzy numbers using the resulting points from the a -cuts. For $\tilde{a}$ and $\tilde{b}$ as two different fuzzy numbers, we use a new fuzzy ranking method for the fuzzy numbers. Let us consider the fuzzy *min* operation to be defined as follows

$$
\begin{aligned}
M\tilde{V} &= \text{Min} \quad \text{value}(\tilde{a},\tilde{b}) \\
&= (\min(a_1,b_1),\ \min(a_2,b_2),\ \min(a_3,b_3), \min(a_4,b_4)).
\end{aligned} \tag{4}
$$

It is evident that, for non-comparable fuzzy numbers $\tilde{a}$ and $\tilde{b}$, the fuzzy *min* operation results in a fuzzy number different from both of them. For example, for $\tilde{a} = (5,10,13,19)$ and $\tilde{b} = (6,9,15,20)$, we get from (4) a fuzzy $M\tilde{V} = (5,9,13,19)$ which is different from both $\tilde{a}$ and $\tilde{b}$. To alleviate this drawback, we use a method based on the distance between fuzzy numbers. We use the distance function introduced by Sadegh-pour-Gildeh and Gien [27]. The main advantages of this distance function are the generality of its usage on various fuzzy numbers, and its reliability in distinguishing unequal fuzzy numbers. Indeed, the use of this distance function worked out to be quite appropriate for our approach here as well as in a different context before where we considered the arc lengths in the network to be all of the same type (see Mahdavi *et al.* [28]).

The proposed $D_{p,q}$ -distance, indexed by parameters $1 < p < \infty$ and $0 < q < 1$, between two fuzzy numbers $\tilde{a}$ and $\tilde{b}$ is a nonnegative function given by:

$$
D_{p,q}(\tilde{a},\tilde{b}) = \begin{cases} \left[(1-q)\int_0^1 \left| a_a^- - b_a^- \right|^p da + q\int_0^1 \left| a_a^+ - b_a^+ \right|^p da \right]^{\frac{1}{p}}, & p < \infty, \\ (1-q) \sup\limits_{0<a\le 1} \left(\left| a_a^- - b_a^- \right| \right) + q \inf\limits_{0<a\le 1} \left(\left| a_a^+ - b_a^+ \right| \right) , & p = \infty. \end{cases} \tag{5}
$$

The analytical properties of $D_{p,q}$ depend on the first parameter p, while the second parameter q of $D_{p,q}$ characterizes the subjective weight attributed to the end points of the support; *i.e.*, (a_a^+, a_a^-) of the fuzzy numbers. If there is no reason for distinguishing any side of the fuzzy numbers, then $D_{p,\frac{1}{2}}$ is recommended. Having q close to 1 results in considering the right side of the support of the fuzzy numbers more favorably. Since the significance of the end points of the support of the fuzzy numbers is assumed to be the same, then we consider $q = \frac{1}{2}$.

For two fuzzy numbers $\tilde{a}$ and $\tilde{b}$ with corresponding a_i-cuts, the $D_{p,q}$ distance is approximately proportional to:

$$
D_{p,q}(\tilde{a},\tilde{b}) = \left[(1-q)\sum_{i=1}^{n} \left| a_{a_i}^- - b_{a_i}^- \right|^p + q\sum_{i=1}^{n} \left| a_{a_i}^+ - b_{a_i}^+ \right|^p \right]^{\frac{1}{p}}. \tag{6}
$$

If $q = \frac{1}{2}, p = 2$, then the above equation turns into:

$$
D_{2,\frac{1}{2}}(\tilde{a},\tilde{b}) = \sqrt{\frac{1}{2}\sum_{i=1}^{n} (a_{a_i}^- - b_{a_i}^-)^2 + \frac{1}{2}\sum_{i=1}^{n} (a_{a_i}^+ - b_{a_i}^+)^2} , \tag{7}
$$

To compare two fuzzy arc lengths $\tilde{a}$ and $\tilde{b}$ with a_i-cuts as their approximations, since they are supposed to represent positive values, we compare them with $M\tilde{V} = (0,0,....,0)$. In fact, we use (7) to compute $D_{2,\frac{1}{2}}(\tilde{a},0)$ and $D_{2,\frac{1}{2}}(\tilde{b},0)$ and use these values for comparison of the two numbers.

3.2. An Algorithm for Computing s Shortest Fuzzy Path

The following dynamic programming algorithm is for computing the shortest path in a network. The algorithm is based on Floyd's dynamic programming method to find a shortest path, if it exists, between every pair of nodes *i* and *j* in the network (see Floyd [29]).

We make use of the following optimal value function $f_k(i,j)$ and the corresponding labeling function $P_k(i,j)$:

$f_k(i,j)$ = length of the shortest path from node *i* to node *j* when the path is considered to use only the nodes from the set of nodes $\{1,....,k\}$.

$P_k(i,j)$ = the last intermediate node on the shortest path from node *i* to node *j* using $\{1,....,k\}$ as intermediate node.

The dynamic updating for the optimal path length and its corresponding labeling are:

$$f_k(i,j)=\min\{f_{k-1}(i,j),\ f_{k-1}(i,k)+f_{k-1}(k,j)\},$$

$$P_k(i,j)=\begin{cases} P_{k-1}(i,j) & \text{if k is not on shortest path from i to j using \{1,...,k\}} \\ P_{k-1}(k,j) & \text{otherwise.} \end{cases}$$

We are now ready to give the steps of the algorithm.

Algorithm: A dynamic programming method for computing a shortest path in a fuzzy network $G=(V,A)$, where V is the set of nodes with $|V|=N$, and A is the set of arcs.

Step 1: Let *k=0* and $\tilde{f}_k(i,j)=\tilde{d}_{ij}$, for all $(i,j)\in A$, $\tilde{f}_k=(i,j)=\infty$, for all $(i,j)\notin A$. **If** an arc exists from node *i* to node *j* **then** let $P_k(i,j)=i$.

Step 2: Let $k=k+1$.

Do the following steps for $i=1,2,3,....,N,\ j=1,2,3,....,N,\ i\neq j.$

2.1 Compute the value of $f_k(i,j)=\min\left[f_{k-1}(i,j),\ f_{k-1}(i,k)+f_{k-1}(k,j)\right]$ (for the addition, our proposed method discussed in Subection 2.2 and for comparison of fuzzy numbers the $D_{p,q}$ method of Subection 3.1 are applied).

2.2 If node *k* is not on the shortest path using nodes $\{1,2,...,k\}$ as intermediate nodes, **then** let $P_k(i,j)=P_{k-1}(i,j)$ **else** let $P_k(i,j)=P_{k-1}(k,j)$

Step 3: **If** $k<N$ **then go to Step 2.**

Step 4: Obtain the shortest path using the $P_k(i,j)$. If $f_N(i,j)=\infty$, then there is no path between *i* and *j*. The shortest path from node *i* to *j*, if it exists, is identified backwards and read by the nodes: *j*, $P_N(i,j)=k$ followed by $P_N(i,k),....,P_N(i,l)=i$, where *l* is the node immediately after *i* in the path.

3.3. Termination and Complexity of the Algorithm

The proposed algorithm terminates after *N* outer iterations corresponding to *k*. A total of *N(N-1)*^{2} additions and comparisons are needed for every *k*. For each addition, *n* fuzzy additions for the a_i-cuts should be performed resulting in *2n(N)(N-1)*^{2} additions. For comparisons, we have *(2n+1)N(N-1)*^{2} additions and *(2n+1) N(N-1)*^{2} multiplications using (7). Therefore, the total needed operations are *(6n+2) N(N-1)*^{2} additions and multiplications, with *N(N-1)*^{2} comparison

4. Comparative Examples

Here, we illustrate examples for a comparison of our proposed method and two other approaches.

Example 1:

Consider the following network Figure 4 considered by Chuang and Kung [30]. The triangular arc lengths are presented in Table 1. The results obtained by the approach ($\tilde{f}_6$ and P_6) of Chuang and Kung [30] are shown in Tables 2 and 3.

The shortest path and the corresponding length using the proposed approach in Chuang and Kung [30] are reported below:

$$\text{Shortest path from 1 to 6}: 1\rightarrow 2\rightarrow 4\rightarrow 6.$$

$$\text{Shortest path length from 1 to 6}: (177,195,256)$$

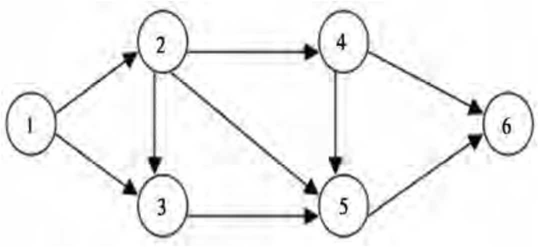


Figure 4. The network for Example 1.

Table 1. The arc lengths for example 1.

Arc	lengths	Arc	lengths	Arc	lengths
(1,2)	(33,45,50)	(1,3)	(42,57,61)	(2,3)	(50,52,61)
(2,4)	(56,58,72)	(2,5)	(51,79,85)	(3,5)	(43,55,60)
(4,5)	(32,40,46)	(4,6)	(88,92,134)	(5,6)	(75,110,114)

Table 2. The $\tilde{f}_6$ values obtained by the approach of Chuang and Kung.

i/j	1	2	3	4	5	6
1	-	(33,45,50)	(42,57,61)	(89,103,122)	(85,112,121)	(177,195,256)
2	-	-	(50,52,61)	(56,58,72)	(51,79,85)	(144,150,206)
3	-	-	-	-	(43,55,60)	(118,165,174)
4	-	-	-	-	(32,40,46)	(88,92,134)
5	-	-	-	-	-	(75,110,114)
6	-	-	-	-	-	-

Table 3. The P_6 values obtained by the approach of Chuang and Kung.

i/j	1	2	3	4	5	6
1	-	1	1	2	3	4
2	-	-	2	2	2	4
3	-	-	-	-	3	5
4	-	-	-	-	4	4
5	-	-	-	-	-	5
6	-	-	-	-	-	-

Here, we solve the same problem using our proposed algorithm givenin Subection 3.2 using the ranking method of Sadeghpour-Gildeh and Gien [27]. The results of the proposed approach for $\tilde{f}_6$ and P_6 are given in Tables 4 and 5.

Here, the shortest path obtained and the corresponding length are exactly the same as the ones we obtained by the approach of Chuang and Kung [30].

Example 2:

Consider the following network Figure 5 considered by Hernandes *et al.* [32]. The fuzzy triangular arc lengths are given in Table 6. The results ($\tilde{f}_6$ and P_6) for the approach of Hernandes *et al.* [32] are given in Tables 7 and 8.

Table 4. The $\tilde{f}_6$ values obtained by our proposed algorithm

i/j	1	2	3	4	5	6
1	-	(33,45,50)	(42,57,61)	(89,103,122)	(85,112,121)	(177,195,256)
2	-	-	(50,52,61)	(56,58,72)	(51,79,85)	(144,150,206)
3	-	-	-	-	(43,55,60)	(118,165,174)
4	-	-	-	-	(32,40,46)	(88,92,134)
5	-	-	-	-	-	(75,110,114)
6	-	-	-	-	-	-

The shortest path and the corresponding length using the proposed approach of Hernandes *et al.* [32] are reported below:

Shortest path from 1 to 11 : $1 \rightarrow 9 \rightarrow 7 \rightarrow 11$.

Shortest path length from 1 to 11 : $(860, 902, 990)$

We solved the same problem using our proposed algorithm of Subsection 3.2 using the ranking method of Sadeghpour-Gildeh and Gien [27]. The results of our proposed approach ($\tilde{f}_{11}$ and P_{11}) are given in Tables 9 and 10.

The shortest path and the corresponding length are reported below:

Shortest path from 1 to 11 : $1 \rightarrow 9 \rightarrow 7 \rightarrow 11$.

Shortest path length from 1 to 11 : $(840, 882, 990)$.

Clearly, the proposed algorithm computes almost the same solution as obtained by Hernandes *et al.* [32].

Table 5. The P_6 values obtained by our proposed algorithm.

i/j	1	2	3	4	5	6
1	-	1	1	2	3	4
2	-	-	2	2	2	4
3	-	-	-	-	3	5
4	-	-	-	-	4	4
5	-	-	-	-	-	5
6	-	-	-	-	-	-

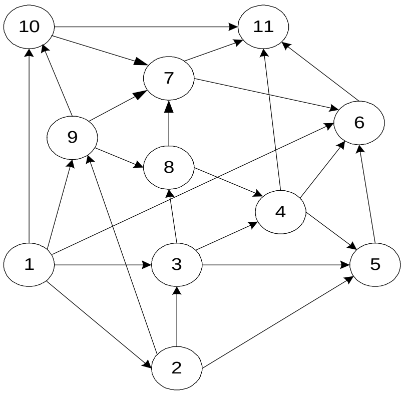


Figure 5. The network for Example 2.

Table 6. The arc lengths for Example 2.

Arc	lengths	Arc	lengths	Arc	lengths
(1,2)	(800,820,840)	(3,5)	(730,748,770)	(8,4)	(710,730,735)
(1,3)	(350,361,370)	(3,8)	(425,443,465)	(8,7)	(230,242,255)
(1,6)	(650,677,683)	(4,5)	(190,199,210)	(9,7)	(120,130,150)
(1,9)	(290,300,350)	(4,6)	(310,340,360)	(9,8)	(130,137,145)
(1,10)	(420,450,470)	(4,11)	(710,740,770)	(9,10)	(230,242,260)
(2,3)	(180,186,193)	(5,6)	(610,660,690)	(10,7)	(330,342,350)
(2,5)	(495,510,525)	(6,11)	(230,242,260)	(10,11)	(1250,1310,1440)
(2,9)	(900,930,960)	(7,6)	(390,410,440)	(3,4)	(650,667,883)
(7,11)	(450,472,490)				

Table 7. The $\tilde{f}_{11}(i, j)$ values obtained by Hernandes *et al*.

i/j	1	2	3	4	5	6	7
1	-	(800,820,840)	(350,361,370)	(1000,1028,1253)	(1080,1109,1140)	(650,677,683)	(410,430,500)
2	-	-	(180,186,193)	(830,853,1076)	(495,510,525)	(1105,1170,1215)	(835,871,913)
3	-	-	-	(650,667,883)	(730,748,770)	(960,1007,1243)	(655,685,720)
4	-	-	-	-	(190,199,210)	(310,340,360)	-
5	-	-	-	-	-	(610,660,690)	-
6	-	-	-	-	-	-	-
7	-	-	-	-	-	(390,410,440)	-
8	-	-	-	(710,730,735)	(900,929,945)	(620,652,695)	(230,242,255)
9	-	-	-	(840,867,880)	(1030,1066,1090)	(510,540,590)	(120,130,150)
10	-	-	-	-	-	(720,752,790)	(330,342,350)
11	-	-	-	-	-	-	-

Table 7. continued.

i/j	8	9	10	11
1	(420,437,495)	(290,300,350)	(420,450,470)	(860,902,990)
2	(605,629,658)	(900,930,960)	(1130,1172,1220)	(1285,1343,1403)
3	(425,443,465)	-	-	(1105,1157,1210)
4	-	-	-	(540,582,620)
5	-	-	-	(840,902,950)
6	-	-	-	(230,242,260)
7	-	-	-	(450,472,490)
8	-	-	-	(680,714,745)
9	(130,137,145)	-	(230,242,260)	(570,602,640)
10	-	-	-	(780,814,840)
11	-	-	-	-

Example 3: A wireless sensor network

Consider a mobile service company which handles 23 geographical centers. A configuration of a telecommunication network is presented in Figure 6. Assume that the distance between any two centers is a trapezoidal fuzzy number (the arc lengths are given in Table 11). The company wants to find a shortest path for an effective message flow amongst the centers.

The results obtained by our approach ($\tilde{f}_{23}$ and P_{23}) are given in Tables 12 and 13.

The shortest path and the corresponding length are reported below:

Shortest path from 1 to 23 : $1 \rightarrow 5 \rightarrow 11 \rightarrow 17 \rightarrow 21 \rightarrow 23$.

$$\text{Shortest path length from 1 to 23}:(38,49,58,65)\cdot$$

5. Discussion

We can also apply the proposed algorithm to networks having possibly a mixture of different fuzzy numbers as arc lengths. To see how the steps of the proposed algorithm are carried out on such networks, a small sized mixed fuzzy network with 4 nodes as shown in Figure 7 is considered, where the arc lengths are considered to be a mixture of trapezoidal and normal fuzzy numbers.

Example 4: Consider the mixed fuzzy network in Figure 4 with four nodes and five arcs having two trapezoidal and three normal arc lengths as specified in Table 14.

Step 1: We gain the $\tilde{f}_k(i,j)=\tilde{d}_{ij}$ for *k=0* as specified in Table 14.

Table 8. The $P_{11}(i,j)$ values obtained by Hernandes *et al*.

i/j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	-	1	1	3	3	1	9	9	1	1	9
2	-	-	2	3	2	5	8	3	2	9	8
3	-	-	-	3	3	4	8	3	-	-	8
4	-	-	-	-	4	4	-	-	-	-	6
5	-	-	-	-	-	5	-	-	-	-	6
6	-	-	-	-	-	-	-	-	-	-	6
7	-	-	-	-	-	7	-	-	-	-	7
8	-	-	-	8	4	7	8	-	-	-	7
9	-	-	-	8	8	7	9	9	-	9	7
10	-	-	-	-	-	7	10	-	-	-	7
11	-	-	-	-	-	-	-	-	-	-	-

Table 9. The $\tilde{f}_{11}(i,j)$ values obtained by our proposed algorithm.

i/j	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	-	(800,820,840)	(350,361,370)	(1000,1028,1253)	(1080,1109,1140)	(650,677,683)	(410,430,500)
2	-	-	(180,186,193)	(830,853,1076)	(495,510,525)	(1105,1170,1215)	(835,871,913)
3	-	-	-	(650,667,883)	(730,748,770)	(960,1007,1243)	(655,685,720)
4	-	-	-	-	(190,199,210)	(310,340,360)	-
5	-	-	-	-	-	(610,660,690)	-
6	-	-	-	-	-	-	-
7	-	-	-	-	-	(390,410,440)	-
8	-	-	-	(710,730,735)	(900,929,945)	(620,652,695)	(230,242,255)
9	-	-	-	(840,867,880)	(1030,1066,1090)	(510,540,590)	(120,130,150)
10	-	-	-	-	-	(720,752,790)	(330,342,350)
11	-	-	-	-	-	-	-

Table 9. continued.

i/j	**8**	**9**	**10**	**11**
1	(420,437,495)	(290,300,350)	(420,450,470)	(840,882,990)
2	(605,629,658)	(900,930,960)	(1130,1172,1220)	(1265,1323,1403)
3	(425,443,465)	-	-	(1085,1137,1210)
4	-	-	-	(540,582,620)
5	-	-	-	(840,902,950)
6	-	-	-	(230,242,260)
7	-	-	-	(430,452,490)
8	-	-	-	(660,694,745)
9	(130,137,145)	-	(230,242,260)	(550,582,640)
10	-	-	-	(760,794,840)
11	-	-	-	-

Table 10. The $P_{11}(i,j)$ values obtained by our proposed algorithm.

i/j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	-	1	1	3	3	1	9	9	1	1	9
2	-	-	2	3	2	5	8	3	2	9	8
3	-	-	-	3	3	4	8	3	-	-	8
4	-	-	-	-	4	4	-	-	-	-	6
5	-	-	-	-	-	5	-	-	-	-	6
6	-	-	-	-	-	-	-	-	-	-	6
7	-	-	-	-	-	7	-	-	-	-	7
8	-	-	-	8	4	7	8	-	-	-	7
9	-	-	-	8	8	7	9	9	-	9	7
10	-	-	-	-	-	7	10	-	-	-	7
11	-	-	-	-	-	-	-	-	-	-	-

Table 11. The arc lengths for Example 3.

Arc	lengths	Arc	lengths	Arc	lengths
(1,2)	(12,13,15,17)	(1,3)	(9,11,13,15)	(1,4)	(8,10,12,13)
(1,5)	(7,8,9,10)	(2,6)	(5,10,15,16)	(2,7)	(6,11,11,13)
(3,8)	(10,11,16,17)	(4,7)	(17,20,22,24)	(4,11)	(6,10,13,14)
(5,8)	(6,9,11,13)	(5,11)	(7,10,13,14)	(5,12)	(10,13,15,17)
(6,9)	(6,8,10,11)	(6,10)	(10,11,14,15)	(7,10)	(9,10,12,13)
(7,11)	(6,7,8,9)	(8,12)	(5,8,9,10)	(8,13)	(3,5,8,10)
(9,16)	(6,7,9,10)	(10,16)	(12,13,16,17)	(10,17)	(15,19,20,21)
(11,14)	(8,9,11,13)	(11,17)	(6,9,11,13)	(12,14)	(13,14,16,18)
(12,15)	(12,14,15,16)	(13,15)	(10,12,14,15)	(13,19)	(17,18,19,20)
(14,21)	(11,12,13,14)	(15,18)	(8,9,11,13)	(15,19)	(5,7,10,12)
(16,20)	(9,12,14,16)	(17,20)	(7,10,11,12)	(17,21)	(6,7,8,10)
(18,21)	(15,17,18,19)	(18,22)	(3,5,7,9)	(18,23)	(5,7,9,11)
(19,22)	(15,16,17,19)	(20,23)	(13,14,16,17)	(21,23)	(12,15,17,18)
(22,23)	(4,5,6,8)				

Table 12. The $\tilde{f}_{23}(i, j)$ values obtained by our algorithm.

i/j	1	2	3	4	5	6	7	8
1	-	(12,13,15,17)	(9,11,13,15)	(8,10,12,13)	(7,8,9,10)	(17,23,30,33)	(18,24,26,30)	(13,17,20,23)
2	-	-	-	-	-	(5,10,15,16)	(6,11,11,13)	-
3	-	-	-	-	-	-	-	(10,11,16,17)
4	-	-	-	-	-	-	(17,20,22,24)	-
5	-	-	-	-	-	-	-	(6,9,11,13)
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-

Table 12. continued.

i/j	9	10	11	12	13	14	15
1	(23,31,40,44)	(27,34,38,43)	(14,18,22,24)	(17,21,24,27)	(16,22,28,33)	(22,27,33,37)	(29,35,39,43)
2	(11,18,25,27)	(15,21,23,26)	(12,18,19,22)	-	-	(20,27,30,35)	-
3	-	-	-	(15,19,25,27)	(13,16,24,27)	(28,33,41,45)	(23,28,38,42)
4	-	(26,30,34,37)	(6,10,13,14)	-	-	(14,19,24,27)	-
5	-	-	(7,10,13,14)	(10,13,15,17)	(9,14,19,23)	(15,19,24,27)	(22,27,30,33)
6	(6,8,10,11)	(10,11,14,15)	-	-	-	-	-
7	-	(9,10,12,13)	(6,7,8,9)	-	-	(14,16,19,22)	-
8	-	-	-	(5,8,9,10)	(3,5,8,10)	(18,22,25,28)	(13,17,22,25)
9	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-
11	-	-	-	-	-	(8,9,11,13)	-
12	-	-	-	-	-	(13,14,16,18)	(12,14,15,16)
13	-	-	-	-	-	-	(10,12,14,15)
14	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-

Table 12. continued.

i/j	16	17	18	19	20	21	22	23
1	(29,38,49,54)	(20,27,33,37)	(37,44,50,56)	(33,40,47,53)	(27,37,44,49)	(26,34,41,47)	(40,49,57,65)	(38,49,58,65)
2	(17,25,34,37)	(18,27,30,35)	-	-	(25,37,41,47)	(24,34,38,45)	-	(36,49,55,63)
3	-	-	(31,37,49,55)	(30,34,43,47)	-	(39,45,54,59)	(34,42,56,64)	(36,44,58,66)
4	(38,43,50,54)	(12,19,24,27)	-	-	(19,29,35,39)	(18,26,32,37)	-	(30,41,49,55)
5	-	(13,19,24,27)	(30,36,41,46)	(26,32,38,43)	(20,29,35,39)	(19,26,32,37)	(33,41,48,55)	(31,41,49,55)
6	(12,15,19,21)	(25,30,34,36)	-	-	(21,27,33,37)	(31,37,42,46)	-	(34,41,49,54)
7	(21,23,28,30)	(12,16,19,22)	-	-	(19,26,30,34)	(18,23,27,32)	-	(30,38,44,50)
8	-	-	(21,26,33,38)	(20,23,27,30)	-	(29,34,38,42)	(24,31,40,47)	(26,33,42,49)
9	(6,7,9,10)	-	-	-	(15,19,23,26)	-	-	(28,33,39,43)
10	(12,13,16,17)	(15,19,20,21)	-	-	(21,25,30,33)	(21,26,28,31)	-	(34,39,46,50)
11	-	(6,9,11,13)	-	-	(13,19,22,25)	(12,16,19,23)	-	(24,31,36,41)
12	-	-	(20,23,26,29)	(17,21,25,28)	-	(24,26,29,32)	(23,28,33,38)	(25,30,35,40)
13	-	-	(18,21,25,28)	(17,18,19,20)	-	(33,38,43,47)	(21,26,32,37)	(23,28,34,39)
14	-	-	-	-	-	(11,12,13,14)	-	(23,27,30,32)
15	-	-	(8,9,11,13)	(5,7,10,12)	-	(23,26,29,32)	(11,14,18,22)	(13,16,20,24)
16	-	-	-	-	(9,12,14,16)	-	-	(22,26,30,33)
17	-	-	-	-	(7,10,11,12)	(6,7,8,10)	-	(18,22,25,28)
18	-	-	-	-	-	(15,17,18,19)	(3,5,7,9)	(5,7,9,11)
19	-	-	-	-	-	-	(15,16,17,19)	(19,21,23,27)
20	-	-	-	-	-	-	-	(13,14,16,17)
21	-	-	-	-	-	-	-	(12,15,17,18)
22	-	-	-	-	-	-	-	(4,5,6,8)
23	-	-	-	-	-	-	-	-

Table 13. The $P_{23}(i, j)$ values obtained by our algorithm.

i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-	1	1	1	1	2	2	5	6	7	5	5	8	11	12	9	11	15	13	17	17	18	21
2	-	-	-	-	-	2	2	-	6	7	7	-	-	11	-	9	11	-	-	17	17	-	21
3	-	-	-	-	-	-	-	3	-	-	-	8	8	12	13	-	-	15	13	-	14	18	18
4	-	-	-	-	-	-	4	-	-	7	4	-	-	11	-	10	11	-	-	17	17	-	21
5	-	-	-	-	-	-	-	5	-	-	5	5	8	11	12	-	11	15	13	17	17	18	21
6	-	-	-	-	-	-	-	-	6	6	-	-	-	-	-	9	10	-	-	16	17	-	20
7	-	-	-	-	-	-	-	-	-	7	7	-	-	11	-	10	11	-	-	17	17	-	21
8	-	-	-	-	-	-	-	-	-	-	-	8	8	12	13	-	-	15	13	-	14	18	18
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	-	-	-	16	-	-	20
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10	10	-	-	16	17	-	20
11	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-	-	11	-	-	17	17	-	21
12	-	-	-	-	-	-	-	-	-	-	-	-	-	12	12	-	-	15	15	-	14	18	18
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13	-	-	15	13	-	18	18	18
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	14	-	21
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15	15	-	18	18	18
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	16	-	-	20
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	17	17	-	21
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	18	18	18
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	19	22
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	21
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	22
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

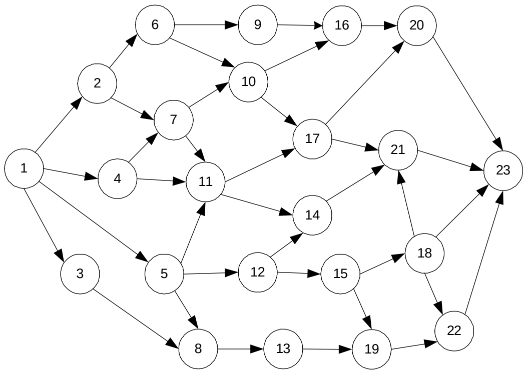


Figure 6. The telecommunication network proposed for Example 3.

