

Development of Link Cost Function using Neural Network Concept in Sensor Network

Yujin Lim¹ and Sanggil Kang²

¹Department of Information Media, University of Suwon
San 2-2, Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do, 445-743, Korea
[e-mail: yujin@suwon.ac.kr]

²Department of Computer Science and Computer Engineering, Inha University
253 Younghyun-dong, Nam-gu, Incheon, 420-751, Korea
[e-mail:sgkang@inha.ac.kr]

*Received August 7, 2010; revised December 8, 2010; accepted January 5, 2010;
published January 31, 2011*

Abstract

In this paper we develop a link cost function for data delivery in sensor network. Usually most conventional methods determine the optimal coefficients in the cost function without considering the surrounding environment of the node such as the wireless propagation environment or the topological environment. Due to this reason, there are limitations to improve the quality of data delivery such as data delivery ratio and delay of data delivery. To solve this problem, we derive a new cost function using the concept of Partially Connected Neural Network (PCNN) which is modeled according to the input types whether inputs are correlated or uncorrelated. The correlated inputs are connected to the hidden layer of the PCNN in a coupled fashion but the uncoupled inputs are in an uncoupled fashion. We also propose the training technique for finding an optimal weight vector in the link cost function. The link cost function is trained to the direction that the packet transmission success ratio of each node maximizes. In the experimental section, we show that our method outperforms other conventional methods in terms of the quality of data delivery and the energy efficiency.

Keywords: Data forwarding method, link cost function, partially connected neural network, sensor network

1. Introduction

Wireless sensor network (WSN) is a multi-hop wireless network whose main purpose is to deliver sensing data collected from multiple sensors to one or more data collecting devices (or sinks). Despite of the power and resource limitations, WSN has attracted the attention of many researchers in recent years [1]. WSN has a wide range of applications [2]. Especially, WSN is useful in continuous information acquisition in inaccessible or perilous areas. When a sensing area is far away from remote monitoring center, sensing data collected at sinks have to be delivered to remote monitoring center through an intermediate system.

The issues on constructing WSN, such as the deployment of sensors/sinks and the data collection of sinks from sensors, are well addressed in the literatures of WSNs [3][4]. The distribution network is responsible for delivering gathered data from sinks to the remote monitoring center through an intermediate system, i.e., the gateway (GW) connected to an external network like the global Internet. To reduce the deployment and management cost of the distribution network, wireless multi-hop communication paradigm is often employed. Since the monitoring center is far away from sinks and it requires delivery of a large volume of information automatically and continuously, the distribution network needs to be designed soundly. The data delivery problem for the distribution network is one of the fundamental issues. Since the GW is connected to the monitoring center through a high-speed wired network, the data delivery between them is highly reliable. The problem of data delivery in the distribution network is the same as the problem of data collecting from sinks to the GW. The data forwarding technology is one of the solutions to the problem in an environment where a single GW gathers data from the static or mobile sinks sparsely populated .

In general, data delivery in WSN is based on the premise that data from sensors to sink are loss- tolerant due to the sheer amount of correlated data [5]. However, in the distribution network, data delivery from sinks to the GW is sensitive to the data loss because the sink sends the aggregated data from a specific area to the GW. Besides, since the remote monitoring center takes actions based on the information provided by the distribution network, the aggregated data from sinks need to be promptly delivered to the monitoring center.

To improve the quality of data delivery in terms of data loss and delay of data delivery in the distribution network, some researchers [6][7] have developed various methods. They model an optimal cost function using more than one input metric, to select next delivering neighbor node. The main drawback of their methods lies in determining the optimal coefficients in the cost function, without considering of the wireless propagation environment or the topological environment around the node. To avoid the drawback, we first derive a new cost function by modifying Neural Network (NN) concept [8] and then determine the optimal coefficients (or weights) in the cost function. Since the conventional NNs do not consider whether inputs are correlated or uncorrelated among them, their performances are limited. In the process of the derivation, the correlated inputs are connected to the hidden layer in a coupled fashion and the uncorrelated inputs in an uncoupled fashion. To determine the optimal weights, we train the cost function in the direction to maximize packet transmission success ratio (PTSR) of each node. Since each packet transmission or retransmission can increase a node's energy consumption, we are able to reduce the energy consumed per packet delivery by maximizing PTSR. In the experimental section, we show the improvement on the quality of data delivery and energy efficiency of our method by comparing with other conventional methods.