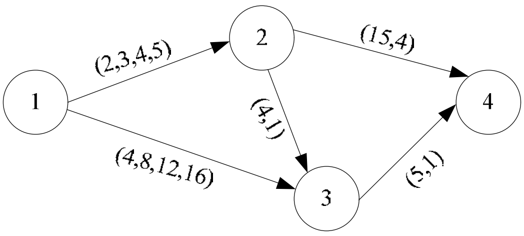


Figure 7. A small sized network having various fuzzy arc lengths.

Table 14. The $\tilde{f}_0(i,j)$ matrix for *k*=0.

i/j	1	2	3	4
1	-	(2,3,4,5)	(4,8,12,16)	2
2	-	-	(4,1)	(15,4)
3	-	-	-	(5,1)
4	-	-	-	-

Therefore, with $P_k(i,j) = i$, we have Table 15.

Step 2: Here, we consider *k*=1 and compute the value of $f_k(i,j) = \min[f_{k-1}(i,j), f_{k-1}(i,k) + f_{k-1}(k,j)]$. The result is shown in Table 16.

Therefore, for $P_k(i,j) = i$, we have Table 17.

Table 15. The $P_0(i,j)$ matrix for *k*=0.

i/j	1	2	3	4
1	-	1	1	-
2	-	-	2	2
3	-	-	-	3
4	-	-	-	-

Table 16. The $\tilde{f}_1(i,j)$ matrix for *k*=1.

i/j	1	2	3	4
1	-	(2,3,4,5)	(4,8,12,16)	-
2	-	-	(4,1)	(15,4)
3	-	-	-	(5,1)
4	-	-	-	-

Table 17. The $P_1(i,j)$ matrix for *k*=1.

i/j	1	2	3	4
1	-	1	1	-
2	-	-	2	2
3	-	-	-	3
4	-	-	-	-

If node *k* is not on the shortest path using $\{1,2,...,k\}$ as intermediate nodes, then we consider $P_k(i,j) = P_{k-1}(i,j)$, otherwise we let $P_k(i,j) = P_{k-1}(i,k)$. We now report the results obtained for other values of *k* in Tables 18-23. Note that, the sets V_i and W_i are the points obtained by a -cut additions, where the *V* and *W* values are obtained by the a_i-cuts considering *n=10.* It includes 10 points for the $a_{a_i}^-$ and 10 points for the $a_{a_i}^+$:

V_1={(4.58257,10.4174), (4.93136,10.0686),
(5.20274,9.79726), (5.44277,9.55723),
(5.66745,9.33255), (5.88528,9.11472),
(6.10278,8.89722), (6.32762,8.67238),
(6.57541,8.42459), (7,8)}

W_1={(11.0303,25.9697), (12.1255,24.8745),
(12.911,24.089), (13.5711,23.4289),
(14.1698,22.8302), (14.7411,22.2589),
(15.3111,21.6889), (15.9105,21.0895),
(16.6016,20.3984), (18,19)}

V_2={(4.58257,10.4174),(4.93136,10.0686),
(5.20274,9.79726), (5.44277,9.55723),
(5.66745,9.33255), (5.88528,9.11472),
(6.10278,8.89722), (6.32762,8.67238),
(6.57541,8.42459), (7,8)}

W_2={(8.06515,16.9349), (8.66273,16.3373),
(9.10549,15.8945), (9.48554,15.5145),
(9.83489,15.1651), (10.1706,14.8294),
(10.5056,14.4944), (10.8552,14.1448),
(11.2508,13.7492), (12,13)}

Table 18. The $\tilde{f}_2(i,j)$ matrix for *k*= 2.

i/j	1	2	3	4
1	-	(2,3,4,5)	V_1	W_1
2	-	-	(4,1)	(15,4)
3	-	-	-	(5,1)
4	-	-	-	-

Table 19. The $P_2(i,j)$ matrix for *k*= 2.

i/j	1	2	3	4
1	-	1	2	2
2	-	-	2	2
3	-	-	-	3
4	-	-	-	-

Table 20. The $\tilde{f}_3(i,j)$ matrix for *k*= 3

i/j	1	2	3	4
1	-	(2,3,4,5)	V_2	W_2
2	-	-	(4,1)	(9,2)
3	-	-	-	(5,1)
4	-	-	-	-

Table 21. The $P_3(i,j)$ matrix for *k*= 3.

i/j	1	2	3	4
1	-	1	2	3
2	-	-	2	3
3	-	-	-	3
4	-	-	-	-

Table 22. The $\tilde{f}_4(i,j)$ matrix for *k*= 4.

i/j	**1**	**2**	**3**	**4**
1	-	(2,3,4,5)	V_3	W_3
2	-	-	(4,1)	(9,2)
3	-	-	-	(5,1)
4	-	-	-	-

Table 23. The $P_4(i,j)$ matrix for *k*= 4.

i/j	**1**	**2**	**3**	**4**
1	-	1	2	3
2	-	-	2	3
3	-	-	-	3
4	-	-	-	-

V_3={(4.58257,10.4174), (4.93136,10.0686), (5.20274,9.79726), (5.44277,9.55723), (5.66745,9.33255), (5.88528,9.11472), (6.10278,8.89722), (6.32762,8.67238), (6.57541,8.42459), (7,8)}

W_3={(8.06515,16.9349), (8.66273,16.3373), (9.10549,15.8945), (9.48554,15.5145), (9.83489,15.1651), (10.1706,14.8294), (10.5056,14.4944), (10.8552,14.1448), (11.2508,13.7492), (12,13)}

Finally, when $k = N$, we identify the shortest path as follows:

Shortest path from 1 to 4: 1-2-3-4.

Shortest path length from 1 to 4:

(8.06515,16.9349),(8.66273,16.3373),(9.10549,15.8945),(9.48554,15.5145),(9.83489,15.1651),(10.1706,14.8294),(10.5056,14.4944),(10.8552,14.1448),(11.2508,13.7492),(12,13)

6. Conclusions

We presented a novel approach for computing a shortest path in a mixed network having various fuzzy arc lengths. First, we developed a new technique for the addition of various fuzzy numbers in a path using α-cuts. Then, we applied a dynamic programming method for finding a shortest path in the network, using a recently proposed distance function to compare fuzzy numbers in the proposed algorithm. Four comparative examples were worked out to illustrate the applicability of our proposed approach as compared to two other methods in the literature as well as demonstrate the additional novel feature offered by our algorithm to find a fuzzy shortest path in mixed fuzzy networks having various settings for the fuzzy arc lengths.

7. References

[1] R. E. Moore, “Method and application of interval analysis,” SIAM, Philadelphia, 1997. M. Delgado, J. L. Verdegay, and M. A. Vila, “A procedure for ranking fuzzy numbers using fuzzy relations,” Fuzzy Sets and Systems, Vol. 26, pp. 49–62, 1988.

[2] M. Delgado, J. L. Verdegay, and M. A. Vila, “A procedure for ranking fuzzy numbers using fuzzy relations,” Fuzzy Sets and Systems, Vol. 26, pp. 49–62, 1988.

[3] H. Ishibuchi and H. Tanaka, “Multi-objective programming in optimization of the interval objective function,” European Journal of Operational Research, Vol. 48, pp. 219–225, 1990.

[4] J. K. Sengupta, “Optimal decision under uncertainty,” Springer, New York, 1981.

[5] T. Shaocheng, “Interval number and fuzzy number linear programming,” Fuzzy Sets and Systems, Vol. 66, pp. 301–306. 1994.

[6] A. Sengupta, T. K. Pal, “Theory and methodology on comparing interval numbers,” European Journal of Operational Research, Vol. 127, 28–43, 2000.

[7] D. Dubois and H. Prade, “Ranking fuzy numbers in the setting of possibility theory,” Information Sciences, Vol. 30, pp. 183–224, 1983.

[8] S. H. Chen, “Ranking fuzzy numbers with maximizing set and minimizing set,” Fuzzy Sets and Systems, Vol. 17, pp. 113–129, 1985.

[9] G. Bortolan and R. Degani, “A review of some methods for ranking fuzzy subsets,” Fuzzy Sets and Systems, Vol. 15, pp. 1–19, 1985.

[10] C.-H. Cheng, “A new approach for ranking fuzzy numbers by distance method,” Fuzzy Sets and Systems, Vol. 95, pp. 307–317, 1998.

[11] M. Blue, B. Bush, and J. Puckett, “Unified approach to fuzzy graph problems,” Fuzzy Sets and Systems, Vol. 125, pp. 355–368, 2002.

[12] L. T. Koczy, “Fuzzy graphs in the evaluation and optimization of networks,” Fuzzy Sets and Systems, Vol. 46, pp. 307–319, 1992.

[13] C. M. Klein, “Fuzzy Shortest Paths,” Fuzzy Sets and Systems, Vol. 39, pp. 27–41, 1991.

[14] Y. Li, M. Gen, and K. Ida, “Solving fuzzy shortest path problem by neural networks,” Computers Industrial Engineering, Vol. 31, pp. 861–865, 1996.

[15] K. Lin and M. Chen, “The fuzzy shortest path problem and its most vital arcs,” Fuzzy Sets and Systems, Vol. 58, pp. 343–353, 1994.

[16] S. Okada and M. Gen, “Order relation between intervals and its application to shortest path problem,” Computers & Industrial Engineering, Vol. 25, 147–150, 1993.

[17] S. Okada and M. Gen, “Fuzzy shortest path problem,” Computers & Industrial Engineering, Vol. 27, pp. 465–468, 1994.

[18] S. Okada and T. Soper, “A shortest path problem on a network with fuzzy arc lengths,” Fuzzy Sets and Systems, Vol. 109, pp. 129–140, 2000.

[19] L. A. Zadeh, “Fuzzy sets as a basis for a theory of possibility,” Fuzzy Sets and Systems, Vol. 1, pp. 3–28, 1978.

[20] R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan,

"Faster algorithms for the shortest path problem," Technical Report CS-TR-154-88, Department of Computer Science, Princeton University, 1988.

[21] A. Andersson, T. Hagerup, S. Nilsson, and R. Raman, "Sorting in linear time?" In Proceedings of 27th ACM Symposium on Theory of Computing, pp. 427–436, 1995.

[22] A. V. Goldberg, "Scaling algorithms for the shortest paths problem," In: Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms, pp. 222–231, 1993.

[23] Y. Asano and H. Imai, "Practical efficiency of the linear-time algorithm for the single source shortest path problem," Journal of the Operations Research Society of Japan, Vol. 43, pp. 431–447, 2000.

[24] Y. Han, "Improved algorithm for all pairs shortest paths," Information Processing Letters, Vol. 91, pp. 245–250, 2004.

[25] S. Saunders, T. Takaoka, "Improved shortest path algorithms for nearly acyclic graphs," Electronic Notes in Theoretical Computer Science, Vol. 42, pp. 1–17, 2001.

[26] S. Chanas, "Fuzzy optimization in networks," In: J. Kacprzyk, S. A. Orlovski (Eds.), Optimization Models Using Fuzzy Sets and Possibility Theory, D. Reidel, Dordrecht, pp. 303–327, 1987.

[27] B. Sadeghpour-Gildeh, D. Gien, Q. La Distance-Dp, *et al*. "Cofficient de Corrélation entre deux Variables Aléatoires floues," Actes de LFA'01, pp. 97–102, Monse-Belgium, 2001.

[28] I. Mahdavi, R. Nourifar, A. Heidarzade, and N. Mahdavi-Amiri, "A dynamic programming approach for finding shortest chains in a fuzzy network," Applied Soft Computing, Vol. 9, No. 2, pp. 503–511, 2009.

[29] R. W. Floyd, Algorithm 97, Shortest path, CACM 5, pp. 345, 1962.

[30] T.-N. Chuang, and J.-Y. Kung, The fuzzy shortest path length and the corresponding shortest path in a network, Computers & Operations Research, Vol. 32, 1409–1428, 2005.

[31] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithm," J. ACM 34, Vol. 3, pp. 596–615, 1987.

[32] F. Hernandes, M. T. Lamata, J. L. Verdegay, and A. Yamakami, "The shortest path problem on networks with fuzzy parameters," Fuzzy Sets and Systems, Vol. 158, pp. 1561–1570, 2007.

Wireless Sensor Network, **2010, February, 161-167**
doi:10.4236/wsn.2010.22021 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Tree Based Energy and Congestion Aware Routing Protocol for Wireless Sensor Networks

Amir Hossein Mohajerzadeh, Mohammad Hossien Yaghmaee
Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
E-mail: *ah.mohajerzadeh@stu-mail.um.ac.ir, hyaghmae@ferdowsi.um.ac.ir*
Received June 15, 2009; *revised October* 15, 2009; *accepted November* 17, 2009

Abstract

Wireless Sensor Networks (WSNs) have inherent and unique characteristics rather than traditional networks. They have many different constraints, such as computational power, storage capacity, energy supply and etc; of course the most important issue is their energy constraint. Energy aware routing protocol is very important in WSN, but routing protocol which only considers energy has not efficient performance. Therefore considering other parameters beside energy efficiency is crucial for protocols efficiency. Depending on sensor network application, different parameters can be considered for its protocols. Congestion management can affect routing protocol performance. Congestion occurrence in network nodes leads to increasing packet loss and energy consumption. Another parameter which affects routing protocol efficiency is providing fairness in nodes energy consumption. When fairness is not considered in routing process, network will be partitioned very soon and then the network performance will be decreased. In this paper a Tree based Energy and Congestion Aware Routing Protocol (TECARP) is proposed. The proposed protocol is an energy efficient routing protocol which tries to manage congestion and to provide fairness in network. Simulation results shown in this paper imply that the TECARP has achieved its goals.

Keywords: Congestion Aware, Energy Efficiency, Routing Protocol, Fairness, Tree Based Routing, Wireless Sensor Networks

1. Introduction

Wireless Sensor Networks have been noticed and researched in recent years. These networks are composed of hundreds or thousands of sensor nodes which have many different types of sensors [1]. Using their sensors, nodes collect information about their environment such as light, temperature, humidity, motion and etc [2]. Sensor nodes should send their collected data to determined nodes called Sink. The sink processes data and performs appropriate actions. Many different paths exist between each node and sink. Using routing protocol, nodes determine a path for sending data to sink. Similar to traditional networks, routing protocols in wireless sensor networks consider different parameters in their routing process depended on their application.

WSNs have inherent and unique characteristics compared with traditional networks [1,2]. These networks have many limitations such as computing power, storage space, communication range, energy supply and etc. Nodes have limited primary energy sources and in most of applications they are not rechargeable [3], therefore energy consumption is the most important factor in routing process for wireless sensor networks. Node’s energy is consumed due to using sensors, processing information and communicating with other nodes. Communications are the main element in energy consumption. Routing protocol directly affects communications volume; therefore energy aware routing protocols are very effective in decreasing energy consumption [4].

Routing protocols which only consider energy as their parameter are not efficient. In addition to energy efficiency, using other parameters makes routing protocol more efficient. For different applications, different parameters should be considered. One of the most important parameter is congestion management. Congestion occurrence leads to increasing packet loss and network energy consumption. Congestion occurs for different reasons in wireless networks; first, due to limited storage capacity in relay nodes. When a node receives packets more than its capacity, congestion will be occurred. Second, due to inherent shared wireless link, congestion occurred for similar reasons in wireless sensor networks. For example, when many nodes simultaneously decide to send packet using a shared medium, congestion will be occurred. Two main methods exist to manage congestion. Chen *et al*. [5] divided the techniques developed to address the problem of data congestion in WSN into two groups: congestion avoidance and congestion control. The former focuses on strategies to avoid congestion from happening and the latter works on removing congestion when it has occurred. In wireless sensor networks due to limitations in resources, avoiding congestion rather than controlling congestion is more reasonable. In congestion control mechanisms when the congestion is occurred, it will be controlled. This mechanism leads to consuming resources more. Generally even congestion control mechanism detect congestion earlier, more energy will be conserved. In this paper we consider congestion avoidance. In congestion avoidance mechanisms, by avoiding congestion occurrence, more energy is conserved. Different methods can be used to avoid congestion in wireless sensor networks. We use limited tree based routing. By definition many parameters about the created routing tree, congestion aware routing is achievable. For example, by limiting number of a node child, amount of traffic which forward to it, will be controllable.

One of the considerable parameters that affect the network energy consumption is providing fairness in nodes energy consumption. To provide fairness, network nodes' energy should be consumed equally. If one part of a network is used more than other parts, its energy will be dropped sooner than others then the network will be partitioned. If a network is partitioned, its energy consumption increases severely. Different routing strategies such as multi path, geographical, flat and hierarchical exist for WSNs. Each mentioned strategies have its own unique characteristics. Using different paths to send data to sink makes providing fairness more efficient.

Tree structure is used by routing protocols due to its inherent characteristics [6,7]. Regarding as in wireless sensor networks structure, in most of applications we have one sink and too many sender nodes, and tree structure is very popular. In many protocols, by limiting routing tree such as determining most number of nodes' Childs or the maximum depth of tree, limited tree structure is used in routing process.

In this paper, a hierarchical routing protocol which considers both energy and congestion as two main parameters in routing process is proposed. Our proposed protocol extends the routing approach in [8]. Simulation results show that TECARP catch its goals. TECARP consider two different traffics: high priority and low priority. Proposed protocol, uses best routes for high priority traffic and manages congestion, therefore we suggest TECARP for transmitting real time traffic. The rest of the paper is organized as follows: Section 2 summarizes related work, in Section 3, TECARP is discussed, Section 4 summarizes the simulation based evaluation of the TECARP routing protocol, and finally Section 5 concludes the paper.

2. Related Work

As mentioned before, energy consumption is the most important factor for routing protocols in WSNs. Different energy aware routing algorithms have been designed for wireless sensor networks. In [9] optimal energy consumption is the most important objective. Akkaya *et al*, propose [8] which is a well known algorithm for transmitting real time traffic in wireless sensor networks. It considers energy consumption in its routing procedure. It is a highly efficient and scalable protocol for sensor networks where the resources of each node are scarce. It is a Hierarchical routing algorithm. By determining real time data forwarding rate, it can support both real time and non real time traffics. Link cost function which is used by [8] is interesting.

Reactive Energy Decision Routing Protocol (REDRP) [10] is another routing algorithm for WSNs whose main goal is optimal energy consumption. This algorithm attempts to distribute traffic in the entire network fairly. Using this mechanism, it decreases total network energy consumption. REDRP is routing reactively, and uses residual node energy in routing procedure. It uses local information for routing, but nodes have a global ID which is unique for the entire network. This algorithm is divided into 4 steps. In the first step, the sink sends a control packet to all network nodes. The nodes estimate their distance to sink relatively by using this packet. The next step is route discovery. Routing is performed on demand in REDRP. This means that the routes are established reactively. After route establishment in route discovery step, data are forwarded to sink by using those routes. If a route is damaged, in route recovery step, it will be recovered or a new route will be established.

Different congestion control techniques have been proposed for wireless sensor networks [11,12]. In [11], CODA, an energy efficient congestion control scheme for sensor networks was proposed. CODA (Congestion Detection and Avoidance) comprises three mechanisms: 1) receiver-based congestion detection; 2) open-loop hop-by-hop backpressure; and 3) closed-loop multi-source regulation. CODA detects congestion based on queue length as well as wireless channel load at intermediate nodes. Furthermore it uses explicit congestion notification approach and also an AIMD rate adjustment technique. In [13], two traffic types are considered: high priority and low priority. It selects a special area in network called conzone. The nodes placed in conzone only forward high priority traffic and other network nodes forward other traffics. Reference [13] proposes two algorithms: CAR and MCAR. CAR is a network-layer solution to provide differentiated service in congested sensor networks. CAR also prevents severe degradation of service to low priority data by utilizing uncongested parts of the network. MCAR is primarily a MAC-layer mechanism used in conjunction with routing to provide mobile and lightweight conzones to address sensor networks with mobile high priority data sources and/or bursty high priority traffic. MCAR Compared with CAR has a smaller overhead but degrades the performance of LP data more aggressively.

3. Proposed Protocol

Proposed protocol is a hierarchical routing protocol. Generally in hierarchical routing protocols, routing process is divided into two main different phases. Each phase is done independently. First, routing intra cluster; in this phase packets are routed between sensor nodes and cluster heads. Second, routing inter cluster; in this phase packets are routed between cluster heads and the sink. TECARP focuses on routing intra cluster. Routing inter cluster is considered for future studies. TECARP is composed of 3 phases: network clustering, creating routing tree and data forwarding. In network clustering phase, network nodes are partitioned into different clusters. During this phase cluster nodes information are delivered to cluster head. In creating routing tree phase, a limited routing tree will be created. During this phase a routing table is created for each of cluster nodes. In data transmission phase, packets are forwarded using relay nodes routing table. During the time, depend on network cluster status (congestion, fairness and energy) node's routing table will be updated. In the rest of this section, these phases are discussed in details.

3.1. Network Clustering Phase

In this phase, network nodes are partitioned into different clusters. In proposed protocol, clustering is done using clustering mechanism presented in [14]. During the clustering phase, a cluster head also should be elected for each cluster. At the end of this phase, information about all of the cluster nodes is delivered to cluster head. Each node in cluster sends its own information to the sink directly. It is important to know that, this phase is done once; therefore direct communication between cluster nodes and the cluster head is negligible.

3.2. Creating Routing Tree Phase

In this phase, using information delivered to cluster node in the former phase. In a routing tree structure, for every cluster node a path to its cluster head is determined. Cluster head knows position of all nodes located in its cluster, and then in first step in this phase, cluster head evaluates link cost between every two nodes located in their communication range [8]. For determining link cost, proposed protocol uses Formula (1).

$$Cost_{ij} = \sum_{k=0}^{3} CF_k = c_0 \times \left(dist_{ij}\right)^L$$

$$+c_1 + c_2 + c_3 \times f\left(e_{ij}\right) \qquad (1)$$

- CF_0 (Communication Cost) = where c_0 is a weighting constant and the parameter L depends on the environment, and typically equals to 2. This factor reflects the cost of the wireless transmission power, which is directly proportional to the distance raised to some power L. The closer a node to the destination, the less its cost factor CF_0 and more attractive it is for routing.
- CF_1 (Energy stock) = this factor reflects the primary battery lifetime, which favors nodes with more energy. The more energy the node contains, the better it is for routing. Applicable in networks which have heterogeneous nodes.
- CF_2 (Sensing-state cost) = where c_2 is a constant added when the node j is in a sensing state. This factor does not favor selecting sensing-enabled nodes to serve as relays. It's preferred not to overload these nodes in order to keep functioning as long as possible.
- CF_3 (Error rate) = where f is a function of distance between nodes i and j and buffer size on node j (i.e. $dist_{ij}$ / $buffer\ \_\ size$). The links with high error rate will increase the cost function, thus will be avoided.

Cluster head using nodes' information, links cost and Dijkstra algorithm selects least cost route between every cluster node and cluster head. Using Dijkstra algorithm, route selected between every node and cluster head is optimum; and only one path is selected between each node and cluster head (only one optimal path is exist between each node and cluster head); therefore the set of all routes has a tree structure called routing tree. If a node uses selected least cost route for transmitting its traffic, network will consume least possible energy for its traffic. But, it is important to note here that, the least cost route is not always the best route. We discuss about providing fairness in network nodes energy consumption in Section 1. If always the path with least cost is selected to forward other nodes data, nodes which are located in mentioned path die so sooner than other nodes which located on paths with higher cost. Generally, if some parts of network die sooner than other parts, network will be partitioned. Partitioned network in comparison with normal network consumes energy more and has higher reliability. By using mechanism which provides fairness in network, network lifetime will be increased and then wireless sensor network can do its task more reliable and longer.

Wireless sensor networks are expensive networks. For decreasing costs, a network is used for more than one application. Each application has its own traffic, therefore in many cases, a sensor network should transmit different traffics with different priorities. The proposed protocol considers two traffic types: high priority and low priority traffic. Traffics based on their priority get network services. TECARP improves routing tree after constructing it; for each node, high priority traffic volume and the ability to forward other node data are determined. Based on mentioned parameters, most number of children for each node in routing tree to avoid congestion will be determined. A node's child is a node that selects former node as its next hop in its route to the sink. Usually a number in [3,6] is determined as the most number of node's children. After determining most number of node's children, routing tree is modified such that no node remains with more than determined maximum number of children. Cluster head has sufficient information about all the cluster routes, therefore it can find number of each node's children simply. The cluster head decreases number of the children of nodes which have children more than the most number of children, using mechanism will be discussed in the next paragraph.

Cluster head evaluates all of the cluster nodes and then chooses nodes that have children more than most number of children. For all of neighbors of each selected nodes, proposed protocol determines the following two parameters.

- Least cost route between node and sink (using one of its neighbors except former neighbor as next hop).
- Number of children of new selected next hop neighbor.

Then among the children of each node, the one which has a neighbor with fewer children than the most number of children and least cost route to sink will be selected. Then the selected child is modified and in future it will be the child of its new qualified neighbor. In some situations, the child with appropriate conditions dose not exist, therefore exceptionally a node with children more than the most number of children is accepted. Here a routing tree with appropriate structure is ready to make for each cluster. But only cluster heads know the routes and routing tree, because they made it by itself and other nodes do not participate in mentioned process.

After selecting the best route and determining the number of children for every node, cluster head creates a routing table for all cluster nodes. A special record in routing table is considered for its best route to cluster head. In addition to best route, many paths which have lower costs in comparison to other paths are also selected to create routing table. For each of selected paths, a record is considered in routing table. Routing table has following fields: ID, residual energy, number of children, cost and average queue length. After constructing routing table, cluster head directly sends each node routing table to it.

In this phase a routing tree is created for each cluster. As mentioned before, maximum number of children for each node in cluster is constant (of course softly). Therefore traffic volume which enters to nodes is managed and controlled. This leads to decrease congestion in network; in other words, by using proposed mechanism which explained in this section congestion is avoided as much as possible. From another point of view, a routing table based on a routing tree is created for each node in cluster. Created routing tree has some records. The node depends on network conditions (congestion and neighbors residual energy), can use each of records to forward its data. Therefore all of routes participate in forwarding packets. Using different routes leads to use different neighbors to forward data.

3.3. Data Transmission Phase

At the end of the former phase, all the nodes have a routing table. As mentioned before, the proposed routing protocol considers two traffic types: high priority and low priority. The main goal of this phase is to determine next hop for each arrival packet at each node. In the rest of this section, the routing process which is done in each node when it receives packets with different priorities is discussed.

When a node receives a high priority packet, it performs following steps:

1) If special record in node routing table is active, this record will be selected. Otherwise Step 2 will be done. (The only special record in node routing table belongs to least cost route)

2) Among the records which have average queue length lower than threshold β, the record with lowest cost field value will be selected. If all the records have average queue length more than threshold β, Step 3 will be done.

3) Among the records, the one with the lowest cost field value will be selected.

After selecting the record, arrival packet will be sent to the node which is determined in record as next hop.

When a node receives a low priority packet, it performs the following steps:

1) Among the records which have average queue length lower than threshold β, the record with the biggest residual energy will be selected. If all of the records have average queue length more than threshold β, Step 2 will be done.

2) Among the records, the one with biggest residual energy will be selected.

After selecting the record, the arrival packet will be sent to the node which is determined in record as the next hop. Threshold β numbered by multiply a number in [0, 1] to node queue capacity.

Nodes' routing table should be updated periodically; otherwise they can not play their role effectively. When a node residual energy becomes less than threshold α, it broadcasts a message and informs its neighbors about its current condition. Neighbors receiving mentioned message, update the sender node's record in their routing table. Also the nodes should inform their neighbors about their average queue length. When a node queue length crosses threshold β (goes up or down the β), it should inform its neighbors. Threshold α numbered by multiply a number in [0, 1] to primary available energy.

To keep routing table update is necessary for proposed protocol. If routing table records not be updated, routing process can not be efficient, because its information is old. In other side, keeping routing table update is costly. For keeping routing table update, routing protocol should force nodes to send their current condition to its neighbors periodically. This process is costly. Therefore, there is a trade off between routing efficiency and node's energy consumption. The short period leads to the efficient routing and high energy consumption; the long period leads to the lower energy consumption and less efficient routing. We do many experiments to determine best values for parameters α, β.

4. Performance Evaluation

Akkaya considers almost similar parameters as our proposed protocol in [8]. Akkaya's algorithm is good relevant protocol to our protocol and well known routing protocol in wireless sensor networks. We simulate both TECARP and [8], using C++ language in UNIX operating system environment.

Most of hierarchical routing protocols are composed of two main parts. The first part is routing intra clusters and the second part is routing inter clusters. The first part's role is more important. The number of cluster nodes in simulations is considered as to be between 29 and 99. The communication range is determined Based on number of nodes in cluster. We consider almost a 40*40 square for each cluster.

The main goal in TECARP is congestion avoidance. TECARP makes a routing tree by considering the most number of children for its nodes. When the number of children in a routing tree is limited, traffic volumes which enter to the node are limited too. Therefore when an event is occurred, congestion occurrence probability will be decreased. Another parameter which affects TECARP success in congestion management is the node's awareness about their neighbors average queue length. In this situation when nodes want to select the next hop for their data, they consider their neighbors average queue length as a parameter in decision making. The node with lower average queue length rather than other nodes has higher hope to be selected as next hop.

In the first graph, we evaluate the number of lost packets due to congestion for two protocols. In Figure 1(a) the number of packet loss in two protocols for different numbers of events is presented.

As is observable in Figure 1(a) the number of packet loss when network uses TECARP is less than another protocol. TECARP manages congestion; therefore the result shown in Figure 1 is obtained. As discussed before, one of the main influences of congestion occurrence in network is packet loss. Using congestion avoidance mechanism, congestion occurrence is decreased and therefore packet loss is decreased too.

In Figure 1(b) and 2(a) the number of received packets to cluster head is plotted versus the number of events. The number of nodes and events' place are different in two experiences. As shown in Figure 1(b) and 2(a), in the same conditions, number of packets received to cluster head in TECARP is always more than another protocol. TECARP manages congestion; it tries to reduce packet loss in nodes which are located in paths between nodes and the cluster head. Lower packet loss leads to more success in delivering packets to cluster head. Both Figure 1(b) and 2(a) show that TECARP is more successful than another protocol.

As mentioned in former sections, TECARP considers two types of traffic. Network nodes service traffics based on their priority. Both of the protocols try to deliver the best possible services to high priority traffic besides deliver suitable service to low priority traffic. In Figure 2(b) the number of loss packets from each type of traffic for both protocols is plotted versus the number of events. TP1 is high priority traffic and TP2 is low priority traffic.

As observable in Figure 2(b), proposed protocol deliv-

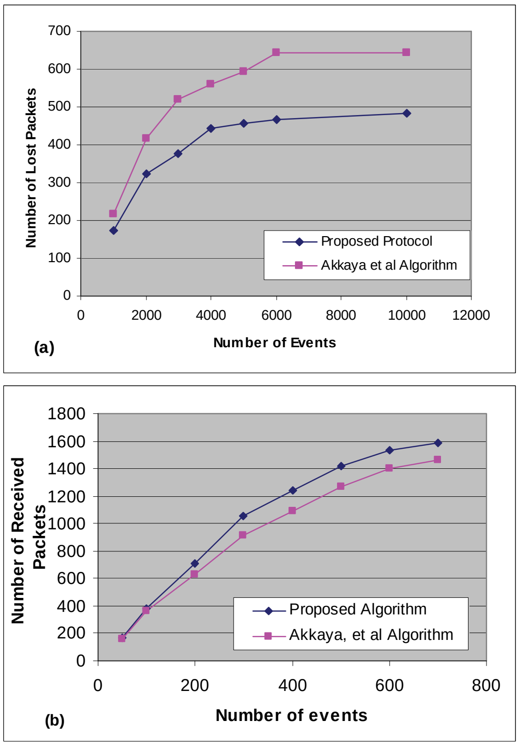


Figure 1. (a) Number of lost packets versus number of events; (b) number of received packets versus number of events.

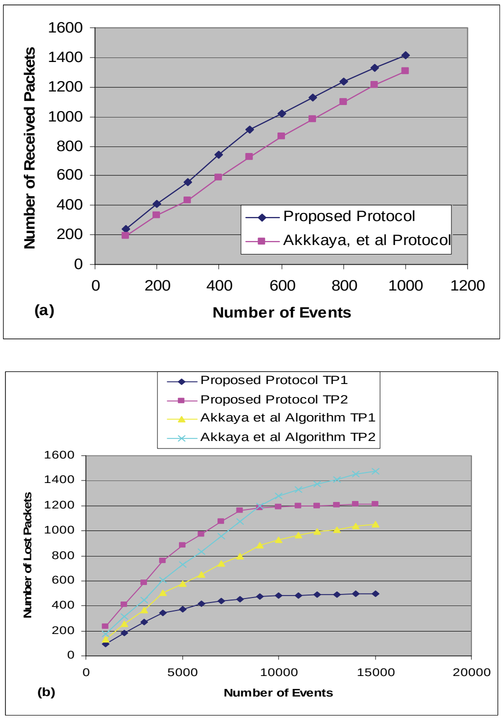


Figure 2. (a) Number of received packets to cluster head versus number of events; (b) number of lost packets versus number of events for different types of traffic.

ers better services to traffics rather than another protocol. Our proposed protocol delivers best possible services to high priority traffics. In forwarding high priority traffics proposed protocol considers only quality of delivered services to packets. But in regard of low priority traffics, in addition to quality of service, Routing protocol tries to provide fairness more efficient and to avoid congestion as much as possible. In other words, in forwarding low priority traffics proposed protocol cares more to its desirable parameters.

In Figure 3(a) performing fairness in two protocols are compared. As mentioned in Section 1, providing fairness is an important factor in wireless sensor networks routing protocols efficiency. Imagine a protocol which sends all of the data from a best cost route, in this situation, the energy of nodes located in the best route are depleted very soon; however other cluster nodes have considerable residual energy. TECARP tries to consume nodes energy as fairly as possible. Using residual energy parameter, TECARP achieves its goal. In Figure 3(a) Deviation parameter which is calculated using formula 2 is plotted versus number of events received to cluster head.

$$Deviation = \left(Node_i Energy - Average\right)^2 \qquad (2)$$

As shown in Figure 3(a), deviation parameter which depends on providing fairness has a better value for TECARP rather than another protocol. If a protocol provides fairness better; deviation parameter is valued less for it. For number of events less than 800, Akkaya protocol provides fairness better than our proposed protocol. Of course it is due to special routing process which is used in proposed protocol.

In Figure 3(b) a snapshot from network is plotted. In this figure, consumed energy of nodes for both of protocols is presented. As observable in Figure 3(b), proposed protocol is more successful than counterpart in providing fairness. Red columns have different heights rather than blue columns; in other words, blue columns are more equivalent in heights in comparison with red columns. The optimal condition occurs, when all of the nodes have the same residual energy.

5. Conclusions

In this paper, we presented a new hierarchical energy efficient routing protocol for sensor networks which considers congestion management based on paper [15]. Routing protocol divides network into many clusters, then using Dijkstra algorithm constructs a routing tree for each cluster. In routing tree, most number of children for cluster nodes is determined. Proposed protocol manages congestion, using routing tree, node's neighbors average queue length and residual energy of nodes as parameters. The effectiveness of the protocol is validated by simulation. Simulation results show that our protocol

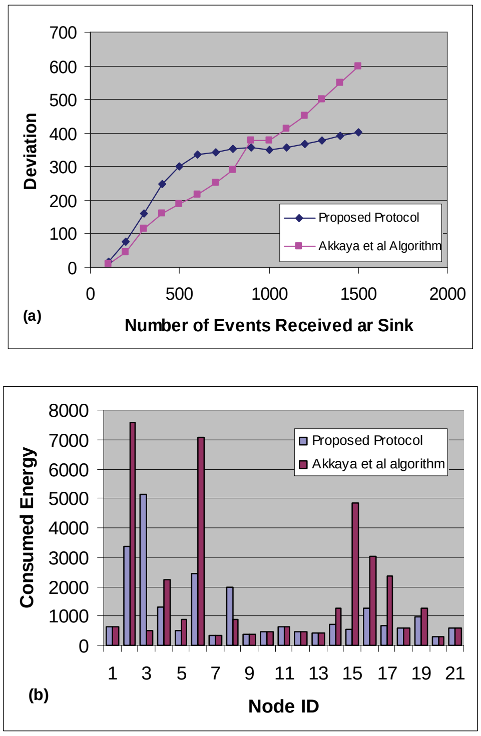


Figure 3. (a) Deviation versus number of events received at cluster head; (b) consumed energy for all cluster nodes separately.

achieved its goals. Proposed protocol considers only intra cluster routing; we are currently extending the protocol to perform routing inter clusters.

6. References

[1] M. Tubaishat and S. Madria, “Sensor networks: an overview,” IEEE Potentials April/May, pp. 20–23, 2003.

[2] I. F. Akyildiz, W. Su, W. Sankarasubramaniam, and E. Cayirci, ”A survey on sensor networks,” IEEE Communication magazine, pp. 102–114, 2002.

[3] H. Hassanein and L. Jing, “Reliable energy aware routing in wireless sensor networks,” Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems, DSSNS’06, pp. 54–64, 24–28 April 2006.

[4] Q. F. Jiang and Manivannan, “Routing protocols for sensor networks,” 1st IEEE Consumer Communications and Networking Conference, pp. 93–98, 2004.

[5] S. G. Chen, and N. Yang, “Congestion avoidance based on lightweight buffer management in sensor networks,” IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 9, pp. 934–946, September 2006.

[6] Z. Eskandari, M. H. Yaghmaee, and A. H. Mohajerzadeh, “Energy efficient spanning tree for data aggregation in wireless sensor networks,” SN’08 Workshop at ICC-CN, 2008.

[7] T. Niwat, T. Yoshito, and S. Kaoru, “Tree-based data dissemination in wireless ssnsor networks,” Proceedings of the IEICE General Conference (Institute of Electronics, Information and Communication Engineers), Vol. 2005, pp. S.41–S.42, 2005.

[8] K. Akkaya and M. Younis, “An energy aware QoS routing protocol for wireless sensor networks,” ICDCS Workshop’03, May 2003.

[9] R. Vidhyapriya and P. T. Vanathi, “Energy aware routing for wireless sensor networks,” Signal Processing, Communications and Networking, ICSCN’07, International Conference on 22–24 February 2007, pp. 545–550, 2007.

[10] Y. H. Wang, C. P. Hsu, Y. C. Lin, C. S. Kuo, and H. Y. Ho, “A routing method by reactive energy decision in wireless sensor networks,” 21st International Conference on Advanced Information Networking and Applications Workshops, AINAW’07, IEEE, 2007.

[11] C. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: congestion detection and avoidance in sensor networks,” In Proceedings of the 1st international Conference on Embedded Networked Sensor Systems, Los Angeles,. SenSys’03. ACM Press, New York, pp. 266–279, 05–07 November, 2003.

[12] M. I. Khan, W. N. Gansterer, and G. Haring, “Congestion avoidance and energy efficient routing protocol for wireless sensor networks with a mobile sink,” Journal of Networks, Vol. 2, No. 6, pp. 42–49, 2007.

[13] R. Kumar, *et al*, “Mitigating performance degradation in congested sensor networks,” IEEE Transactions on Mobile Computing, Vol. 7, No. 6, pp. 682–697, 2008.

[14] A. Abbasi and M. Younis, “A survey on clustering protocols for wireless sensor networks,” Vol. 30, Issues 14–15, pp. 2826–2841, 2007.

[15] A. H. Mohajerzadeh, M. H. Yaghmaee, and Z. Eskandari, “Tree based energy efficient and congestion aware routing protocol for wireless sensor networks,” IEEE ICCS, China, pp. 1707–1711, 2008.

Wireless Sensor Network, **2010, 2, 168-172**
doi:10.4236/wsn.2010.22022 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

An Energy-Efficient Access Control Algorithm with Cross-Layer Optimization in Wireless Sensor Networks

Zhi Chen, Shaoqian Li
National Key Laboratory of Communication, University of Electronic Science and Technology, Chengdu, China
E-mail: *chenzhi@uestc.edu.cn*

Received July 20, 2009; *revised October* 25, 2009; *accepted November* 17, 2009

Abstract

This paper presents a wireless sensor network (WSN) access control algorithm designed to minimize WSN node energy consumption. Based on slotted ALOHA protocol, this algorithm incorporates the power control of physical layer, the transmitting probability of medium access control (MAC) layer, and the automatic repeat request (ARQ) of link layer. In this algorithm, a cross-layer optimization is preformed to minimizing the energy consuming per bit. Through theory deducing, the transmitting probability and transmitting power level is determined, and the relationship between energy consuming per bit and throughput per node is provided. Analytical results show that the cross-layer algorithm results in a significant energy savings relative to layered design subject to the same throughput per node, and the energy saving is extraordinary in the low throughput region.

Keywords: Wireless Sensor Network (WSN), Cross-Layer Design, Energy Efficient, Energy Consumption per Bit, Throughput

1. Introduction

Wireless sensor network (WSN) consist of a large number of small, low data rate and inexpensive node that communicate in order to sense or control a physical phenomenon. Most of the applications proposed for WSN depend on node designs that minimize complexity and energy consumption [1]. This is because WSN node battery replacement will be difficult due to deployment in remote locations or in difficult environment. Even when nodes are accessible, their low cost may make it more efficient to simply replace the entire node rather than just its battery.

Opportunities for minimizing WSN node energy consumption exist at all layers of the protocol state. Many of the proposed WSN routing algorithm account for energy consumption in some fashion [2]. Considerable work has also been performed on minimizing energy consumption at the medium access control (MAC) layer [3,4]. Energy can also be saved at the physical layer by optimizing parameters such as modulation scheme and number of transmit antennas [5,6].

While working with individual protocol layers will help to conserve WSN energy, it has been shown that cross-layer optimization of the entire protocol stack will result in the greatest savings [7]. However, computing this optimal design and implementing it across a WSN is difficult due to the strict complexity constraints imposed on the WSN nodes. As a result, researchers have started to investigate simplified cross-layer energy conservation solutions that are more suited to low complexity WSN devices [8–10].

In [8,9], promising results have been presented for minimizing WSN energy consumption using across-layer power control algorithm that accounts for link layer and physical layer behavior. This approach balances the energy consumed by the physical layer hardware with the energy consumed by frames retransmitted as part of an automatic repeat request (ARQ) error recovery scheme. The basic idea is that lowering transmit power decreases physical layer energy consumption but a transmit power that is too low increases the energy wasted on excess ARQ retransmissions. As a result, an optimal transmit power exists that achieves a compromise between these two effects. One drawback to the work presented in [8,9] is that the physical layer model used in both papers assumes only thermal noise in the radio channel and does not include multiple access interference (MAI). When the SNR is high, the MAI effect on the BER can't be

This work is supported in part by Key Projects in National Science & Technology Program under Grant 2008ZX03005-001

ignored, so the optimized transmit power level presented in [8,9] would not be the actual optimized value.

Based on the work of [8,9], a new power control algorithm that accounts for MAI and MAC layer behavior is proposed in [10]. The contribution of [10] is that the MAI effect on physical BER is taken into account, and an accurate queuing system is used to model for ALOHA MAI. Although how physical layer reliability is affected by the MAI is considered, the parameter designs of MAC layer, such as transmit probability, are not involved in the cross-layer optimization in [10]. Meanwhile, the comparisons of energy consumption alone are not sufficient, because the energy consumption per bit is always relative to the throughput per node. So the trade-off between the energy consumption per bit and the throughput per node should be taken into consider.

The contribution of this paper is to present a new cross-layer access control (CLAC) algorithm that accounts for power control of physical layer, access control of MAC layer, and ARQ behavior of link layer. Using the CLAC algorithm, a significant energy savings relative to layered algorithms can be achieved subject to the same throughput demand.

2. The System Model

A star network topology is assumed where each sensor node transmits its send information to a sink node. Although the star network showed in Figure 1 is very simple, it could represent a complicate mesh network where the routing algorithm forms clusters with all nodes in cluster transmitting to a central node designated as the cluster head.

In the link layer, the ARQ mechanism is adopted. For the simplification of analysis, the overhead of packet header and CRC in front channel, the error and delay in back channel are all ignored.

In the MAC layer, slotted ALOHA is selected as the random access protocol. Each sensor node has the nodes transmit signals in the same frequency band, and the band width is B. Each node transmits information at a fixed rate of R. The signal transmit power and the noise power are denoted by P and P_N respectively, and hence the transmit $\mathrm{SNR} = P/P_N$. We assume a Rayleigh block fading channel for every sensor node in every slot. The fading gain for user i in the time slot t is denoted by $h_i(t)$ and is assumed to have unit variance. The fading is also assumed to be independent across users and slots, so h_i can be used to define the fading gain for sensor node i in any time slot. The sink node employs single user decoding to decode information from sensor nodes. By that we mean the sink node decode one sensor's codeword assuming ever other sensor node's signal as noise. In order not to be constrained with a special modulate scheme, outage probability ε is used to describe the transmitting error. The outage probability of k sensor nodes transmit in a slot is [12].