This paper is organized as follows. Section 2 presents the data collection problem from sinks to a GW in WSN and explains drawbacks of related works. In addition, we explain a fundamental concept of Neural Network in Section 2. In Section 3, our link cost function using NN concept is derived in detail. In Section 4, the performance of the link cost function is evaluated from the NS-2 based simulation results, and finally we conclude in Section 5.

2. Problem Statement

2.1 Data Collection Problem

The main research issues in WSN are data forwarding from sensors to sink and extension of the network lifetime. Literatures of the data forwarding can be classified into the mobile sink approach [9][10] and the relay node approach [11][12]. In the former approach, the mobile sink goes back and forth between sensors and a GW, to gather sensing data and to deliver the data to the GW. It takes time for the mobile sink to travel into a transmission range of the GW. In the meantime, the GW cannot collect the sensing data. Considering that sinks are sparsely populated even if sensors are densely deployed, the latter approach seems the most suitable solution for the data collection between sinks and the GW. Relay Node (RN) is supposed to only forward data from source to destination, not to generate or consume data. One of the important issues in the relay node approach lies in the optimal placement of RNs to guarantee data delivery. In data collection between sensors and a sink, the optimal place of each RN can be computed by considering locations of sensors and sink because information sources (i.e., sensors) are not mobile. In data collection between sinks and the GW, it is not possible to optimally locate RNs because it is difficult to predict the trajectory of mobile sinks and the number of sinks. From the rationale, we assume RNs are deployed randomly in the distribution network.

Usually, the path management methods model a cost function using more than one input metric to improve their performance [13-20]. ETX (Expected Transmission Count) [13] finds high-throughput data delivery path using the delivery ratios of transmission successfully sent and received. ETX may decrease the energy consumed per packet as each transmission or retransmission may increase a node's energy consumption. ETT (Expected Transmission Time) and WCETT (Weighted Cumulative ETT) [14] consider metrics such as packet size or link bandwidth, as well as ETX. PRR-d (Packet Reception Rate-distance) [15] models a cost function using signal-to-noise ratio and frame length to improve the delivery rate and energy efficiency. In [16-20], the link cost is computed using the energy-based inputs such as network lifetime and remaining energy of a node. Even though the above methods use the input metrics reflecting environments around each node, the coefficients in the cost functions are unified for all nodes. Since the environments are different from each other, the performance is degraded when the unified coefficients are employed to determine the data delivery path.

To obtain the customized coefficients for environments around each node instead of the unified coefficients, NN has been employed. SSR (Self-Selective Routing) [21] finds the next node with the smallest number of hops to the destination using the lecture hall algorithm originated in the field of NN. SSR uses the hop distance to select the next forwarding node as input metric and then estimates the hop distance from a node to the destination, using NN technique. From the above rationale, SSR is a kind of the conventional method using hop distance in computing the cost function. In the experimental section, we will compare the performance of our method with that of SSR.

The quality of data delivery is affected by not only hop distance to the destination and remaining energy of nodes but also wireless channel condition and channel contention between nodes. However, most conventional methods determine the unified coefficients in the cost function without considering the surrounding environment of a node. To solve this problem, we derive a new cost function using the concept of NN to improve the quality of data delivery and the energy efficiency.

2.2 Neural Network

Neural Network (NN) is a system in which computational units analogous to the human brain are interconnected. Fully Connected Neural Network (FCNN) is a class of NNs whose structure has input, hidden, and output layers, as seen in Fig. 1-(a). They can be defined as neural networks with acyclic connections from the input layer to the output layer.

For input-output mapping problems [22][23], FCNN has been commonly used as a matter of course, since it usually does not need a priori information about data.