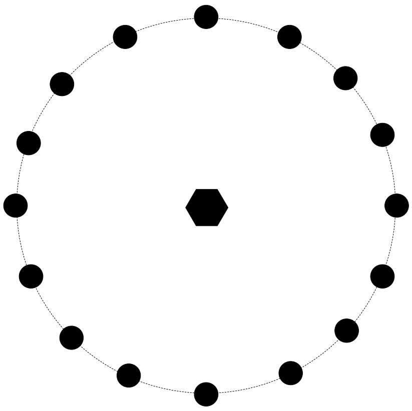


Figure 1. A star topology for WSN.

$$\varepsilon = 1 - \Pr\left\{ R \le B\log\left(1 + \frac{|h_1|^2\,\mathrm{SNR}}{\sum_{i=2}^{k}|h_i|^2\,\mathrm{SNR}+1}\right)\right\} \tag{1}$$

When the fading coefficients are i.i.d. Rayleigh with unit variance, then

$$\varepsilon = 1 - \exp\left(-\frac{\left(2^{R/B}-1\right)}{\mathrm{SNR}}\right)2^{-R(k-1)/B} \tag{2}$$

From (2), it is obviously that the outage probability ε is relative to SNR and k. In this scheme, the throughput is given by

$$\rho(\mathrm{SNR},k) = (1-\varepsilon)R = R\times\exp\left(-\frac{\left(2^{R/B}-1\right)}{\mathrm{SNR}}\right)2^{-(k-1)R/B} \tag{3}$$

The sensor node has small bulk and simple functions, most of the energy is consumed by the signal transmittting, so the circuit energy consumption is ignored in our analysis. The energy consumption per bit involves the energy consumed by the physical layer signal transmitting and the energy consumed by link layer frames retransmitting. The energy consumption per bit with the scheme of SNR and k is given by

$$E(\mathrm{SNR},k) = P/\left[R(1-\varepsilon)\right] = (P/R)\times\exp\left(\frac{2^{R/B}-1}{\mathrm{SNR}}\right)2^{(k-1)R/B} \tag{4}$$

The number k of sensor nodes transmitting in a slot is relate to the transmits probability p, the throughput and energy consumption per bit can be computed by averaging $\rho(\mathrm{SNR},k)$ in (3) and $E(\mathrm{SNR},k)$ in (4) by k [12]. The results are given as follows

$$\begin{aligned}\rho(\mathrm{SNR},p) &= \sum_{k=1}^{N}\binom{N}{k}p^k(1-p)^{N-k}\rho(\mathrm{SNR},k)\\ &= \left(2^{-R/B}p+1-p\right)^{N-1}pNR\exp\left(-\frac{2^{R/B}-1}{\mathrm{SNR}}\right)\end{aligned} \tag{5}$$

$$\begin{aligned}E(\mathrm{SNR},p) &= \sum_{k=1}^{N}\binom{N}{k}p^k(1-p)^{N-k}E(\mathrm{SNR},k)\\ &= \left(2^{R/B}p+1-p\right)^{N}P/R\exp\left(\frac{2^{R/B}-1}{\mathrm{SNR}}\right)2^{-R/B}\end{aligned} \tag{6}$$

3. Cross-Layer Access Control Algorithm

According to (5) and (6), the optimization problem of minimizing energy consumption per bit can be described as follow, subject to the throughput demand ρ, the optimal transmit power level P and transmit probability p should be decided to achieve the minimal energy consumption per bit E. This is a nonlinear programming problem with constraints, the straightforward solution is much complicated, which is not suited to low complexity WSN devices. Therefore, a simplified optimization algorithm should be developed for sensor nodes.

The partial derivative of function $E(\mathrm{SNR},p)$ with respect to argument p is computed as

$$\begin{aligned}\frac{\partial E(\mathrm{SNR},p)}{\partial p} &= \left(2^{R/B}p+1-p\right)^{N}P/R\exp\left(\frac{2^{R/B}-1}{\mathrm{SNR}}\right)2^{-R/B}\\ &= N\left(2^{R/B}p+1-p\right)^{N-1}\left(2^{R/B}-1\right)P/R\exp\left(\frac{2^{R/B}-1}{\mathrm{SNR}}\right)2^{-R/B} >\end{aligned} \tag{7}$$

According to (7), $E(\mathrm{SNR},p)$ is a monotonic increasing function of p. Similarly, the partial derivative of function $E(\mathrm{SNR},p)$ with respect to argument P is computed as

$$\begin{aligned}\frac{\partial E(\mathrm{SNR},p)}{\partial P} &= \left(2^{R/B}p+1-p\right)^{N}P/R\exp\left(\frac{\left(2^{R/B}-1\right)P_N}{P}\right)2^{-R/B}\\ &= \frac{\left(2^{R/B}p+1-p\right)^{N}2^{-R/B}\exp\left(\frac{2^{R/B}-1}{P/P_N}\right)}{R}\left(1-\frac{\left(2^{R/B}-1\right)P_N}{P}\right)\end{aligned} \tag{8}$$

According to (8), there exists an optimal solution of P to achieve the minimal $E(\mathrm{SNR},p)$. By solving the Equation $\frac{\partial E(\mathrm{SNR},p)}{\partial P}=0$, the optimal solution of P is given by

$$\tilde{P} = \left(2^{R/B}-1\right)P_N \quad , \quad \text{or} \quad \mathrm{SNR} = 2^{R/B}-1 \tag{9}$$

It is obvious that $\tilde{P}$ is independent of the value of P.

In the same way, by solving the partial derivative of function $\rho(\mathrm{SNR},p)$ with respect to arguments P and P, it is found that $\rho(\mathrm{SNR},p)$ is a monotonic increasing function of P, and there exists an optimal solution of P to achieve the maximal $\rho(\mathrm{SNR},p)$. The optimal solution of p is given by

$$\tilde{p} = \frac{1}{N\left(1-2^{-R/B}\right)} \tag{10}$$

Note that $\tilde{p}$ is independent of the value of P.

Substituting P and p using relevant optimal value $\tilde{P}$ and $\tilde{p}$ in (5) and (6), we get the cross-layer optimized throughput and energy consumption per bit as follows

$$\tilde{\rho} = \frac{R}{1-2^{-R/B}}e^{-1}\left(1-\frac{1}{N}\right)^{N-1} \tag{11}$$

$$\tilde{E} = \frac{2^{R/B(N-1)}\left(2^{R/B}-1\right)P_N e}{N^N R} \tag{12}$$

The low energy consumption performance of this CLAC algorithm is analyzed by comparing with the traditional layered access control (TLAC) algorithm in which the transmit power and the transmit probability is determinate independently.

In TLAC algorithm, the power control balances the energy consumed by the physical layer and the link layer, therefore, the Equation $\mathrm{SNR}=2^{R/B}-1$ is satisfied still. The difference of TLAC from CLAC is that the transmit probability P is independent of the value of SNR. The throughput and energy consumption per bit in TLAC algorithm are given by

$$\rho' = \left(2^{-R/B}p+1-p\right)^{N-1}pNRe^{-1} \tag{13}$$

$$E' = \frac{\left(2^{R/B}p+1-p\right)^{N}e\left(2^{R/B}-1\right)P_N 2^{-R/B}}{R} \tag{14}$$

4. Comparison of Layered and Cross-Layer Approaches

Because there exists the trade-off between the throughput and energy consumption per bit, the plot of energy consumption per bit versus the throughput can compare the energy saving performance of CLAC and TLAC fairly and clearly.

In order to generate this plot, some assumptions must be made about the parameters of the WSN. It is assumed that the WSN consist of $N = 16$ sensor nodes. A data rate of $R = 20\text{kbps}$, a noise power of $P_N = -110\text{dbmw}$ and a frequency bandwidth of $B = 20\text{kHz}$ are chosen for the physical layer. A transmit probability $p = 1/16$ is used for TLAC. The resulting energy consumption per bit versus the throughput plot is shown in Figure 2. The upper curve denotes the TLAC algorithm and the lower curve denotes the CLAC algorithm.

Figure 2 shows that CLAC consumes less energy per bit than TLAC at the same throughput demand. In the high throughput region, two curves are close to each other. The SNR increase along with the increase of throughput, the MAI has more evident influence on BER than noise. This evident influence of MAI result in that the transmit probability designs can ignore the value of SNR, so the performances of two algorithm are close. On the other hand, CLAC have significant energy savings relative to TLAC in the low throughput region. The SNR decrease along with the decrease of throughput, the effect of noise on BER is obvious, so it is very necessary for the transmit probability variety according to SNR.

Another interesting phenomenon can be found in Figure 2. For CLAC, the energy consumption per bit decreease as the decreasing of throughput, but the energy consumption per bit for TLAC trended to be constant as the decreeasing of throughput. The reason is that the transmit probability for TLAC dose not change along with SNR, the multiple access channel capacity is not used adequately, and the decreasing of SNR and the decreasing of data rate have canceling effect on the energy consumption per bit.

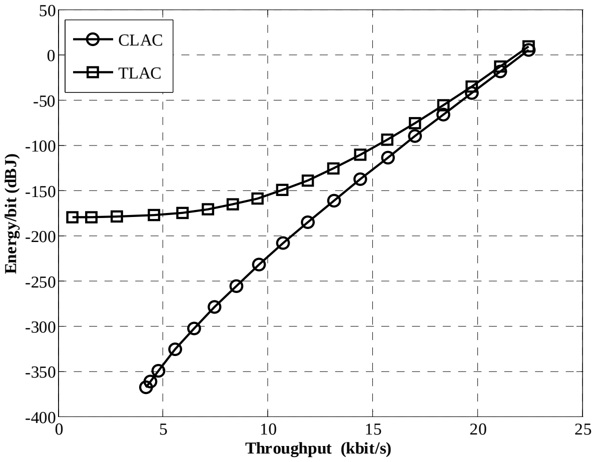


Figure 2. Energy consumption per symbol vs. throughput per node.

It also emphasized that the cross-layer optimizes access control is indispensable for low throughput WSN.

5. Conclusions

This paper presents a WSN access control algorithm. Based on slotted ALOHA protocol, this algorithm incorporates the power control of physical layer, the transmitting probability of MAC layer, and the ARQ of link layer. In this algorithm, a cross-layer optimization is preformed to minimizing the energy consuming per bit. Analytical results show that the cross-layer algorithm results in a significant energy savings relative to layered design subject to the same throughput per node. At the same time, the cross-layer algorithm has low complexity of implementation, and it is suitable to the low energy consumption and low complexity WSN devices.

6. References

[1] K. Romer and F. Mattern, “The design space of wireless sensor networks,” IEEE Wireless Communications, Vol. 11, No. 6, pp. 54–61, 2004.

[2] J. N. Al-Karaki, and A. E. Kamal, “Routing techniques in wireless sensor networks: A survey,” IEEE Wireless Communications, Vol. 11, No. 6, pp. 6–28, 2004.

[3] Y. E. Sagduyu and A. Ephremides, “The problem of medium access control in wireless sensor networks,” IEEE Wireless Communications, Vol. 11, No. 6, pp. 44–53, 2004.

[4] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” IEEE/ACM Transactions Networking, Vol. 12, No. 3, pp. 493–506, 2004.

[5] S. Cui, A. J. Goldsmith, and A. Bahai, “Energy-constrained modulation optimization,” IEEE Transactions Wireless Communications, Vol. 4, No. 5, pp. 2349–2360, 2005.

[6] S. Cui, A. J. Goldsmith, and A. Bahai, “Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks,” IEEE Journal on Selected Areas in Communications, Vol. 22, No. 6, pp. 1089–1098, 2004.

[7] J. Goldsmith and S. B. Wicker, “Design challenges for energy constrained ad hoc wireless networks,” IEEE Wireless Communications, Vol. 9, No. 4, pp. 8–27, 2002.

[8] M. De Sanctis, E. Cianca, and M. Ruggieri, “Energy efficient transmit power control for HDR WPAN,” in Proceeding’06 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC’06, pp. 1–5, 2006.

[9] Howitt and J. Wang, “Energy efficient power control policies for the low rate WPAN,” in Proceeding’04 IEEE Conference on Sensor and Ad Hoc Communications and Networks, SECON’04, pp. 527–536, 2004.

[10] G. M. Geoffrey, A. H. Jennifer, and J. D. Robert, “A sensor network cross-layer power control algorithm that

incorporates multiple-access interference," IEEE Transactions on Wireless Communications, Vol. 7, No. 8, pp. 2877–2883, 2008.

[11] L. Tong, Q. Zhao, and G. Mergen, "Multi packet reception in random access wireless networks: from signal processing to optimal medium access control [J]," IEEE Communications Magazine, Vol. 39, No. 11, pp. 108–12, 2001.

[12] V. Naware and L. Tong, "Cross layer design for multiaccess communication over rayleigh fading channels," IEEE Transactions on Wireless Communications, Vol. 7, No. 3, pp. 1095–103, 2008.

Wireless Sensor Network, 2010, 2, 173-185
doi:10.4236/wsn.2010.22023 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

A Study on Vehicle Detection and Tracking Using Wireless Sensor Networks

G. Padmavathi[1], D. Shanmugapriya[2], M. Kalaivani[1]
[1]*Department of Computer Science, Avinashilingam University for Women, Coimbatore, Tamil Nadu, India*
[2]*Department of Information Technology, Avinashilingam University for Women, Coimbatore, Tamil Nadu, India*
E-mail: {*ganapathi.padmavathi, kalaivanim*}*@gmail.com, ds_priyaa@rediffmail.com*

Received October 26, 2009; *revised November* 11, 2009; *accepted December* 7, 2009

Abstract

Wireless Sensor network (WSN) is an emerging technology and has great potential to be employed in critical situations. The development of wireless sensor networks was originally motivated by military applications like battlefield surveillance. However, Wireless Sensor Networks are also used in many areas such as Industrial, Civilian, Health, Habitat Monitoring, Environmental, Military, Home and Office application areas. Detection and tracking of targets (eg. animal, vehicle) as it moves through a sensor network has become an increasingly important application for sensor networks. The key advantage of WSN is that the network can be deployed on the fly and can operate unattended, without the need for any pre-existing infrastructure and with little maintenance. The system will estimate and track the target based on the spatial differences of the target signal strength detected by the sensors at different locations. Magnetic and acoustic sensors and the signals captured by these sensors are of present interest in the study. The system is made up of three components for detecting and tracking the moving objects. The first component consists of inexpensive off-the shelf wireless sensor devices, such as MicaZ motes, capable of measuring acoustic and magnetic signals generated by vehicles. The second component is responsible for the data aggregation. The third component of the system is responsible for data fusion algorithms. This paper inspects the sensors available in the market and its strengths and weakness and also some of the vehicle detection and tracking algorithms and their classification. This work focuses the overview of each algorithm for detection and tracking and compares them based on evaluation parameters.

Keywords: Wireless Sensor Networks, Acoustic and Magnetic Sensors, Acoustic and Magnetic Signals, Detection and Tracking Algorithms

1. Introduction

The Wireless Sensor Networks comprise of relatively inexpensive sensor nodes capable of collecting, processing, storing and transferring information from one node to another. These nodes are able to autonomously form a network through which sensor readings can be propagated. Since the sensor nodes have some intelligence, data can be processed as it flows through the network. Sensing devices will be able to monitor a wide variety of ambient conditions: temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects and so on. These devices will also be equipped with significant processing, memory and wireless communication capabilities. Emerging low-level and low-power wireless communication protocols can be used to networks the sensors. This capability will add a new dimension to the capabilities of sensors. Sensors will be able to coordinate among themselves on a higher-level sensing task. The sensors can be deployed in any facility or area, which has to be sensed in three main types. It can either be 1) triangular sensor deployment, 2) square sensor deployment and 3) irregular sensor deployment [1]. These deployments are depicted in Figure 1.

Networking inexpensive sensors can revolutionize information gathering in a variety of situations.

A sensor node usually consists of four sub-systems:

- A computing subsystem: In a sensor node, the microprocessor (microcontroller unit, MCU) is responsible for functions such as control of sensors and execution of communication protocols.
- A communication subsystem: This comprises of

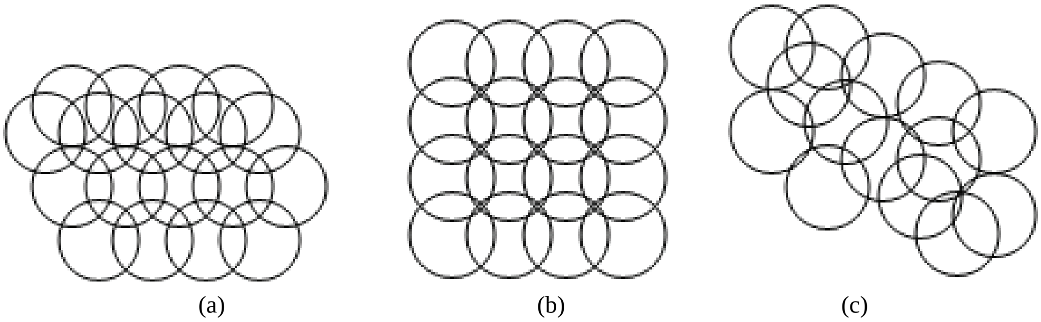


Figure 1. (a) Triangular; (b) Square; (c) Irregular deployments.

short range radios used to communicate with neighbouring nodes and the outside world. These devices operate under the Transmit, Receive, Idle and Sleep modes having various levels of energy consumption.

- A sensing subsystem: Low power components can help to significantly reduce power consumption. Since this subsystem (sensors and actuators) is responsible for the sharing of information between the sensor network and the outside world.
- A power supply subsystem: It consists of a battery which supplies power to the node.

1.1. Characteristics of WSN

Some of the unique characteristics of a WSN include:

- Limited power they can harvest or store.
- Ability to withstand harsh environmental conditions
- Ability to cope with node failures
- Mobility of nodes
- Dynamic network topology
- Communication failures
- Heterogeneity of nodes Large scale of deployment
- Unattended operation

1.2. Applications

The various sensor network applications include,

1) *Military Applications*

- Monitoring Friendly Forces, Equipments and Ammunition
- Battlefield Surveillance
- Targeting
- Battle Damage Assessment
- Nuclear, Biological and Chemical Attack Detection

2) *Environmental Applications*

- Forest Fire Detection
- Flood Detection
- Monitoring through Internet
- Monitoring Biodiversity

3) *Habitat Monitoring Applications*

- Habitat Monitoring
- Great Duck Island System

4) *Health Applications*

- Tele-monitoring of Physical Data
- Tracking and Monitoring Doctors and Patients in a Hospital
- Drug Administration in Hospitals

5) *Home and office Applications*

- Smart Homes
- Managing Inventory Control
- Home automation
- Environmental control in office buildings

6) *Other Applications*

- Monitoring Nuclear Reactor.
- Target Tracking
- Suspicious Individual detection: Interactive museums
- Fire Fighters Problem (First Responders Problem)

The major contribution of this paper includes classification of sensors available and vehicle detection and tracking algorithms using Wireless Sensor Networks. The paper is organized as follows: Section 2 gives the classification of sensors used for vehicle detection and mainly concentrates on the acoustic and magnetic sensors. Section 3 gives the gist about the acoustic and magnetic signals acquired from the sensors. Section 4 discusses the various algorithms used in vehicle detection and tracking, analysis about these algorithms is given in Section 5 and summary of these algorithms given in Section 6 followed by the conclusion.

2. Classification of Sensors Used in Vehicle Detection

There are a wide variety of sensors available in market today [2]. Table 1 shows the variety of current sensor technologies and compares the strengths and weaknesses with respect to installation, parameters measured, and performance in bad weather, variable lighting, and changeable traffic flow. Many over-roadway sensors are compact and mounted above or the side of the roadway, making installation and maintenance relatively easy. Some sensor installation and maintenance applications may require the closing of the roadway to normal traffic to ensure the safety of the installer and motorist. All the sensors listed here operate under day and night conditions. Sensors are broadly classified as Intrusive and Non-Intrusive sensors.

Table 1. Strengths and weaknesses of commercially available sensor technologies.

Technology	Strengths	Weaknesses
Inductive loop	Flexible design to satisfy large variety of applications. Mature, well understood technology. Large experience base. Provides basic traffic parameters (e.g., volume, presence, occupancy, speed, headway, and gap). Insensitive to inclement weather such as rain, fog, and snow. Provides best accuracy for count data as compared with other commonly used techniques. Common standard for obtaining accurate occupancy measurements. High frequency excitation models provide classification data.	Installation requires pavement cut. Improper installation decreases pavement life. Installation and maintenance require lane closure. Wire loops subject to stresses of traffic and temperature. Multiple loops usually required to monitor a location. Detection accuracy may decrease when design requires detection of a large variety of vehicle classes.
Magnetometer (two-axis fluxgate magnetometer)	Less susceptible than loops to stresses of traffic. Insensitive to inclement weather such as snow, rain, and fog. Some models transmit data over wireless radio frequency (RF) link.	Installation requires pavement cut. Improper installation decreases pavement life. Installation and maintenance require lane closure. Models with small detection zones require multiple units for full lane detection.
Magnetic (induction or search coil magnetometer)	Can be used where loops are not feasible (e.g., bridge decks). Some models are installed under roadway without need for pavement cuts. However, boring under roadway is required. Insensitive to inclement weather such as snow, rain, and fog. Less susceptible than loops to stresses of traffic.	Installation requires pavement cut or boring under roadway. Cannot detect stopped vehicles unless special sensor layouts and signal processing software are used.
Microwave radar	Typically insensitive to inclement weather at the relatively short ranges encountered in traffic management applications. Direct measurement of speed. Multiple lane operation available.	Continuous wave (CW) Doppler sensors cannot detect stopped vehicles
Active infrared (laser radar)	Transmits multiple beams for accurate measurement of vehicle position, speed, and class. Multiple lane operation available.	Operation may be affected by fog when visibility is less than ≈ 20 feet (ft) (6 m) or blowing snow is present. Installation and maintenance, including periodic lens cleaning, require lane closure
Passive infrared	Multizone passive sensors measure speed.	Passive sensor may have reduced vehicle sensitivity in heavy rain, snow and dense fog. Some models not recommended for presence detection.
Ultrasonic	Multiple lane operation available Capable of over height vehicle detection. Large Japanese experience base.	Environmental conditions such as temperature change and extreme air turbulence can affect performance. Temperature compensation is built into some models. Large pulse repetition periods may degrade occupancy measurement on freeways with vehicles travelling at moderate to high speeds.
Acoustic	Passive detection. Insensitive to precipitation. Multiple lane operation available in some models.	Cold temperatures may affect vehicle count accuracy. Specific models are not recommended with slow-moving vehicles in stop-and-go traffic.
Video image processor	Monitors multiple lanes and multiple detection zones/lane. Easy to add and modify detection zones. Rich array of data available. Provides wide-area detection when information gathered at one camera location can be linked to another.	Installation and maintenance, including periodic lens cleaning, require lane closure when camera is mounted over roadway (lane closure may not be required when camera is mounted at side of roadway). Performance affected by inclement weather such as fog, rain, and snow; vehicle shadows; vehicle projection into adjacent lanes; day-to-night transition; vehicle/road contrast; and water, salt grime, icicles, and cobwebs on camera lens. Reliable night-time signal actuation requires street lighting. Requires 30- to 50-ft (9- to 15-m) camera mounting height (in a side-mounting configuration) for optimum presence detection and speed measurement. Some models are susceptible to camera motion caused by strong winds or vibration of camera mounting structure. Generally cost effective when many detection zones within the camera field of view or specialized data are required.

2.1. Intrusive Sensors

Intrusive sensors are those that need to be installed under the pavement, in saw-cuts or holes on the roads. Popular intrusive sensors include inductive loops, magnetometers, micro loop probes, pneumatic road tubes, piezoelectric cables and other weigh-in-motion sensors. The main advantage of these sensors is their high accuracy for vehicle detection while the drawbacks include the disruption of traffic for installation and repair, resulting in high installation and maintenance cost.

2.2. Non-Intrusive Sensors

To overcome the disadvantage of intrusive sensors, non-intrusive sensors are developed eg, above ground vehicle detection sensors. Above ground sensors can be mounted above the lane of traffic or on the side of a roadway where they can view multiple lanes of the traffic at angles perpendicular to or at a slanting angle to the flow direction. Technologies used in aboveground sensors include video image processing (VIP), microwave radar, laser radar, passive infrared, ultrasonic, passive acoustic array, and combinations of these sensor technologies. However, these non-intrusive sensors tend to be large size and power hunger.

2.3. Smart-Dust Sensor Node-Hardware Platform

Smart-Dust is one of the potential sensor nodes which can be used in the future vehicle detection system. In Smart-Dust sensor node, essential components for vehicle detection (processor, memory, sensor and radio) could be integrated together as small as a quarter through MEMS technology. Together with its low power design [3], the Smart-Dust sensor node is suitable for implementing the vehicle detection sensor networks. Figure 2 shows the different generations of Smart Dust sensor nodes (Motes). The left picture is a 1st generation smart dust sensor node "Rene Mote". From left to right, the right picture shows the "MICA Mote" (2nd generation), "MICA2 Mote" (3rd generation) and "MICA2-Dot Mote (3rd generation)". Smart-Dust sensor node is designed by EECS department in UC Berkeley and Intel [4] using modular component approach and it consists of two major components: mother board and sensor board. Thus, different sensor boards could be attached to the same mother board for different applications. Thus, Smart-Dust sensor node could potentially be used in a wide range of applications such as vehicle detection, enemy monitoring in the battlefield, temperature measurement in a building, environmental monitoring etc.

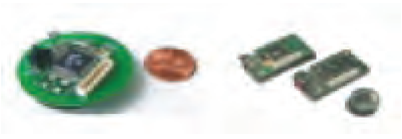


Figure 2. Dust family.

The basic components of MICA mote shown in Figure 3 belongs to the Smart-Dust family. The components are listed in Table 2. The mother board consists of an Atmel 90LS8535 processor, 512KB SRAM, 8KB Flash RAM and a RF transceiver for wireless communication. The Sensor board consists of a 10-bit analog to digital converter, a Magnetometer (Honeywell HMC1002), a temperature sensor, a photo camera and an accelerometer sensor.