FCNNs work on complicated input-output mapping problems [24] even when data is corrupted; it has been proved that they can be applied to real applications [25][27]. Nevertheless, we need to study them further because of their black-box learning style. A couple of concerns when FCNNs are used for mapping problems can be summarized as follows:

- **Difficulty in understanding input-output relationship:** The black-box style learning models a system by adjusting internal weights without considering intrinsic relationship of input and output. Because of their learning style, it is difficult to understand an input-output relationship even though FCNNs have been attractive for modeling a system when the input-output relationship is unknown due to the learning style.
- **Generalization:** The performance of FCNNs for unseen data during training may not be as good as we expected because of their complexity.

Noyes [28] mentioned that the generalization ability is the most valuable feature of NNs so that many researchers and scientists have studied it. Noyes summarized the various approaches to improving the generalization of NNs. As one approach researchers have studied ways to simplify NNs, and it has been found that NNs with a simple structure perform than complicated ones in many cases.

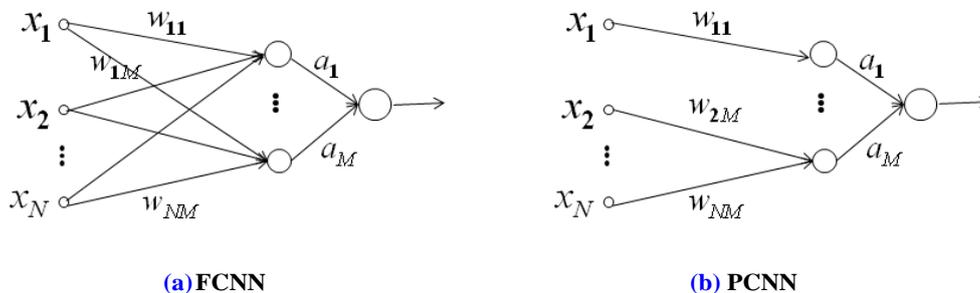


Fig. 1. Examples of FCNN and PCNN.

To improve performance, Partially Connected Neural Network (PCNN) has been tried. PCNN is defined as an NN in which there are missing connections of nodes between adjacent layers. Usually, PCNNs [29][30] have been used in special cases in which the relationship between input and output is known to some extent. For example, Y.Y. Chung et al. [29]

obtained a decision tree [31] by using the classifier program called C4.5 [32] in order to structure a PCNN by knowing the number of neurons and how the neurons are supposed to be connected within the PCNN. They obtained a better performance when a PCNN rather than an FCNN was used.

In our previous work [33], we structured a PCNN according to the input types for generating outputs. To build PCNN, we need to correctly identify the input type of each input. For example, x_1 is an uncoupled input, while other inputs are coupled inputs as seen in Fig. 1-(b). The identification of input type includes an analysis of input sensitivity changes. Here, the input sensitivity [34] can be defined as the quantitative or qualitative influence of outputs due to a variation in the inputs. If all inputs are coupled, an FCNN is a good choice. However, if there are uncoupled inputs, a PCNN can achieve better performance than that of an FCNN.

3. Link Cost Function using Partially Connected Neural Network

Since the packet loss probability in wireless multi-hop communication environment increases with the number of hops [35], we choose the hop distance from an RN to the GW as one of the metrics for path management. The GW periodically floods a probe message over the entire network so that each RN can infer the hop distance from itself to the GW through each of its neighbor RNs. Since the topology of the distribution network is static in terms of hop distance, flooding interval of a probe message is set to be large. The overhead of its periodic flooding is negligible. It is well known that packet loss is due to either collision or weak signal [36]. Each RN periodically sends a HELLO message to its neighbor RNs as its heartbeat. By exchanging HELLO messages among RNs, each RN measures Received Signal Strength (RSS) and HELLO Message Reception Ratio (HMRR) of its neighbor RNs. HMRR represents the ratio of the number of HELLO message received from a neighbor RN to the number of the Hello message sent by the RN. HMRR reflects the impact of channel contention from neighbor RNs. Suppose that each RN sends a HELLO message for every 100 milliseconds. A RN measures HMRR for each neighbor RN every one second, e.g., 6/10 for one neighbor RN, 9/10 for the other neighbor RN. The measured RSS and HMRR are normalized using the maximum value of the measured RSSs or HMRRs for one second in order to scale down into one. The inferred hop distance is also normalized using the network diameter. Using the normalized RSSs, HMRRs, and hop distances for neighbor RNs, the link cost function calculates link costs for neighbor RNs and the RN with maximum link cost is selected as the next data forwarding RN (i.e., next-hop RN).