For vehicle detection system, the sensors used are the magnetometer and acoustic sensors. Next, the basic operating principles of magnetometer and acoustic sensors are will be reviewed.

2.3.1. Acoustic Sensors

The acoustic sensor in the Smart Dust sensor node is a condenser type microphone. The schematic for a typical condenser acoustic sensor is shown in Figure 4. It includes a stretched metal diaphragm that forms one plate of a capacitor. A metal disk placed close to the diaphragm acts as a backplate. A stable DC voltage is applied to the plates through a high resistance to keep electrical charges on the plates. When a sound field excites the diaphragm, the capacitances between the two plates vary according to the variation in the sound pressure. The change in the capacitance generates an AC output proportional to the sound pressure, which shows the ultra low-frequency pressure variation. A high-frequency voltage (carrier) is applied across the plates and the acoustic sensor output signal is the modulated carrier. The photo in the right of Figure 4 shows the Panasonic

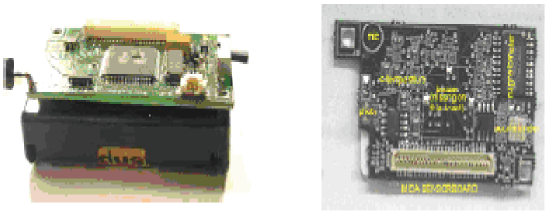


Figure 3. MICA mote.

Table 2. Components of smart dust mote.

Mother Board	Sensor Board
Atmel 90LS8535 processor (clocked at 4 MHz)	10-bit analog to digital converter
RF Monolithics transceiver (916.50 MHz)	Magnetometer(Honeywell HMC1002)
512KB SRAM, 8KB Flash RAM	Microphone (Panasonic WM-62A) Temperature Sensor Photo Camera Accelerometer Sensor

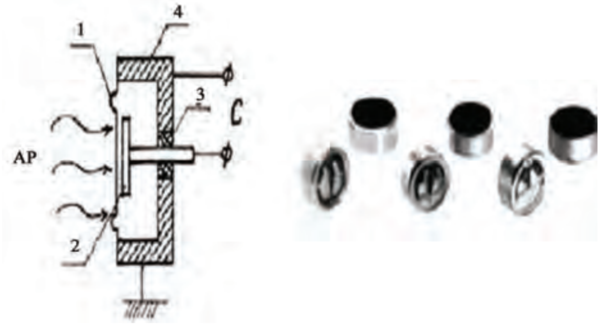


Figure 4. Condenser microphone. AP-the acoustic pressure; C-the variable capacitance; 1-the metal diaphragm; 2-the metal disk; 3-the insulator; 4-the case.

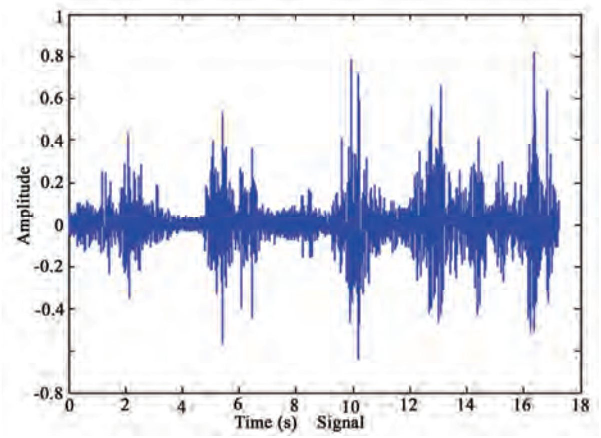


Figure 5. Waveforms of acoustic signals emitted from truck.

WM-62A condenser microphones used in Smart Dust Motes. Figure 5 shows a typical vehicle acoustic signal waveforms.

2.3.2. Magnetic Sensors

Magnetic sensors differ from most other detectors in that they do not directly measure the physical property of interest. Devices that monitor properties such as temperature, pressure, strain, or flow provides an output that directly reports the desired parameter (Figure 6). Magnetic sensors, on the other hand, detect changes, or disturbances, in magnetic fields that have been created or modified, and from them derive information on properties such as direction, presence, rotation, angle, or electrical currents. The output signal of these sensors requires some signal processing for translation into the desired parameter. Although magnetic detectors are somewhat more difficult to use, they do provide accurate and reliable data without physical contact.

The Honeywell HMC1002 magnetometer on the MICA sensor board is a magnetoresistive sensor. Honeywell magnetic sensors and magnetometers offer complete magnetic field sensing solutions that are highly accurate, and allow for easy integration for virtually any application. Figure 7 shows the honeywell magnetic sensors.

The Anisotropic Magneto Resistive (AMR) sensor is one type that has a wide Earth's field sensing range and can sense both the strength and direction of the Earth field. The AMR sensor is made of a nickel-iron (Permalloy) thin film deposited on a silicon wafer and patterned as a resistive strip. The strip resistance changes about 2%-3% when a magnetic field is applied. Typically, four of these resistive strips are connected in a wheatstone bridge configuration so that both magnitude and direction of a field along a single axis can be measured. The key benefit of AMR sensors is that they can be bulk manufactures on silicon wafers mounted in commercial integrated circuit packages. Figure 8 shows a typical vehicle magnetic signal waveforms.

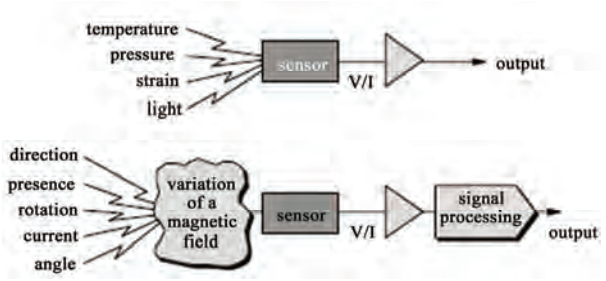


Figure 6. Magnetic sensor parameters.

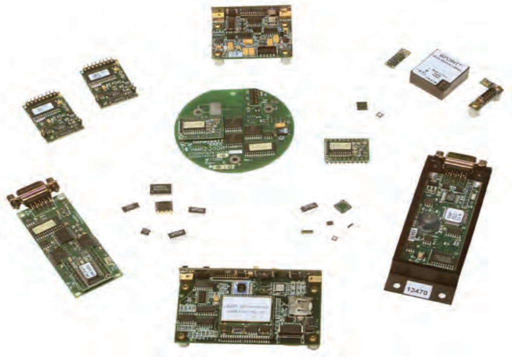


Figure 7. Magnetic sensors.

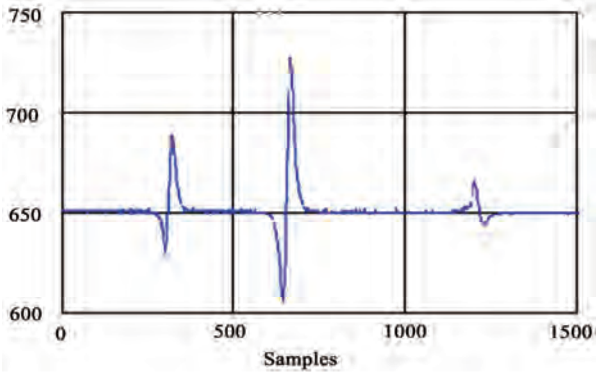


Figure 8. Waveforms of magnetic signals.

Magnetic Field

A magnetic field is a vector field that surrounds magnets and electric currents, and is detected by the force it exerts on moving electric charges and on magnetic materials. When placed in a magnetic field, magnetic dipoles tend to align their axes parallel to the magnetic field. Magnetic fields also have their own energy with an energy density proportional to the square of the field intensity.

Magnetic Data

Magnetic data is the term used for data that is acquired from magnetic (as opposed to optical) motion capture systems. A central magnet is used to create a field in which sensors can determine their position and orientation as they move about in the field. Raw magnetic data has no hierarchy information; the sensors do not know where they are relative to the other sensors, and know their position in the magnetic field. The file formats for magnetic motion capture data reflect this. In this respect, they are somewhat similar to the BVA file format.

The problem associated with magnetometer vehicle detection is similar to the acoustic sensors but the magnetic signals are much cleaner than acoustic signals. The magnetometers available today can sense magnetic fields within the earth's field-below 1 gauss. They can be used for detecting the vehicles, which are ferrous objects that disturb the earth's field. The earth's field provides a uniform magnetic field over wide area in the scale of kilometres and even a ferrous object can create a local disturbance in this field. This local field disturbance can be sensed by the magnetometers for vehicle detection. Figure 9 shows the disturbance of earth's magnetic field by a car. After presenting the basic principles of sensors, the characteristics of the measured acoustic and magnetic signals will be studied and algorithms are proposed for reliable low cost vehicle detection.

3. Signals for Vehicle Detection and Tracking

The signals taken into consideration for study in this proposal are acoustic and magnetic signals.

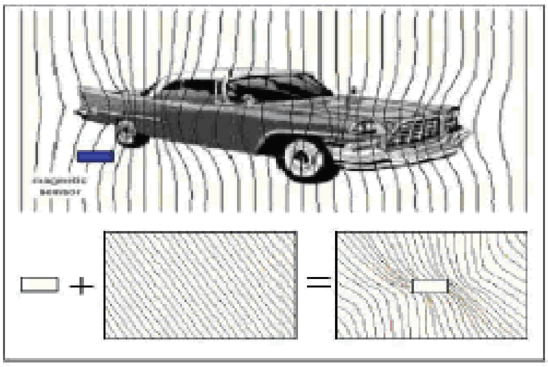


Figure 9. Earth's magnetic field disturbed by a car.

3.1. Acoustic Signals

The acoustic signature is made up of a number of individual elements. These include:

- Machinery noise: noise generated by a ships engines, propeller shafts, fuel pumps, air conditioning systems, etc.
- Cavitation noise: noise generated by the creation of gas bubbles by the turning of a ship's propellers.
- Hydrodynamic noise: noise generated by the movement of water displaced by the hull of a moving vessel.

These emissions depend on a hull's dimensions, the installed machinery and ship's displacement. Therefore, different ship classes will have different combinations of acoustic signals that together form a unique signature.

Sonar operating in passive mode can detect acoustic signals radiated by invisible submarines and these signals are used to target attacks.

3.2. Magnetic Signals

Magnetic Signals are changes that happen in the magnetic field caused by the movement of ferrous objects.

4. Algorithms for Vehicle Detection and Tracking

There are a wide variety of algorithms generally available for detection and Tracking of target. Figure 10 shows the classification of detection and tracking algorithms.

4.1. Target Detection Background

The basic idea behind a target detection algorithm is that sensor nodes are deployed within an area and the nodes determine whether or not a target has entered or is presently within the area. Nodes usually determine if a target is present by detecting a change in some sort of signal, whether it is acoustic, light, temperature or some other type [5]. Since the strength of this signal will scatter throughout the sensor network the nodes need to collaboratively work together in order to determine if a target has been detected or not. Several of these collaborative detection algorithms are discussed in the following section:

4.1.1. Basic Fusion

This algorithm is presented first since it has an extremely simple design and implementation and serves as an ideal algorithm to use as the foundation for explaining other more complex detection algorithms. This algorithm collects the captured data from each of the nodes in a network. Then, to remove any invalid data provided by faulty nodes, the largest and smallest data values are dropped. The remaining data is then averaged together. If this average exceeds a predetermined threshold, then the system is notified that a target has been detected. This algorithm has two versions in which the data provided by each node differs. The data is either the original data collected by the node or it is a detection decision already made by the individual nodes using their original data [5].

4.1.2. Hybrid

This algorithm is similar to the basic fusion algorithm. In that a detection decision is based on data collected by numerous nodes throughout the network. However, this algorithm differs in that either a yes/no target present bit decision is sent or the collected data is sent. In this algorithm, each node is given two threshold values for a previously determined detection signal type, such as sound. One of these threshold values determines that if the data collected by the node exceeds this value then a target has been detected. Correspondingly, the other threshold is a minimum in that if the data is below, then no target is present. If either of these two thresholds is met then a bit message indicating the presence or lack of a target is transmitted. However, if neither of these thresholds is exceeded then the raw data collected by the node is passed on to the base station where it is used with the data of other nodes to determine if a target is present or not [18].

4.1.3. Predicate Clustering

In this algorithm nodes are divided into groups called clusters in which each group has a cluster head that all sub nodes of the cluster report their data to. Typically, one cluster is formed around a single target with multiple clusters being in the network if multiple targets are present. Cluster heads are chosen by all nodes within the area of a detected target “electing” to become the head by broadcasting their intent to become a cluster head. The node with the peak sensor reading is chosen as the cluster head [6]. Once cluster heads are chosen, the remaining nodes choose group to join by running a decision predicate on their captured sensor data and the data of their neighbours. One particularly interesting feature of this algorithm is that a cluster head can be responsible for multiple targets. When it is detected, multiple targets are going to enter the same cluster, both clusters provide the target information to the “merged” cluster head that is now responsible for tracking both targets.

4.1.4. Grids

In this approach the sensor network is broken up into “virtual grids” where each grid contains points. For each grid point, it is determined which nodes’ sensing area covers the grid point and then these nodes create a schedule so that they all collaboratively cover this grid point at alternating times. So at any point of time‘t’ at least one node’s sensing area covers grid point ‘x’. Since each node covers multiple grid points, each node will contain several sensing schedules to keep in sync so that at any point of time all grid points are covered by at least one node. In order to do this, all nodes first determine the schedules for each grid point, then schedule all their individual point schedules together so that each node is not constantly waking and sleeping instead sleeps a little and then wakes for a while. Once each node determines its own individual schedule, it must be correlated with those of its neighbours so that as few as possible nodes are awake at any point of time but all grid points are covered at all times. Once the schedules are completed for each node, the nodes start to follow their schedule, going through states of being asleep and awake in order to detect a new target [7].

4.1.5. Wave

The Wave protocol has the same goal in mind as that of the Grid algorithm, detecting targets while conserving energy by not having all the nodes on all the time. The basic idea of the wave algorithm is to wake up a series of nodes at certain times so that only nodes that are awake at one point of time are in one concentrated area. Then, over time, this section moves over the network by putting nodes to sleep and waking new ones. This gives the appearance of an awake section of nodes flowing over the entire network like a wave [8].

This algorithm proposes three types of waves:

- Two straight lines of nodes are awake at the same time, with the lines being parallel to each other. The two lines start on opposite sides of the network from each other and gradually work towards each other to meet in the centre of the network.
- A wide line of nodes are awake at the same time and this line moves across the network from one edge to the opposite edge.
- Waves of nodes are awake and the wave traverses from one edge of the network to the other. This approach differs from the wide line in that the boundary shape is curvy or wave like instead of being a straight line boundary. The wave is created by an awake sensor waking all nodes within its communication range on the side of it that the wave needs transition too. This approach also differs from the previous two in that it uses radio communication to wake up the next nodes to start sensing instead of time syncing the movement of wake/sleep nodes across the network.

Like the other algorithms, when a target is detected, the node(s) pass the information on for use.

4.1.5.1. Patrolling

Patrolling algorithms are similar to the wave algorithms in that only a series of nodes are awake at one point of time. This algorithm just chooses which nodes to wake differently. This can be done on an on-demand basis or a coverage-orientated basis. The difference between the two is that on-demand is some predetermined path that is currently interesting whereas coverage-orientated just repeatedly watches an area [9].

The algorithm for both approaches is the same. First, a patrol is defined as a set of information containing the patrol speed, duration, iteration period and path. Once this information is collected, a Patrol Host node is selected, which is usually the first node on the patrol path. The duty of the Patrol Host is to periodically transmit the patrol data along with the current patrol time. The idea is that over the time the patrol virtually traverses down the specified path. As the patrol advances, nodes that are closest to the patrol path wake up and collect information. Nodes determine if they are near the patrol path by using the patrol information that the Patrol Hosts broadcast. Additionally, as the patrol moves, new Patrol Hosts must be elected. As the patrol virtually moves out of the current patrol host's range, a new patrol host is selected on the path which then takes over the patrol host duties and the previous patrol host can now sleep. This process of waking nodes and handing off patrol host duties continues until the end of the path is reached. Once the path end is reached, the patrol is reversed in order to traverse the path back to the original starting point. This process continues according to the duration and iteration parameters are set in the original patrol parameters [9].

4.1.6. Mobile Nodes

Until this point, all the target detection algorithms discussed so far have assumed that the sensor nodes remain stationary. This section looks at a few target detection algorithms in which the nodes are mobile within the network. Even though these algorithms allow nodes to be mobile, their goal remains the same as the previously mentioned target detection algorithms. These algorithms are still trying to detect targets in a timely and efficient manner while trying to cover as much of the network as possible using as little energy as possible. The theory behind allowing nodes to be mobile is that they may now detect a target that would have otherwise gone undetected in a stationary node network [10].

4.1.6.1. Past Detections

Like other algorithms, the network in this algorithm is divided into cell areas. Each cell is unique in that it maintains the current state and past history of targets that have been in the cell. This includes the location in which the target was first detected within in the cell. Using this data, the system calculates the areas of the cell that are most likely to contain a target based on the locations of previously detected targets [11].

Once these locations are determined, each node is assigned an area to watch and a priority. These priorities are used in the "coordination mechanism" part of the algorithm. The coordination part of the algorithm is responsible for altering the path nodes so that their observation paths overlap as little as possible. For example, the node that watches the area where targets are the most likely to occur is given the highest priority of the system and its path is not altered. A second node, whose priority is slightly less than the first node, whose path also slightly intersects that of the first node has its path altered so that the two paths will not cross. This path altering continues throughout the system from highest priority down to lowest priority until all node paths are altered to prevent too much overlapping of observation areas [11].

4.1.6.2. Collaborative Coverage

This algorithm is little different in that the paths the nodes traverse are not coordinated in any way. Each node is allowed to create and follow its own arbitrary path. While the node is travelling on this path at different locations and times, it collects data and saves. The sensed data are saved with time and location details [12]. While the nodes move across their paths they share their collected data with those nodes they come into radio communication with. Data is usually shared on a request basis. Each node is provided a coverage threshold and confidence threshold. Actual target detection in this system is determined by examining the data taken at several points that are close to each other. If the combined data collected by this group of close points exceeds some predetermined threshold then it is determined that a target is detected at this area. Note that the value of this detection threshold will determine how sensitive/ insensitive the system is at detecting targets.

4.1.6.3. Bayesian Estimation Track Before Detect (TBD) Estimator

Integrated tracking and detection, based on unthresholded measurements, also referred to as Track Before Detect (TBD) is a hard nonlinear and non-Gaussian dynamical estimation and detection problem. However, it is a technique that enables the user to track and detect targets that would be extremely hard to track and detect, if possible at all with ''classical'' methods. TBD enables one to be better able to detect and track weak, stealthy or dim targets in noise and clutter. The particle filters have shown to be very useful in the implementation of TBD algorithms.

In TBD, the detection problem is done using the track output over multiple scans. The detection decision will be made at the end of the processing chain, i.e., when all information has been used and integrated over time [15]. Although it is called track before detect, the tracking and detection processes occur simultaneously. In this way, the energy of a (weak) target is integrated and correlated over time and position. The concept will lead to a better performance when detecting and tracking weak targets. TBD also implicitly solves the problem with data association.

4.2. Target Tracking Background

Target tracking algorithms usually focus on the aspect of the sensor nodes' interaction with a target after the target has already been detected within the area the sensor nodes cover. Once the object has been detected, the nodes collect information and then use one of many different types of algorithms to calculate the current location of the object relative to the sensor nodes' locations. From here, it is the goal of the sensor network to track the object as it moves through the network. This may or may not involve predicting the next location of the object when it moves. In order to forewarn those nodes it will be heading towards to prepare to capture data [14]. Several of these tracking algorithms are discussed in the following section:

4.2.1. Simple Triangulation

This algorithm is an extremely simple design and implementation and serves as an ideal algorithm to use as the foundation for explaining other more complex tracking algorithms. The whole goal of this algorithm [14] is to provide a simple algorithm that uses simple computation in order to calculate an object's current location and predict where it is headed and notify those nodes near the predicted next location of the object.

This algorithm first assumes that all nodes in the network are localized to a common reference point and can detect and estimate the distance to a target using signal strength [14]. When a node detects an object within its range it broadcasts a *TargetDetected* message. This message contains the location of the sensor node and the distance to the target. All nodes that hear this message store its data in their local memory. When a node that has detected the target hears two other *TargetDetected* messages from two other nodes it performs triangulation on the three coordinates to calculate the location of the target. (Note that this means that more than one node may perform this calculation for the same target at the same time.) This node then continues to project the trajectory of the target. When the estimated target trajectory has been calculated, all nodes that are within some distance 'd' perpendicular to the target's trajectory are sent a Warning message to alert them that the target is headed towards them. These newly awoken nodes then track the object as it enters their area and repeat the *TargetDetected* and *Warning message* sending process.

4.2.2. Clusters

The cluster target tracking algorithm has been widely discussed through many research papers. In this section, the basic idea of target tracking using clusters is discussed, followed by an overview of the variances in each of the different cluster algorithm research papers.

4.2.2.1. Basic Cluster Algorithm

The basic algorithm for tracking an object using clusters is as follows [20]:

- Some (or all) of the nodes in a cluster detect the object and report their data to a cluster head. Note that each cluster has only one cluster head.
- The cluster head node uses all the target detection information from the sensor nodes to estimate the target's location.
- The cluster head uses the calculated target location and past locations of the target to predict the next location of the target.
- Those sensors around the predicted location are woken up to form a new cluster (if not already in one) to detect the target.
- When the target is detected in this new cluster, the previous cluster's nodes are all put into a sleep state. This new cluster then continues the cluster tracking algorithm.

4.2.2.2. Hierarchical Supernodes

This algorithm deviates from the basic algorithm in that the cluster heads (called supernodes) have a higher communication range and more computational power [21]. These "supernodes" are distributed throughout the network and the otherwise normal nodes are assigned to supernodes. Clusters are not dynamically generated in this algorithm. Interestingly enough, supernodes do share target location information among each other, whereas regular sensor nodes do not.

4.2.2.3. Dynamic Clustering

Like the supernode algorithm, this algorithm also assumes that cluster head nodes have more power than normal sensor nodes. However, sensor nodes are not assigned to clusters in this algorithm. Instead, they are invited to join a cluster by the cluster head. The cluster head does this by broadcasting a join message that includes the time and signature of the target the cluster head detected [19]. Those sensor nodes that have stored data that matches the data in the broadcast message respond to the broadcast by sending their captured data to the cluster head node. Interestingly, the cluster head only waits for a certain number of replies and when the required numbers of replies are received, the cluster head calculates the target's location. Unlike the supernode algorithm, this algorithm has one active cluster head at a time. In other words, cluster heads do not work together [22].

4.2.2.4. Cell Collaboration

Like predicate clustering all nodes in this algorithm are of the same type. Nodes are formed into clusters (called cells) in which all nodes within the cells collaboratively decide when a target has entered their cell and if they should track it [22]. It is interesting to note that cells can have different sizes; the size of cells is determined by the observed velocity of the target. So cell size will increase for faster moving targets.

4.2.2.5. Probabilistic Localization

This algorithm deviates even further from the previously discussed algorithms. Cluster heads are special high powered nodes that know the location of every node within their cluster [23]. This algorithm takes advantage of the fact that the cluster head knows these locations. When sensor nodes detect a target, they send a very small notification message to their cluster head and store the target location data, time and other relevant data in their local memory. Upon receiving several notifications, the cluster head performs a probabilistic localization algorithm to determine which sensor nodes to query saved data from. In other words, the cluster head runs an algorithm that helps to predict which sensor nodes are closest to the target. Therefore, the cluster head has the best data to calculate an accurate estimation of the target location. When the cluster head determines which nodes to query, they are asked for the data they saved in their local memory and the cluster head uses this information to calculate the location of the target.

4.2.2.6. Distributes Predictive Tracking

This algorithm is very similar to the predicate localization algorithm in that cluster head knows the ID, location and energy level of each of the nodes within their cluster. However, not all nodes belong to a cluster. Nodes on the border of the network are not in a cluster and those are on sense at all times. Similarly, nodes next to the border are not in a cluster. Moreover the, cluster heads only choose 3 nodes from target tracking data. Cluster heads determine which 3 nodes to query based on their location in relation to the predicted target location calculated by a previous cluster [24]. When the cluster head has chosen which nodes to query, those nodes are woken up to prepare to detect the approachable target. This algorithm differs even further in that if a cluster head cannot find 3 suitable nodes within its own cluster, it can seek the help of neighbouring clusters and ask for one or more of their nodes to be turned on and report information.

4.2.3. Rooted Tree

Related to the clusters tracking algorithm is the idea of the rooted tree tracking algorithm. This algorithm differs somewhat in that instead of having multiple cluster heads, there is only one head node and it is referred to as the root. The root node is the node that is closest to the target. When a new target location is predicted, if there is another node that is closer to that position than the current root then new node becomes the root. Other nodes around the root work together to form a tree in which their sensed data is collected and passed up through the tree (children to parents) until all data reaches the root [25]. When the root receives all the data it calculates the target position and predicts the new target position as mentioned in the basic target tracking algorithm. The tree itself is reconfigured on every root change. When a root change occurs, a “reconfigure” message is broadcast containing the location of the new root. When a node detects the “reconfigure” message, it detaches itself from the old tree and attaches itself to the new tree by recognizing its neighbour node that is closest to the new root node as its new parent. New data is then collected and passed up the tree for the new root to use.

4.2.4. Particle Filtering

The basic idea behind a particle filtering tracking algorithm is that numerous object state descriptions are saved that contain data necessary to calculate the target position at a certain time. These state descriptions are referred to as particles and each particle has its own weight. The weight of a particle determines how much the data it contains will contribute to the location estimation of an object. When new particles are created, the weights of the pre-existing particles are adjusted and then all the particles are used to calculate a new target location [26]. Eventually, particles with weights that are below a certain threshold are eliminated as duplicate particles.

Some of the particles filtering algorithms differ a little. Certain algorithms have all the particles stored in a central node and this node does all the target location processing [28]. Other algorithms distribute the stored particles across the network nodes.

5. Analysis of Algorithms

This section analyses each target tracking algorithm mentioned with a focus on how it could be improved by combining it with one or more of the ideas presented in the target detection algorithms.

5.1. Simple Triangulation

The main problem with this algorithm lies in the fact that when a node has detected a target and hears two other nodes broadcast the same target detection, the nodes starts to perform the triangulation localization calculation. This means that at the same point of time there will be multiple nodes calculating the same target location. This is a waste of computation and energy resources. This algorithm needs to be improved so that when a node starts to calculate the location of the target, it first broadcasts its intention to calculate the location to all the other nodes, similar to the data sharing that is done in the collaborative coverage mobile detection algorithm. Then all the other nodes that would have performed this same calculation will not perform it once they hear the broadcasted intention of the first node.

5.2. Basic Cluster

The majority of the cluster algorithms had all the nodes on during their target detection phase. It would be better if they instead adopt a technique similar to that used in the wave detection algorithm to detect targets. Since each area of the network is already divided into clusters, each node within the cluster could take turns monitoring the area for the arrival of a target using one of the wave detection methods mentioned. This would increase the lifetime of the network.

However, using one of the mobile detection algorithms is not recommended since the nodes have to register with the cluster head. Depending on the mobile detection algorithm used, the nodes would constantly be moving in and out of clusters. This would cause an increase in the amount of message communications with these nodes in order keep track of the nodes within the cluster. Mobile detection algorithm of this type can be applied to the situation only if the mobility of the nodes is limited and the node always stay within the same cluster.

Additionally, the cluster algorithms could benefit from the basic fusion detection algorithm. Once a target has been detected and all the information is passed to the cluster head the cluster head could use the technique used in the basic fusion algorithm to eliminate any data provide by faulty nodes. This would help make the network less vulnerable to faulty nodes.

5.2.1. Dynamic Clustering

The biggest problem with this algorithm is that it will not work with mobile nodes. The network calculates the sensor node's locations at start-up. Since these locations need to be known by the cluster head and are calculated only once, this prevents any nodes from moving or even being added to the network. This problem could be easily fixed by periodically refreshing the locations of the sensor nodes, although this will decrease the network's lifetime. So it is better to use a detection algorithm and constantly refreshing the nodes' location isn't an optimal solution is better in this case.

5.2.2. Cell Collaboration

This clustering algorithm is already perfectly setup to use the grid detection algorithm for detection of targets. Since the nodes in this algorithm are already divided into cells, the basic structure to use the grid algorithm is already in place. All that needs to be done is to add the grid algorithm for the detection part. Once, this is done the algorithm would be more efficient since the nodes would be alternating sleep schedules with each other in order to detect the target instead of all being awake at the same time.

5.2.3. Probabilistic Localization

This algorithm already sounds like that it has incorporated parts of the hybrid detection algorithm in it. When a target is detected, this algorithm sends a small message to the cluster head. Similarly, when a target is detected in the hybrid algorithm a small yes/no bit message is sent to the decision maker in the network, or cluster head. Then the cluster head uses this data to determine which nodes to query for further data. In determining which nodes to query, it would be a good idea to use an averaging technique similar to that used in the basic fusion detection algorithm in order to eliminate the impact of any data provided by a faulty node. The similarities to the hybrid algorithm continue. When the cluster head determines which nodes to query, the nodes respond back to the cluster head with more detailed information.

However, this algorithm should not be combined with a mobile node detection algorithm because of the way it decides which nodes to query. In this algorithm, the cluster heads create probability tables in order to determine which sensors to query [23]. The use of a table is acceptable until nodes move outside or are added to the system. When this happens, the probability table must be updated. Therefore, the use of a mobile detection scheme would cause a lot of additional computational overhead since these tables are to be constantly refreshed.

5.2.4. Distributed Predictive Tracking

This algorithm is unique in that it is already doing its own target detecting. Instead of having the border nodes being on at all times to detect nodes it would be beneficial to have these nodes adopt a wave or patrol like technique. This way, they would still detect targets but would be consuming a considerably smaller amount of energy.

5.3. Rooted Tree

One of the basic problems with the rooted tree technique is that it doesn't provide any fault tolerance facts. In order to make this algorithm less vulnerable to problems caused by faulty nodes, it should use an averaging technique similar to that used in the basic fusion detection algorithm.

Additionally, this algorithm encounters problems when the velocity of the target is very high. As the velocity of the target increases so does the number of times in which the tree must be reconfigured [25]. This will cause problems when a threshold velocity of the target is reached in which the tree can no longer be generated quickly enough to keep up with the moving target. This problem could be eliminated by altering this algorithm to work more like the patrolling algorithm. Instead of constantly rebuilding the tree, the tree is treated as a path that can have steps (or nodes) along the path added or deleted from it. This way when the target moves, just add a new node to the path and remove any one that is too far away to contribute any longer. The basic nodes needed to collect data are still members of the path and new ones are added. It saves an enormous amount of time and message communication needed to periodically rebuild the tree structure each time.

5.4. Particle Filtering

In order to track targets, this algorithm saves an enormous amount of data in the form of particles. Over time, these particles take up a large amount of memory space. The amount of memory actually used could be decreased if this algorithm is to adopt the idea behind the TBD algorithm. Bayesian Estimation Track Before Detect (TBD) based on particle filtering gives high performance and hybrid detection algorithm stops recording data once the data it currently contains exceeds a threshold to know whether a target is present or not. The particle filtering algorithm could use the same principle. Once enough particle data is collected to exceed some threshold, the node would stop collecting data. This would not only decrease the amount of memory used but also decrease the amount of work to be done to save energy.

5.5. Evaluation of Target Detection and Tracking Algorithms

The best algorithm is established based on the evaluation criteria. The parameters used for evaluation are time, energy, propagation delay, vulnerability, memory used, stability and scalability. Particle Filtering based on the Bayesian TBD estimator algorithm that sounds the best for vehicle detection and tracking in WSNs. Table 4 shows the evaluation summary.

6. Summary of Algorithms

It is observed that, one of the best target detection and tracking algorithms is the combination of the Bayesian Estimation Track before Detect (TBD) Estimator and particle filtering algorithm. The fact that the TBD covers the network area in a short amount of time and enables nodes to sleep at times saves a huge amount of energy. This enables a target to be detected, without too much delay, in an energy efficient manner.

The particle filtering algorithm is also on the right track since the pre-existing particles are adjusted when the new particles are created and hence particles with weights that are below a certain threshold are eliminated as duplicate particles, which also conserves energy. Additionally, the use of several central localization calculating points instead of one helps to reduce the vulnerability of the system to an attacker. It eliminates the one point of failure problem. It also shares the burden of energy usage that one node would face across several nodes.

Particle filtering approach allows all the nodes to share the energy burden. Hence particle filtering algorithm combined with Bayesian Estimation Track before Detect (TBD) Estimator algorithm can be used for target detection and tracking.

7. Conclusions and Future Directions

In this paper an attempt is made to gather the information about the unauthorized vehicle detection and tracking in the battlefield surveillance and also to survey the sensors that are widely available for vehicle detection. The effort is also set to survey and evaluate the detection and tracking algorithms intended for target (eg vehicle or animal) detection and tracking using Wireless Sensor Networks. An overview of each algorithm type was presented. The combinations of target detection algorithms with target tracking algorithms are also discussed. The various combinations of algorithms have been classified and each category is evaluated according to the identified criteria. Particle Filtering based on the Bayesian TBD estimator algorithm is an interesting one for target detection and tracking using WSNs. It has good resource efficiency, propagation delay is minimum. It also increases scalability and particle based approach increases stability. Real time experiments are required in order to conclude whether a Particle Filtering based on Bayesian TBD estimator algorithm will work in a practical scenario. This paper aims to give a brief overview of sensors and algorithms used for Vehicle detection and tracking which directs the future researchers to show new directions used in battlefield surveillance or in any place where human monitoring is not possible.

8. Acknowledgment

The authors would like to thank the Armament Research Board (ARMREB-DRDO) for supporting this Research project by funding.

9. References

[1] Y. C. Tseng, S. P. Kuo, H. W. Lee, and C. F. Huang, “Location tracking in a wireless sensor network by mobile agents and its data fusion strategies,” Lecture notes in computer science, SpringerLink, pp. 2–3, 2003.

[2] E. Y. Luz and A. Mimbela, “Summary of vehicle detection and surveillance technologies used in intelligent transportation systems,” The Vehicle Detector Clearinghouse, Southwest Technology Development Institute (SWTDI) at New Mexico State University (NMSU), Fall 2007. http://www.nmsu.edu/traffic/.

[3] S. Coleri, M. Ergen, and T. J. Koo, “Lifetime analysis of a sensor network with hybrid automata modelling,” Processings of ACM International Workshop on Wireless Sensor Networks and Applications (Atlanta, GA), pp. 98–104, 2002.

[4] D. Jiagen, S. Y. Cheung, C. W. Tan, and V. Pravin, “Signal processing of sensor node data for vehicle detection,” IEEE Intelligent Transportation Systems, pp. 70–75, 2004.

[5] C. Thomas, K. S. Kewel, and R. Parameswaran, "Fault tolerance in collaborative sensor networks for target detection," IEEE Transactions on Computers, pp. 320–333, 2004. http://citeseer.ist.psu.edu/clouqueur03fault.html.

[6] F. Qing, Z. Feng, and L. Guibas, "Lightweight sensing and communication protocols for target enumeration and aggregation," Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pp. 165–176, 2003.

[7] Y. Ting, H. Tian, and A. S. John, "Differentiated surveillance for sensor networks," Proceedings of the 1st international conference on Embedded networked sensor systems, pp. 51–62, 2003. http://www.cs.virginia.edu/papers/diff_surveillance_sn_p51-yan.pdf.

[8] S. S. Ren, Q. Li, H. N. Wang, and X. D. Zhang, "Design and analysis of wave sensing scheduling protocols for object-tracking applications," Lecture notes in computer science, SpringerLink, pp. 228–243, 2005. http://www.cs.wm. edu/~liqun/paper/dcoss05.pdf.

[9] C. Gui and P. Mohapatra, "Virtual patrol: a new power conservation design for surveillance using sensor networks," Information Processing in Sensor Networks, pp. 246–253, 2005.

[10] B. Y. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks. international symposium on mobile ad hoc networking & computing," pp. 300–108, 2005.

[11] M. K. Krishna, H. Hexmoor, and S. Sogani, "A t-step ahead constrained optimal target detection algorithm for a multi sensor surveillance system," IEEE Intelligent Robots and Systems, pp. 357–362 2005. http://arxiv.org/ftp/cs/papers/0505/0505045.pdf.

[12] K. C. Wang and P. Ramanathan, "Collaborative sensing using sensor of uncoordinated mobility," Lecture notes in computer science, SpringerLink, pp. 293–306, 2005. http: //www.ece.wisc.edu/~wander/papers/dcoss05_wang.pdf

[13] E. B. Ermis and V. Saligrama, "Adaptive statistical sampling methods for decentralized estimation and detection of localized phenomena," ACM'05. http://iss.bu.edu/srv/Publications/Manuscript-final.pdf.

[14] R. Gupta and S. R. Das, "Tracking moving targets in a smart sensor network," Vehicular Technology Conference, pp. 3035–3039, 2003. http://www.cs.sunysb.edu/~samir/Pubs/VTC03-long.pdf.

[15] S. J. Davey, M. G. Rutten, and B. Cheung, "A com parison of detection performance for several track-bef ore-detect algorithms," EURASIP Journal on Advance s in Signal Processing, 2008, http://www.hindawi.com /journals/asp/2008/428036.html.

[16] B. Tatiana, H. Wen, K. Salil, R. Branko, G. Neil, B. Travis, R. Mark, and J. Sanjay, "Wireless sensor networks for battlefield surveillance," Land Warfare Conference, 2006.

[17] L. A. Klein, M. K. Mills, and D. R. P. Gibson, "Traffic detector handbook," Operations and Intelligent Transportation Systems Research, 2006. http://www.tfhrc.gov/its/pubs/06108/index.htm.

[18] L. Yu, L. Yuan, G. Qu, and A. Ephremides, "Energy-driven detection scheme with guaranteed accuracy" Information Processing in Sensor Networks, pp. 284–291, 2006.

[19] W. P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," 11th IEEE International Conference on Network Protocols (ICNP'03), November 2003.

[20] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks" IEEE Signal Processing Magazine, pp. 1163–117, 2002.

[21] S. Oh and S. Sastry, "A hierarchical multiple-target tracking algorithm for sensor networks," IEEE Robotics and Automation," pp. 2197–2202, 2005. http://www.eecs.berkeley.edu/~sho/papers/icra05_hmttsn.pdf.

[22] R. B. Richard, R. Parameswaran, and M. S. Akbar, "Distributed target classification and tracking in sensor networks," In Proceedings of the IEEE, Vol. 91, No. 8, pp. 1163–1171, 2003. http://www-net.cs.umass.edu/cs791_sensornets/papers/brooks.pdf.

[23] Y. Zou, and K. Chakrabarty, "Target localization based on energy considerations in distributed sensor networks," 1st IEEE Workshop Sensor Network Protocols and Applications (SNPA'03), pp. 51–58, 2003.

[24] H. Yang and B. Sikdar. "A protocol for tracking mobile targets using sensor networks, sensor network protocols and applications," pp. 71–81, 2003. http://networks.ecse.rpi.edu/~bsikdar/papers/hua_icc03.pdf.

[25] W. S. Zhang and G. H. Cao, "Optimizing tree reconfiguration for mobile target tracking in sensor networks," Infocom, pp. 2434–2445, 2004.

[26] G. Ing, "Distributed particle filters for object tracking in sensor networks," Proceedings of the 3rd international symposium on Information processing in sensor networks pp. 99–107, 2004.

Wireless Sensor Network, 2010, 2, 186-196
doi:10.4236/wsn.2010.22024 Published Online February 2010 (http://www.SciRP.org/journal/wsn/).

Achieving Directionality and Transmit Diversity via Integrating Beam Pattern Scanning (BPS) Antenna Arrays and OFDM

Peh Keong Teh, Seyed Alireza Zekavat
Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, USA
E-mail: {*pteh, rezaz*}@*mtu.edu*

Received July 28, 2009; *revised October* 25, 2009; *accepted November* 19, 2009

Abstract

In this paper, we introduce a novel merger of antenna arrays with scanning beam patterns, and Orthogonal Frequency Division Multiplexing (OFDM) systems. Controlled time varying phase shifts are applied to the antenna array elements mounted at the base station with beam patterns directed toward the desired user. This creates a small beam pattern movement called Beam Pattern Scanning (BPS). In rich scattering environments BPS creates a time varying environment leading to time diversity exploitable at the receiver enhances its probability-of-error performance. Here, we apply OFDM signals to BPS antenna arrays, and we achieve: (1) directionality, which supports Space Division Multiple Access (SDMA); and (2) a time diversity gain, which leads to high performance. We discuss the structure of the base station antenna array and the OFDM receiver that exploits time diversity. We also introduce the merger of BPS and multi-carrier OFDM (MC-OFDM) systems. In MC-OFDM each bit is transmitted over all sub-carriers after serial to parallel conversion. BPS/ MC-OFDM receiver exploits both time diversity inherent in BPS, and frequency diversity inherent in MC-OFDM transmission technique. Simulation results show high Probability-of-error performance is achievable via BPS/OFDM and BPS/MC-OFDM schemes comparing to the traditional OFDM and MC-OFDM, respectively. Simulations also reveal that MC-OFDM system as well as its merger with BPS is capable of mitigating large Peak-to-Average Ratio (PAPR) problem in traditional OFDM system. In addition, performance simulations with coded OFDM (COFDM) and coded MC-OFDM (MC-COFDM) and their merger with BPS are studied.

Keywords: OFDM, MC-OFDM, Antenna Array, Beam Pattern Scanning, Transmit Diversity.

1. Introduction

OFDM (Orthogonal Frequency Division Multiplexing) is an emerging technique capable of high data rate transmission over frequency selective channels without implementation of complex equalizers [1–5]. Due to its inherent benefits, OFDM has been proposed as the basic modulation technique for the 4th generation wireless systems. In an N sub-carriers OFDM system, a block of N information bits are serial to parallel converted and modulated via N orthogonal sub-carriers (each bit over one sub-carrier). They are then summed and transmitted simultaneously [1–5]. This process extends the symbol duration from T_s to NT_s.

MC-OFDM (Multi-Carrier OFDM) is an innovative OFDM transmission technique where each MC-OFDM user's bit is transmitted over all available sub-carriers simultaneously [6–8]. To ensure separability of the bits at the receiver and reduce inter-bit-interference (IBI), orthogonal codes, e.g., Hadamard Walsh codes, are applied to the sub-carriers of each bit. Through MC-OFDM, frequency diversity can be exploited to improve the performance of the system with minimal complexity in the transmitter and receiver [7,8].

Properly designed OFDM and MC-OFDM systems convert the channel to a flat fading channel and eliminate the need for complex equalizers. This is achieved via selecting N such that NT_s becomes much larger than channel delay spread. However, there is a drawback associated with flat fading: If sub-carriers experience deep fade, the bit will be rendered unrecoverable and thereby degrades the system probability-of-error performance. This issue can be resolved via diversity techniques. In MC-OFDM signals are transmitted over all sub-carriers, which allow frequency diversity to be exploited at the receiver [6–8]. Besides frequency diversity, various methods such as: 1) Forward Error Correction Coding at the cost of the overall system throughput [1,2], and 2) transmit diversity has been used to improve the probability-of-error performance of OFDM and MC-OFDM systems [9,10]. In this work, we merge OFDM systems with a transmit diversity scheme created via beam pattern movement (scanning).

The concept of beam pattern movement has been referenced in the literature, some with a different approach, and some with no further exploration. For example, in [11], researchers introduce "jitter diversity". In that work, with a high angular spread at the mobile receiver, the beam pattern is jittered around its usual position to create angle diversity and enhance the receiver performance; and it does not lead to time diversity. In [12], there is a very short (just one line) reference to the idea of antenna pattern movement to create time diversity. The authors state that one can force the antenna to oscillate when the vehicle is traveling at low speeds, channel's fade is slow, and time diversity cannot be exploited.

Recently, a powerful transmit diversity technique has been introduced called beam pattern scanning (BPS) (also known as beam pattern oscillation). In this scheme, antenna arrays are installed at the base station (BS) with their scanning (oscillating) antenna patterns directed toward the desired users [13,15]. A time varying phase shift is applied to each antenna element in order to steer and move the antenna pattern within one symbol duration T_s (or within NT_s in OFDM systems). The beam pattern starts sweeping an area of space at time zero, it returns to its initial position after time T_s (NT_s in OFDM) and repeats its sweeping. The movement of the beam pattern is small, e.g., in the order of 5% of half power beam width (HPBW). Hence, the desired user stays in the antenna array HPBW at all times.

In rich scattering environments, as a result of the departure and arrival of scatterers within the window of antenna beam pattern, BPS creates a time varying channel with a small controlled coherence time T_c with respect to T_s (NT_s) [8,13,14]. This leads to a fast fading channel via which time diversity benefits can be exploited at the receiver [16]. Therefore, BPS is introduced as a transmit diversity scheme that enhances: a) receiver probability-of-error performance via time diversity, and b) wireless network capacity via Spatial Division Multiple Access (SDMA) [17] or spatial filtering interference reduction (SFIR) [18].

In this paper, we merge BPS transmit diversity with OFDM and MC-OFDM systems. This merger achieves: 1) high probability-of-error performance by time diversity induced at the receiver via BPS [8,15,19], and 2) high capacity via directionality created via antenna arrays mounted at the transmitter [17–20], with a 3) low complexity due to the structure of OFDM and BPS. The structure of BPS is simple because the complexity is mainly focused at the base station antenna array.

In this work, we present and discuss: a) the antenna array structure that makes BPS possible for OFDM systems incorporating a number of sub-carriers, b) the OFDM receiver structure capable of exploiting the time diversity induced by BPS, and c) the MC-OFDM receiver capable of exploiting time diversity via BPS and frequency diversity through Multi-carrier scheme. We simulate the probability-of-error performance and the peak-to-average ratio (PAPR) curves for both BPS/OFDM and BPS/MC-OFDM.

Traditional OFDM and MC-OFDM utilizing antenna arrays without BPS scheme are used as benchmark against BPS/OFDM and BPS/MC-OFDM schemes. Adaptive antenna arrays with beam patterns directed towards intended users leads to capacity enhancement via SDMA [20–22] without enhancing the performance. This paper highlights performance benefits achieved through BPS and OFDM systems (OFDM and MC-OFDM) merger. In addition, coded version of OFDM systems and BPS merged systems (BPS/COFDM and BPS/MC-COFDM) are simulated and compared to further underline the performance improvement through BPS merger.

Section 2 introduces OFDM and MC-OFDM systems, BPS technique and the antenna array structure. In Section 3, we present BPS/OFDM and BPS/MC-OFDM received signal and their receiver structure. In Section 4, we present the probability-of-error performance and peak-to-average ratio simulations. Section 5 concludes the paper.

2. OFDM, MC-OFDM, Antenna Array Structure and BPS Techniques

1) *OFDM system*: In OFDM, a block of N bits are transmitted simultaneously over N sub-carriers (each bit over one sub-carrier) after serial to parallel conversion, which converts the duration of transmitted symbols from T_s to NT_s (see Figure 1). The number of sub-carriers N is chosen to ensure flat fading channel at all times (*i.e.*, channel delay spread $T_m << NT_s$), and the sub-carriers are separated by

$$\Delta f = {}^{1}\!\!\diagup_{NT_s} \tag{1}$$

to ensure the orthogonality of sub-carriers and avoid inter-carrier interference at the receiver. The OFDM transmitted signal corresponds to:

$$s_i(t) = \mathrm{Re}\left\{\sum_{n=0}^{N-1} b[n+iN]e^{j2\pi(f_o+n\Delta f)t}\right\}\cdot g_{NT_s}(t-iNT_s) \tag{2}$$

where $b[\cdot]\in\{+1,-1\}$ is the transmitted bit, $i\in\{0,1,2,\ldots\}$ is the i^{th} group of N bits simultaneously converted to par-

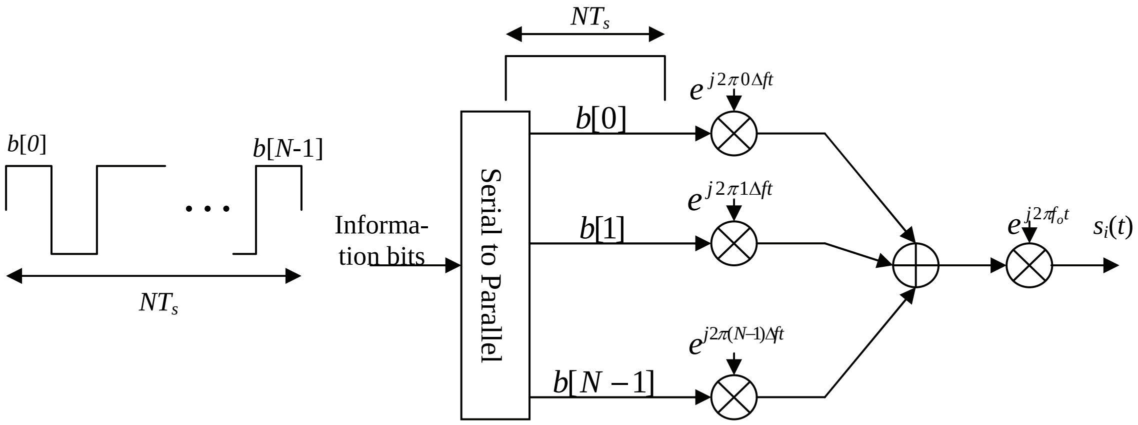
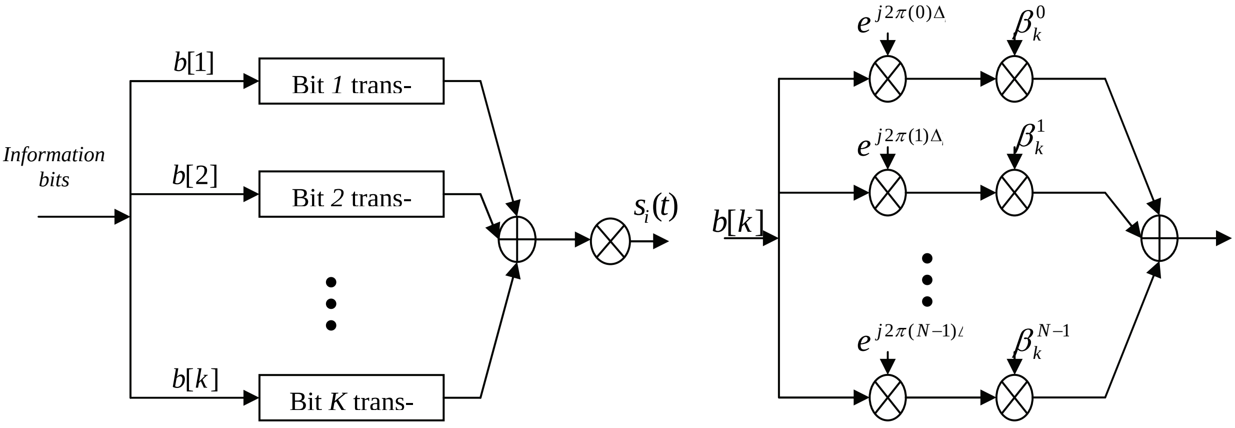


Figure 1. OFDM transmitter structure.



(a) (b)

Figure 2. (a) MC-OFDM transmitter structure; (b) k^{th} bit transmitter design.

allel (n represents the bit number), f_o is the carrier frequency and $g_{NT_s}(t)$ is a rectangular waveform of unity height over zero to NT_s (T_s refers to symbol duration).

2) *The MC-OFDM system:* In MC-OFDM system, all N bits are transmitted simultaneously over all N sub-carriers. To maintain orthogonality across bits, an orthogonal code, e.g., Hadamard Walsh code, is assignedto the sub-carrier of each bit (see Figure 2). By transmitting all N bits simultaneously, the symbol duration is converted from T_s to NT_s. Hence, the MC-OFDM transmitted signal corresponds to:

$$s_i(t) = \mathrm{Re}\left\{ \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} b[k+iN] \cdot \beta_k^n \cdot e^{j2\pi(f_o + n\Delta f)t} \right\} \cdot g_{NT_s}(t - iNT_s) \tag{3}$$

where β_k^n is the k^{th} bit and n^{th} sub-carrier orthogonal codes' element. Other parameters have been defined in (2).

3) *Proposed Antenna Array Structure*: We assume an M-element linear antenna array mounted at the Base Station (B.S.) (Figure 3). In order to move the antenna pattern, a time varying phase, $m\theta(t)$, is applied to the m^{th} antenna array element (Figure 3). Moreover, in Figure 3 the angle ϕ_o represents the direction of the antenna beam pattern (angle-of-arrival of the signal) with respect to the antenna array main axis. Here, for simplicity in presentation, it is assumed that the desired mobile is located at angle $\phi_o = 0$ (*i.e.*, antenna array main axis and beam pattern main axis are overlapped). In this case, the normalized array factor characterizing the resulting antenna pattern corresponds to:

$$AF(t,\phi) = \frac{1}{M}\left[\frac{\sin\left(\frac{M}{2}\gamma(t,\phi)\right)}{\sin\left(\frac{1}{2}\gamma(t,\phi)\right)}\right], \tag{4}$$

where

$$\gamma(t,\phi) = \frac{2\pi}{\lambda_o} d \cdot \sin\phi - \theta(t)\,. \tag{5}$$

Here, λ_o is the wavelength (c/f_o), d is the distance between antenna elements, and $(2\pi d/\lambda_o)\cdot \sin\phi$ represents the phase offset due to the difference in distance between antenna array elements and the mobile.

In general, antenna array half power beam width (HPBW) changes with frequency (sub-carrier). For narrowband OFDM systems, the variation in HPBW is negligible (see Figure 4); however, for wideband and high data rate OFDM systems, e.g., 60 Mb/s data rate, the variation of beam pattern with sub-carriers is considerable (see Figure 4). In order to achieve directionality via sectoring strategy (e.g., in switched beam smart antenna systems), identical beam patterns are required for all N sub-carriers. Since different sub-carriers create different HPBWs, users located near the border of two sectors may experience either: 1) large interference from the beam pattern of unintended sector, or 2) a reduction in power in the desired signal in the beam pattern from the desired sector [15].

We must then ensure that the size of sub-carriers applied to antenna array is small enough that all the sub-carriers placed in the antenna array generate identical patterns. Hence, we introduce the following definition:

Definition 1: Beam Patterns are considered identical if the variation in null-to-null beam width (ΔB.W. in Figure 4) of the beam patterns created via sub-carriers are within 10% of the average null-to-null beam width $(\text{B.W.})_{\text{ave}}$. Based on this definition, it can be shown via simulations that whenever the minimum frequency, f_{min}, and the maximum frequency, f_{max}, applied to an antenna array satisfy

$$\frac{f_{\max.} - f_{\min.}}{f_o} < 0.01 \quad , \tag{6}$$

"identical" beam patterns are observed (see [15]). Considering a contiguous bandwidth for OFDM systems, this criterion corresponds to

$$N \cdot {}^{\Delta f}\!\!\diagup_{f_o} < 0.01 \tag{7}$$

If (6) is not satisfied, the N sub-carriers of the OFDM system should be divided into P groups of $Q \le N$ ($N = PQ$) neighboring sub-carriers such that $Q\Delta f_o/f \le 0.01$. Then, either each set of Q sub-carriers should be applied into a

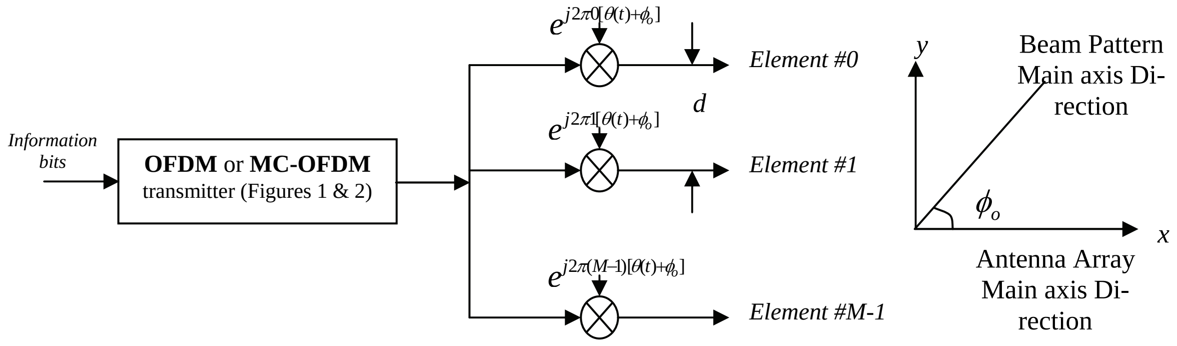
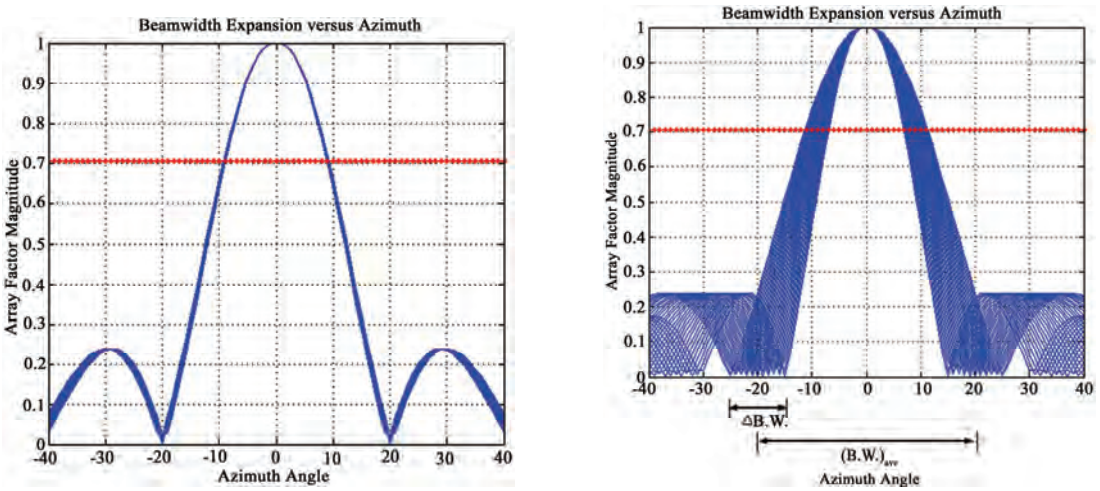

Information bits
OFDM or MC-OFDM
transmitter (Figures 1 & 2)
$e^{j2\pi 0[\theta(t)+\phi_o]}$
$e^{j2\pi 1[\theta(t)+\phi_o]}$
$e^{j2\pi(M-1)[\theta(t)+\phi_o]}$
Element #0
Element #1
Element #M-1
d
y
x
Beam Pattern Main axis Direction
ϕ_o
Antenna Array Main axis Direction

Figure 3. Antenna structure.



(a) (b)

Figure 4. 32 sub-carriers OFDM system, center frequency 4.0GHz. (a) Narrow band system with bandwidth of 200MHz; (b) Wide band system with bandwidth of 2GHz.

unique *M*-element antenna array or unique complex weights should be applied to antenna elements for each set of sub-carriers [15,17,19]. In this work, we assume a narrow band system with $Q = N$, $P = 1$, *i.e.*, all of sub-carriers, $f_n = f_o + n\Delta f$, $n \in \{0,1,\ldots,N\text{-}1\}$ are applied to one antenna array (see Figure 3).

4) *Beam Pattern Scanning Technique:* The scanning of beam pattern is created at the B.S. by applying time varying phase offsets $m\theta(t)$ to the antenna array element $m\in\{1,2,\ldots,M\}$ (Figure 3). The antenna array's beam pattern movement should ensure: 1) constant large scale fading, and 2) *L* independent fades to be generated within each NT_s. This creates an *L*-fold diversity gain at the mobile receiver. After each NT_s, $\theta(t)$ returns the beam pattern to its $t = 0$ position and repeats an identical spatial sweeping movement over the next NT_s period (Figure 5(a)).

In order to ensure constant large-scale fading, the mobile must remain within the antenna array's HPBW over each symbol duration NT_s. This condition corresponds to [22].

$$\left|NT_s \frac{d\phi}{dt}\right| = \kappa \cdot B, \qquad 0<\kappa<1 \qquad (8)$$

where *B* is the HPBW, ϕ is the azimuth angle (direction of arrival), $d\phi/dt$ is the rate of antenna pattern movement, and $NT_s\cdot(d\phi/dt)$ is the amount of antenna pattern movement within NT_s. The parameter κ, $0 < \kappa < 1$, is the control parameter ensures the received antenna pattern amplitude is within the HPBW for the entire symbol duration. The phase offset applied to the antenna array is calculated using (8) leads to [9] (see Figure 5(b)):

$$\theta(t)=\kappa\cdot\frac{2\pi d\cdot|\cos\phi|\cdot B}{\lambda_o NT_s}\cdot\left(t-\frac{NT_s}{2}\right) \qquad (9)$$

Specifically, to solve for (9) from (8) (detailed in [20]), we proceed as follows: 1) Using (5), we solve for ϕ at a fixed value of $\gamma(t,\phi) = \gamma_o$; 2) substituting this ϕ value into (8) and differentiating, we create a differential equation for $\theta(t)$; and (3) finally, we solve the differential equation which leads directly to (9).

Sweeping of the antenna pattern creates a time-varying channel with a coherence time that is a function of scattering environment and may lead to *L* independent fades over NT_s. Geometric-based stochastic channel modeling scheme is used to evaluate the channel diversity gain [11]. Channel coherence time calculated in [15] assuming a medium size city center (e.g., 3 scatterers per 1000m^2), with $0.0005 < \kappa < 0.05$ shows that diversity gain as high as $L \approx 7$ is achievable via BPS scheme. The main assumption in this modeling scheme is:

1) A semi-elliptical coverage with the mobile at its center (suitable when we assume the height of the BS antenna array is close to the height of surrounding buildings);

2) The movement generated by antenna array oscillation is dominant, and hence the movement of mobile and other relative speed object in the environment is ignored;

3) Scatters are assumed to have dimensions in accordance with a known PDF [13];

4) Scatters in the surrounding BS are uniformly distributed;

5) Scatters are consider diffused reflectors which reflect the incident radiation in all direction; and,

6) The signal received at the mobile is sum of horizontally propagated plane waves interacting with just one scatterer.

3. Receiver Design

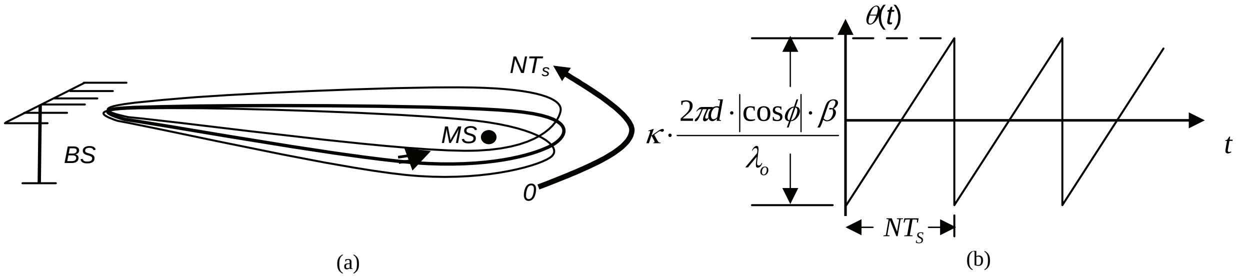
1) *BPS/OFDM:* For simplicity in presentation, we consider $i = 0$ in (2). We represent the OFDM transmitted signal as:

$$s_0(t)=\sum_{n=0}^{N-1} b[n]\cos\left(2\pi(f_o+n\Delta f)t\right), \qquad t\in[0,NT_s] \qquad (10)$$

Applying this signal to the antenna array in Figure 3 the output of the m^{th} element of the antenna array is

$$s_m(t)=\sum_{n=0}^{N-1}\left[b[n]\cos\left(2\pi(f_o+n\Delta f)t+m\theta(t)\right)\right], \qquad t\in[0,NT_s] \qquad (11)$$

The total normalized downlink transmitted signal, considering all antenna elements (all *m*) is:


NT_s
MS
BS
0
$\theta(t)$
$\kappa\cdot\frac{2\pi d\cdot|\cos\phi|\cdot\beta}{\lambda_o}$
t
NT_S

(a) (b)

Figure 5. (a) Beam pattern scanning; (b) Antenna element delay line function.

$$s(t)=\sum_{n=0}^{N-1} b[n]\frac{1}{M}\sum_{m=0}^{M-1}\cos\big(2\pi(f_o+n\Delta f)t+m\theta(t)\big)\ t\in[0,NT_s] \tag{12}$$

Since the range of the control parameter is taken $0<\kappa<0.05$ in (8), the frequency offset induced by $\theta(t)$ in the transmitted signal is less than 5% and can be ignored. Typically, as it is seen in (9), assuming half wavelength spacing across antenna elements, and N = 128, and $B\cong\pi/6$ (for 4 element antenna), and $T_s=10^{-6}$, the maximum frequency drift would be in the order of 100 Hz, which is minimal compared to the typical carrier frequency of 2.4 GHz. At the receiver side, considering the transmit diversity leads to L-fold time diversity, the received signal $[0,NT_s]$ can be divided into time slots $[lNT_s/L,(l+1)NT_s/L]$, $l\in\{0,1,\ldots,L-1\}$, and these time slots demonstrate independent fades. The received signal can then be represented as

$$r_l(t)=\frac{1}{L}\sum_{n=0}^{N-1}\alpha_l^n b[n]\cdot\frac{1}{M}\cdot\sum_{m=0}^{M-1}\cos\big(2\pi(f_o+n\Delta f)t+m\gamma(t,\phi)+\xi_l^n\big)+n_l(t)\quad t\in[lNT_s/L,(l+1)NT_s/L] \tag{13}$$

where, $n_l(t)$ is an additive white Gaussian noise (AWGN), which is independent for different time slots (l), α_l^n is the Rayleigh fade amplitude on n^{th} sub-carrier in the l^{th} time slots, and ξ_l^n is the fading phase offset in the n^{th} sub-carrier and l^{th} time slot (hereafter, this phase offset is assumed to be tracked and removed). The Rayleigh fade amplitudes are considered independent over time and correlated over sub-carriers. The correlation coefficient between carrier n' and n'' is characterized by [23]

$$p_{n',n''}=\frac{1}{1+\big((n''-n')\cdot(\Delta f/(\Delta f)_c)\big)^2} \tag{14}$$

where $(\Delta f)_c$ is the coherence bandwidth of the channel. Moreover, $\gamma(t,\phi)$ in (13) is introduced in (5). Applying the summation over m, (13) corresponds to

$$r_l(t)=\sum_{n=0}^{N-1}\alpha_l^n\cdot b[n]\cdot AF(t,\phi)\cdot\cos\left(2\pi(f_o+n\Delta f)\cdot t+\frac{M-1}{2}\gamma(t,\phi)\right)+n_l^n(t) \tag{15}$$

Here, $AF(t,\phi)$ introduced in (4), is the normalized antenna array factor. Assuming a narrow-beam width antenna array and the mobile is located at $\phi_o=0$, (5) can be approximated by $\gamma(t,\phi)=\gamma(t)=\theta(t)$. Assuming that antenna array peak directed towards the intended mobile at time 0, and with small movements of antenna array pattern over NT_s, the array factor experience over $[0,NT_s]$ is well approximated by $AF(t,\phi)\cong 1$.

The BPS/OFDM receiver is shown in Figure 6. In this figure, "Re" refers to Real Part, and "Im" refers to Imaginary Part. In addition, after the application of

$$v=\sqrt{\frac{1}{2NT_s}}e^{-j\left(2\pi f_o\cdot t+\frac{M-1}{2}\gamma(t)\right)} \tag{16}$$

and returning the OFDM to baseband, a bank of bandpass filters are used to separate the OFDM signal into its N multiple sub-carriers: The baseband signal is integrated over each interval of $t\in[lNT_s/L,(l+1)NT_s/L]$, $l\in\{0,1,\ldots,L\text{-}1\}$ in order to exploit time diversity components created by beam pattern scanning [14,15]. The received signal for each subcarrier, n and l time slot corresponds to

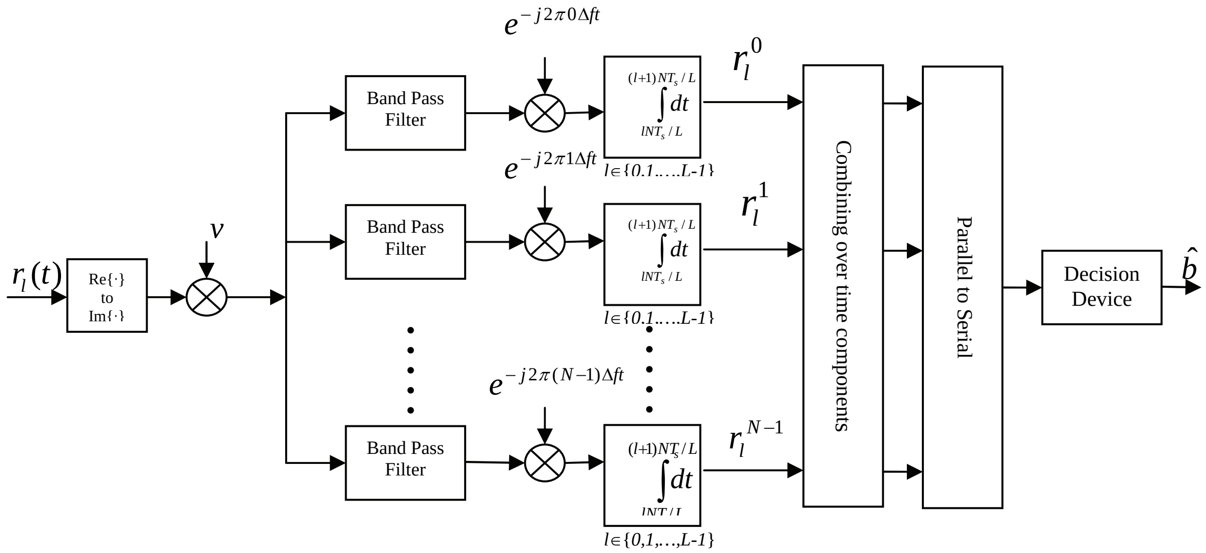


Figure 6. BPS-OFDM receiver.

$$r_l^n = \frac{1}{L}\sqrt{\frac{NT_s}{2}}\alpha_l^n \cdot b[n] + n_l^n, n \in \{0,1,...,N-1\}, l \in \{0,1,...,L-1\} \quad (17)$$

where n_l^n is a zero-mean Gaussian random variables with variance $N_o/2$. The first term in (17) represents l^{th} × n^{th} component of the desired signal and the second term is noise. With $N \times L$ diversity components, N over frequency and L over time, the combiner can be design to utilize different combining techniques such as Equal Gain Combining (EGC) or Maximum Ratio Combining (MRC) (in single user environment, Minimum Mean Square Error Combining (MMSEC) leads to a similar result as MRC).

2) *BPS/MC-OFDM:* Here, we consider $i = 0$ in (3) and we ignore pulse shaping function $g_{NT_s}(t)$. The MC-OFDM transmitted signal corresponds to

$$s_0(t) = \sum_{k=0}^{N-1}\sum_{n=0}^{N-1} b[k]\cdot\beta_k^n \cdot \cos\left(2\pi(f_o + n\Delta f)t\right),\ t \in [0, NT_s] \quad (18)$$

This signal is applied to the antenna array in Figure 3 and the output of the m^{th} element of the antenna array is given by,

$$s_m(t) = \sum_{k=0}^{N-1}\sum_{n=0}^{N-1}\left[b[k]\cdot\beta_k^n \cdot \cos\left(2\pi(f_o + n\Delta f)t + m\theta(t)\right)\right], \quad t \in [0, NT_s], m \in \{0,1,...,M-1\} \quad (19)$$

Considering all antenna elements (all m), the total normalized downlink transmitted signal is

$$s(t) = \sum_{k=0}^{N-1}\sum_{n=0}^{N-1} b[n]\cdot\beta_k^n \cdot \frac{1}{M}\sum_{m=0}^{M-1}\cos\left(2\pi(f_o + n\Delta f)t + m\theta(t)\right) \quad t \in [0, NT_s] \quad (20)$$

Again, in (20) the frequency offset induced by $\theta(t)$ in the transmitted signal is minimal and it is ignored, since the range of the control parameter $\kappa < 0.05$ in (6) creates less than 5% bandwidth expansion. At the receiver, since the transmit diversity leads to an L-fold time diversity, the received signal can be divided into time slots $[lNT_s/L, (l+1)NT_s/L]$, $l \in \{0,1,…,L\text{-}1\}$ and each individual slot demonstrates independent fades. The received signal can be represented as

$$r_l(t) = \sum_{k=0}^{N-1}\sum_{n=0}^{N-1}\alpha_l^n b[n]\cdot\beta_k^n \cdot \frac{1}{M}\cdot\sum_{m=0}^{M-1}\cos\left(2\pi(f_o + n\Delta f)t + m\gamma(t,\phi) + \xi_l^n\right) + n_l(t) \quad t \in [lNT_s/L, (l+1)NT_s/L] \quad (21)$$

where, $n_l(t)$ is an additive white Gaussian noise (AWGN), which is considered independent for different time slots (l), α_l^n is the Rayleigh fade amplitude on the n^{th} sub-carrier in the l^{th} time slots, and ξ_l^n is the fading phase offset in the n^{th} sub-carrier in the l^{th} time slot (hereafter, this phase offset is assumed to be tracked and removed). The Rayleigh fade amplitudes, α_l^n, are independent over time (l) and correlated over sub-carriers (n) with the correlation coefficient between sub-carriers n' and n'' characterized by (14). Applying the summation over m, (21) corresponds to

$$r_l(t) = \sum_{k=0}^{N-1}\sum_{n=0}^{N-1}\alpha_l^n \cdot b[n]\cdot\beta_k^n \cdot AF(t,\phi)\cdot\cos\left(2\pi(f_o + n\Delta f)\cdot t + \frac{M-1}{2}\gamma(t,\phi)\right) + n_l(t) \quad (22)$$

The structure of BPS/MC-OFDM receiver is shown in Figure 7 (Figure 7(b) represents jth bit receiver). The received signal for each sub-carrier, n and time slot l can be represented by

$$r_l^n[j] = \frac{1}{L}\sqrt{\frac{NT_s}{2}}\alpha_l^n \cdot b[j] + \frac{1}{L}\sum_{\substack{k=0\\k\neq j}}^{N-1}\left[\sqrt{\frac{NT_s}{2}}\alpha_l^n \cdot b[k]\cdot\beta_k^n\cdot\beta_j^n\right] + n_l^n, n \in \{0,1,...,N-1\}, l \in \{0,1,...,L-1\} \quad (23)$$

where n_l^n is a zero-mean Gaussian random variables with variance $N_o/2$. In (23), the first term represent the desired signal, the second term represents the inter-bit-interference and the third term represents the noise. The factor *1/L* is the direct consequence of dividing the received signal into L partitions creating L-fold time diversity. The combiner can be designed to utilize different combining techniques in frequency and time domain. Different combining methods in frequency domain are:

1) Equal Gain Combining (EGC)

$$R_{l,EGC} = \sum_{n=0}^{N-1} r_l^n, \quad (24)$$

2) Maximum Ratio Combining (MRC)

$$R_{l,MRC} = \sum_{n=0}^{N-1}\alpha_l^n \cdot r_l^n, \quad (25)$$

3) Minimum Mean Square Error Combining (MMSEC)

$$R_{l,MMSEC} = \sum_{n=0}^{N-1} r_l^n \cdot \left[\frac{\alpha_l^n}{N(\alpha_l^n)^2 + N_0/2}\right] \quad (26)$$

Here, we present only Equal Gain Combining (EGC) in the time domain to exploit the diversity induced by BPS.

4. Simulated Performance

Simulations are provided for MC-OFDM systems with antenna arrays. The assumptions for these simulations are as follow: 1) *N = 32* sub-carriers in the MC-OFDM system; 2) *L = 7* independent fades are achievable as a result of the beam-pattern movement in the duration NT_s

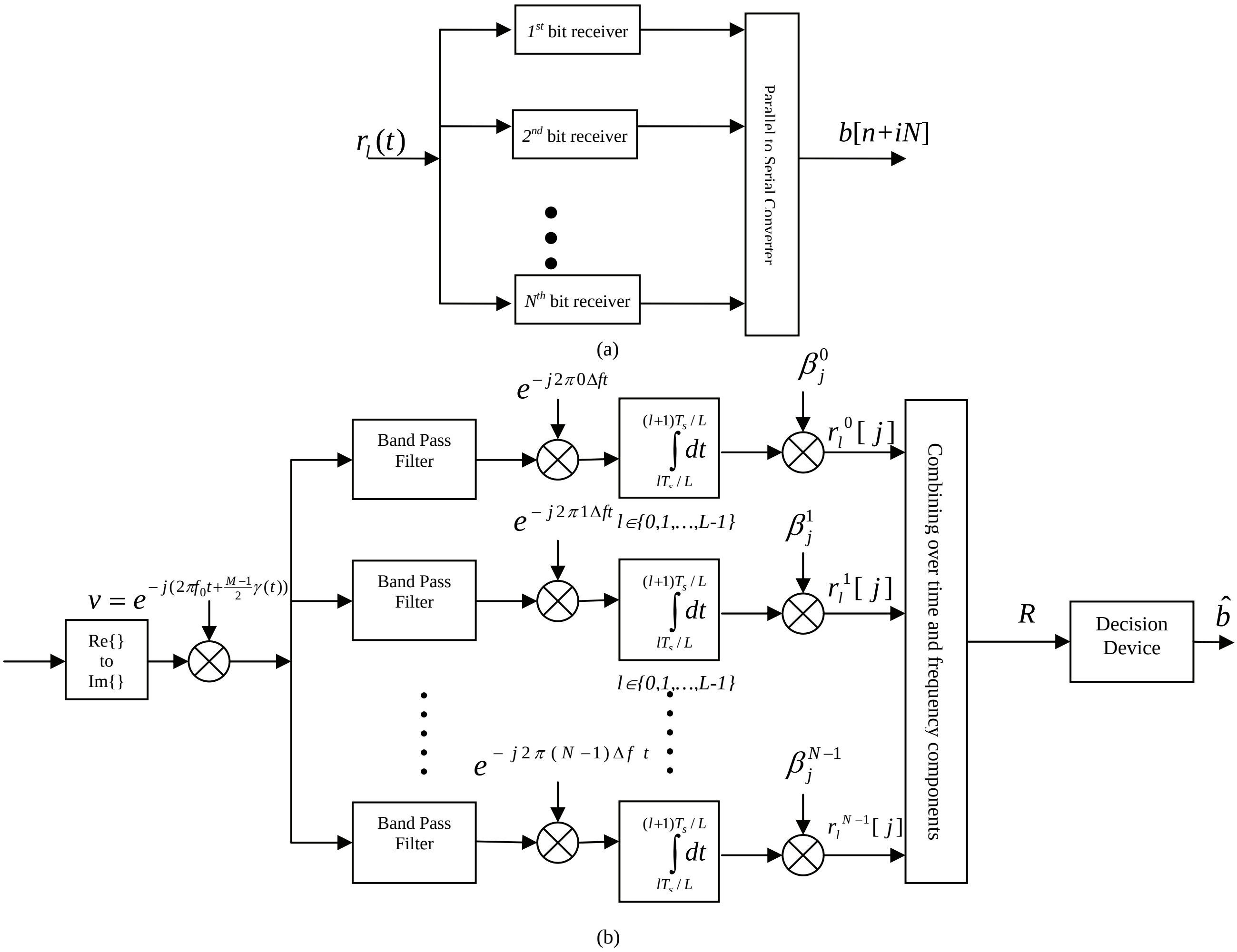


Figure 7. (a) BPS/MC-OFDM receiver structure (b) *j*th bit receiver.

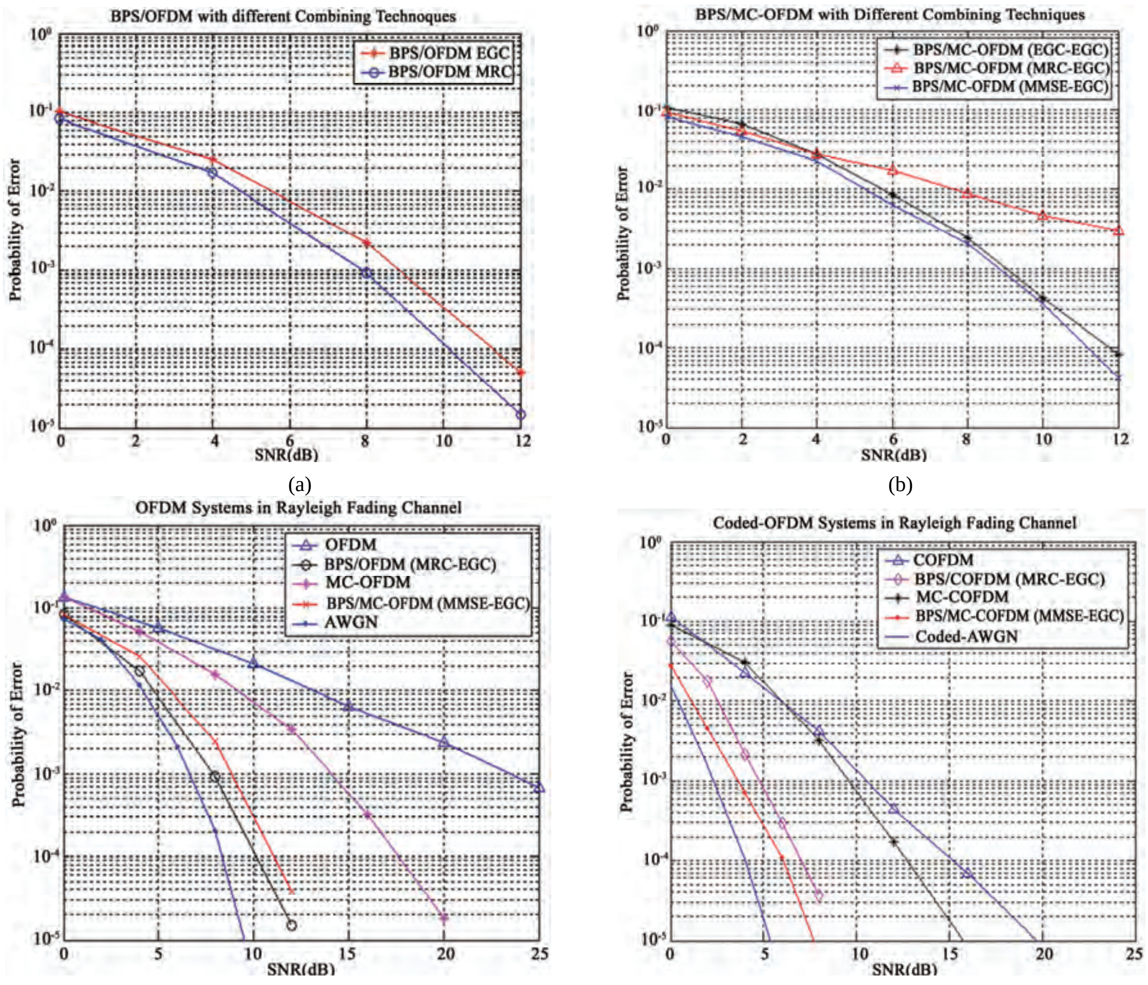
(see [8]); and 3) Frequency-selective channel with four-fold frequency diversity over the entire bandwidth, *i.e.*, in (14) $\Delta f/(\Delta f)_c = 4/N$ ·

Simulation results are presented in Figure 8. Simulations of BPS/OFDM with different combining techniques (Figure 8(a)) in time domain show that MRC provides the best probability of error performance with the penalty of extra complexity. For single user case, MMSEC and MRC lead to similar performance results. MRC leads to a better performance as expected. Hence, MRC is chosen to exhibit the benefits of BPS merger with OFDM systems.

Simulations of MC-OFDM with different combining techniques (Figure 8(b)) reveal that MMSE combining provides the best performance. MRC leads to the worse performance, since in MC-OFDM systems each bit is transmitted over all sub-carriers utilizing orthogonality provided by Hadamard Walsh codes, and the orthogonality is destroyed when MRC is employed [24].

The top curve in Figure 8(c) represents the performance results for the benchmark system, *i.e.*, traditional OFDM system with antenna array and without transmit diversity. The next curve shows MC-OFDM system performance with antenna array without BPS scheme. The BPS/MC-OFDM depicts a 15 dB and 5 dB improvement in performance at the probability-of-error of 10^{-3} compared to the traditional OFDM system and MC-OFDM system, respectively. This performance improvement is generated via time diversity created by beam pattern movement and is exploited using BPS/MC-OFDM receiver. It is also observed that BPS/OFDM performance is about 1 dB better than BPS/MC-OFDM system because MRC combining is the optimal combining when inter-bit-interference (IBI) is not available which lowersthe performance of BPS/MC-OFDM systems, comparatively. However, it has to be noted that BPS/MC-OFDM system mitigates the Peak-to-Average Power Ratio (PAPR) problem faced in OFDM systems.

Figure 8(d) shows the comparison of the coded MC-OFDM (MC-COFDM) with antenna array, with and without scanning. A ½ - rate convolution code is considered and soft Viterbi algorithm is used for decoding. The simulation reveals that up to 6 dB and 5.5 dB improvement



(a) (b)

(c) (d)

Figure 8. (a) Comparison of BPS/OFDM with different combining techniques; (b) Comparison of BPS/MC-OFDM with different combining techniques; (c) Comparison of OFDM and MC-OFDM systems; (d) Comparison of coded OFDM and MC-OFDM systems.

in performance is achievable via BPS scheme at the probability-of-error of 10^{-3} comparing to COFDM and MC-COFDM systems. This clearly reinforce that BPS antenna arrays create time diversity that highly enhances the probability-of-error performance of MC-OFDM systems.

Despite the vast improvement in probability-of-error performance when applying forward error correction coding to MC-OFDM system, the decrease in the throughput makes it a less attractive solution. However, it is clear that BPS/MC-OFDM scheme (without coding) in Figure 8(c) offers a better performance at the probability-of-error 10^{-3}, compared to the traditional MC-COFDM in Figure 8(d), without sacrificing the throughput of the entire system.

Simulations of PAPR in Figure 9 shows that in general, MC-OFDM leads to a lower PAPR compared to OFDM systems. (e.g., 98% of the MC-OFDM transmissions demonstrate PAPR < 9 while it is just 83% for traditional

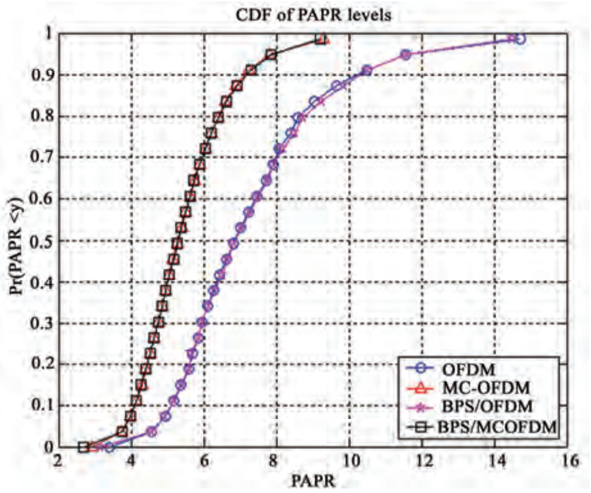


Figure 9. Comparison of PAPR of OFDM and BPS/MC-OFDM.

OFDM systems). The same results are generated for BPS/MC-OFDM systems. The simulations had proven that BPS and MC-OFDM merger is a superior technique capable of delivering high probability-of-error performance as well as reducing the PAPR problem in traditional OFDM systems. This introduces BPS merger with MC-OFDM as a very attractive scheme for future generations of wireless communication systems.

6. Conclusions

The merger of BPS and MC-OFDM (BPS/MC-OFDM) was introduced and the PAPR together with the probability-of-error performance was studied. Time diversity is created using BPS scheme that highly enhances the probability-of-error performance of MC-OFDM systems. Moreover, the inherent unique transmission method of MC-OFDM lowers the PAPR problem compared to traditional OFDM system. This makes BPS/MC-OFDM merger a very competitive technique in wireless communications. The simulations performed also reveal that better performance is achievable via BPS/MC-OFDM (without coding) compared to MC-COFDM without reducing the throughput of the system inherent in MC-COFDM systems. The cost of deploying BPS/MCOFDM system is minimal due to the fact that the complexity of BPS system is mainly at the base station and the receiver complexity itself is minimal because the diversity components enter the receiver serially in time. Therefore, major improvement in performance at a minimal cost makes BPS/MC-OFDM a promising candidate for future wireless systems.

7. References

[1] R. V. Nee and R. Prasad, "OFDM for wireless multimedia communications," Artech House Publisher, Boston, MA. 2000.

[2] S. B. Weinstein and P. M. Ebert, "Data transmission by frequency-division multiplexing using the discrete fourier transform," IEEE Transactions on Communications, Vol. COM-19, No. 5, pp. 628–634, 1971.

[3] L. J. Cimini and Jr., "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing," IEEE Transactions on Communications, Vol. COM-33, No.7, pp. 665–675, 1985.

[4] R. W. Chang, "Orthogonal frequency division multiplexing," U.S. Patent 3, pp. 488–445, filed 1966, issued January 6, 1970.

[5] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications: where fourier meets shannon," IEEE Signal Processing Magazine, Vol. 17, No. 3, pp. 29–48, May 2000.

[6] D. A. Wiegandt, Z. Wu, and C. R. Nassar, "High-throughput, high-performance OFDM via pseudo-orthogonal carrier interferometry spreading codes," IEEE Transactions on Communications, Vol. 51, No. 7, pp. 1123–1134, July 2003.

[7] B. Natarajan, C. R. Nassar, S. Shattil, Z. Wu, and M. Michelini, "High performance MC-CDMA via carrier interferometry codes," IEEE Transactions on Vehicular Technology, Vol. 50, No. 6, pp. 1344–1353, November 2001.

[8] P. K. Teh and S. A. Zekavat, "Merging multi-carrier OFDM and beam pattern scanning smart antennas: achieving low PAPR, high performance and high directionality," Proceedings 2003 Wireless Networking Symposium, The University of Texas at Austin, October 22–24, 2003.

[9] Y. G. Li, J. Chuang, and N. R. Sollenberger, "Transmit diversity for OFDM systems and its impact on high-rate data wireless networks," IEEE Journal of Selected Areas in Communications, Vol. 17, pp. 1233–1243, July 1999.

[10] D. Agarwal, V. Tarokh, A. Naguib, and N. Seshadri, "Space-time coded OFDM for high data rate wireless communication over wideband channels," in Proceedings 48th IEEE Vehicular Technology Conference, pp. 2232–2236, Ottawa, Canada, 18–21 May 1998.

[11] O. Norklit, P. C. F. Eggers, and J. B. Anderson, "Jitter diversity in multipath environments," IEEE 45th Vehicular Technology Conference, VTC'95, Vol. 2, pp. 853–857, 25–28 July 1995.

[12] W. C. Wong, R. Steele, B. Glance, and D. Horn, "Time diversity with adaptive error detection to combat rayleigh fading in digital mobile radio," IEEE Transactions on Communications, Vol. COM-31, No. 3, pp. 378–387, March 1983.

[13] S. A. Zekavat and C. R. Nassar, "Antenna arrays with oscillating beam patterns: characterization of transmit diversity using semi-elliptic coverage geometric-based stochastic channel modeling," IEEE Transactions on Communications, Vol. 50, No. 10, pp. 1549–1556, October 2002.

[14] S. A. Zekavat and C. R. Nassar, "Achieving high capacity wireless by merging multi-carrier CDMA systems and oscillating-beam smart antenna arrays," IEEE Transactions on Vehicular Technology, Vol. 52, No. 4, pp. 772–778, July 2003.

[15] S. A. Zekavat, C. R. Nassar, and S. Shattil, "Oscillating beam adaptive antennas and multi-carrier systems: Achieving transmit diversity, frequency diversity and directionality," IEEE Transactions on Vehicular Technology, Vol. 51, No. 5, pp. 1030 –1039, September 2002.

[16] A. M. Sayeed and B. Azhang, "Joint multipath-doppler diversity in mobile wireless communications," IEEE Transactions on Communications, Vol. 47, No. 1, pp. 123–132, January 1999.

[17] J. C. Liberti, Jr., and T. S. Rappaport, "Smart antennas for wireless communications: Is-95 and third generation CDMA applications," Prentice Hall, PTR, Upper Saddle River, NJ, 1999.

[18] J. Fuhl, A. Kuchar, and E. Bonek, "Capacity increase in cellular PCS by smart antennas," IEEE 47th Vehicular Technology Conference, VTC'97, Vol. 3, No. 10, pp. 1962–1966, 1997.

[19] P. K. Teh and S. A. Zekavat, "A merger of OFDM and antenna array Beam Pattern Scanning (BPS): achieving directionality and transmit diversity," Proceedings IEEE 37th Asilomar conference on Signals, Systems and Computers, Asilomar, CA, November 9–12, 2003.

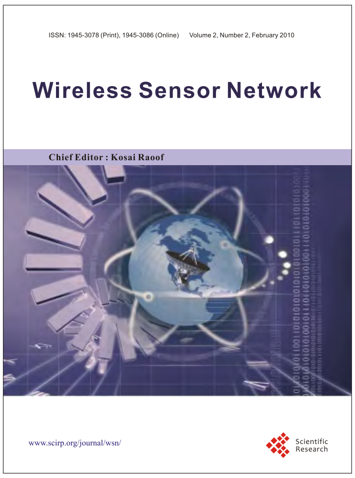
[20] A. Kavak, "Adaptive antenna arrays for downlink capacity increase in third generation wireless CDMA," in IEEE Radio and Wireless Conference (RAWCON'01), pp. 77–80, Boston, MA, August 2001.

[21] A. F. Naguib, A. Paulraj, and T. Kailath, "Capacity improvement with base-station antenna arrays in cellular CDMA," IEEE Transactions on Vehicular Technology, Vol. 43, No. 3, pp. 691–698, August 1994.

[22] S. A. Zekavat, C. R. Nassar, and S. Shattil, "Smart antenna spatial sweeping for combined directionality and transmit diversity," Journal of Communications and Networks (JCN), Special Issue on Adaptive Antennas for Wireless Communications, Vol. 2, No. 4, pp. 325–330, December 2000.

[23] J. W. C. Jakes, "Microwave mobile communications," John Wiley, New York, NY, 1974.

[24] J. M. Auffray and J. F. Helard "Performance of multicarrier CDMA technique combined with space-time block coding over rayleigh channel," IEEE 7th International Symposium on Spread-Spectrum Technology, Vol. 2, pp. 348–352, 2–5 September 2002.


ISSN: 1945-3078 (Print), 1945-3086 (Online) Volume 2, Number 2, February 2010
Wireless Sensor Network
Chief Editor : Kosai Raoof
www.scirp.org/journal/wsn/
Scientific Research

Wireless Sensor Network (WSN)

Call For Papers

http://www.scirp.org/journal/wsn

ISSN 1945-3078 (Print) ISSN 1945-3086 (Online)

WSN is an international refereed journal dedicated to the latest advancement of wireless sensor network and applications. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these areas.

Editor-in-Chief

Dr. Kosai Raoof , GIPSA LAB, University of Joseph Fourier, Grenoble, France

Subject Coverage

This journal invites original research and review papers that address the following issues in wireless sensor networks. Topics of interest are (but not limited to):

- Network Architecture and Protocols
- Self-Organization and Synchronization
- Quality of Service
- Data Processing, Storage and Management
- Network Planning, Provisioning and Deployment
- Integration with Other System
- Software Platforms and Development Tools
- Routing and Data Dissemination
- Energy Conservation and Management
- Security and Privacy
- Developments and Applications
- Network Simulation and Platforms

We are also interested in short papers (letters) that clearly address a specific problem, and short survey or position papers that sketch the results or problems on a specific topic. Authors of selected short papers would be invited to write a regular paper on the same topic for future issues of the WSN.

Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. Authors are responsible for having their papers checked for style and grammar prior to submission to WSN. Papers may be rejected if the language is not satisfactory. For more details about the submissions, please access the website.

Website and E-Mail

http://www.scirp.org/journal/wsn

Email: wsn@scirp.org

Wireless Sensor Network, 2010, 2, 93-196
Published Online February 2010 in SciRP (http://www.SciRP.org/journal/wsn/).

TABLE OF CONTENTS

Volume 2 Number 2 February 2010

Evaluation of Multiusers' Interference on Radiolocation in CDMA Cellular Networks
A. J. Bamisaye, M. O. Kolawole, V. S. A. Adeloye.. 93

Energy Harvesting Strategy Using Piezoelectric Element Driven by Vibration Method
D.-G. Kim, S.-N. Yun, Y.-B. Ham, J.-H. Park.. 100

Linear Pulse-Coupled Oscillators Model—A New Approach for Time Synchronization in Wireless Sensor Networks
Z. L. An, H. S. Zhu, M. L. Zhang, C. N. Xu, Y. J. Xu, X. W. Li... 108

K-Nearest Neighbor Based Missing Data Estimation Algorithm in Wireless Sensor Networks
L. Q. Pan, J. Z. Li.. 115

Recharging Sensor Nodes Using Implicit Actor Coordination in Wireless Sensor Actor Networks
M. Sharifi, S. Sedighian, M. Kamali.. 123

Finding the Optimal Percentage of Cluster Heads from a New and Complete Mathematical Model on LEACH
A. B. M. Alim Al Islam, C. S. Hyder, H. Kabir, M. Naznin.. 129

Coordination for Networks of Dynamic Agents with Time-Varying Delays
H. W. Yu, B. S. Zhang, Y. F. Zheng.. 141

A Novel Approach for Finding a Shortest Path in a Mixed Fuzzy Network
A. Tajdin, I. Mahdavi, N. M. Amiri, B. S.Gildeh, R. Hassanzadeh... 148

Tree Based Energy and Congestion Aware Routing Protocol for Wireless Sensor Networks
A. H. Mohajerzadeh, M. H. Yaghmaee... 161

An Energy-Efficient Access Control Algorithm with Cross-Layer Optimization in Wireless Sensor Networks
Z. Chen, S. Q. Li.. 168

A Study on Vehicle Detection and Tracking Using Wireless Sensor Networks
G. Padmavathi, D. Shanmugapriya, M. Kalaivani.. 173

Achieving Directionality and Transmit Diversity via Integrating Beam Pattern Scanning (BPS) Antenna Arrays and OFDM
P. K. Teh, S. A. Zekavat.. 186

9771945307008 07