When a mobile sink enters to or a static sink is powered on in the service area of an RN and the sink sends the aggregated data to the RN, a path setup from the RN to the GW needs to deliver the data from the sink to the GW. Since more than one RN may be in a transmission range of a sink, the sink needs to select its next-hop RN on the path from the sink to the GW. The sink sends a request message to its neighbor RNs by one-hop flooding. Then, the RN receives the request message and it sends a response message having the maximum link cost among its neighbor RNs. When the sink receives the response messages, the sink selects the RN with the largest link cost as its next-hop RN and it sends a confirmation message to the selected RN. After being selected, the next-hop RN selects its next-hop RN and forwards data received from the sink to its next-hop RN. It is repeated until the data is delivered to the GW.

To improve the quality of data delivery in terms of data loss and delay of data delivery, it is important to determine an optimal link cost function to select the next-hop RN. We

define the optimal cost function as maximizing PTSR for each node. In this subsection, we derive the link cost function to determine the next-hop RN out of neighbor RNs by exploring the NN modeling philosophy.

3.1 Derivation of Link Cost Function

The link cost function depends on the input features based on the characteristics of wireless propagation, channel contention, and topological environment surrounding an RN such as RSS denoted as x_1 , HMRR denoted as x_2 , and the hop distance denoted as x_3 . In this paper, *input* and *input feature* are exchangeable. Also, we generalize the number of inputs during derivation of the cost function because the number of inputs is varied according to applications.

The link cost function can be characterized as a nonlinear function of a weighted sum of the inputs as seen in Eq. (1). In general, each weight value is determined by the importance of the corresponding input.

$$Cost_i = f(X_i \cdot W^T), \quad (1)$$

where $Cost_i$ is the link cost of the i^{th} neighbor RN out of N neighbor RNs and f is a nonlinear function. X_i is the input vector collected from the i^{th} neighbor RN, composed of $[x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ (n is the number of inputs) and W is the corresponding weight vector composed of $[w_1, w_2, \dots, w_n]$. Also, T is the notation of vector transpose. For fair comparison in the function, we normalize each input into the range in $[0, 1]$ using min. and max. value of each input samples. Eq. (1) can be represented in a two layered NN in which the input layer consisted of input features and the output layer with the activation function f as seen in Fig. 2.

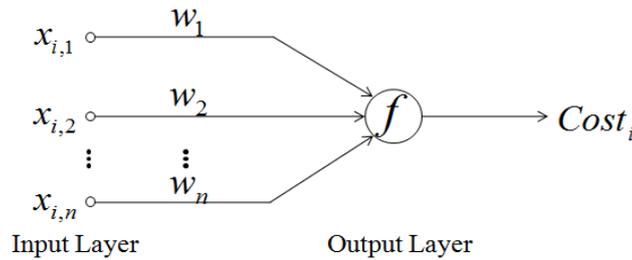


Fig. 2. Link cost function represented in a two-layered neural network.

As in Fig. 2, we employ \log function as the nonlinear function f because of its promising characteristic. By using \log function as in Eq.(2), many natural processes have a history dependent progression in which it begins small and accelerates to some point and then approaches to a saturation point over input features.

$$Cost_i = \log(X_i \cdot W^T), \quad (2)$$

Now, let's discuss about the connectivity of the inputs in the network. In Fig. 2, all inputs are fully connected to the function. It is just like *black-box* style connection which is commonly used in NN. However, we intuitively know that some inputs are highly correlated to generate the output of the cost function. For instance, the higher RSS becomes, the smaller the degree of affection by the slow fading and the fast fading becomes [37]. In other words, if an RN has high RSS(x_1) value then it is highly possible to have high HMRR(x_2). To take this into the consideration, we connect the inputs in the coupled and uncouple connection style according to whether inputs are correlated or

uncorrelated to the hidden layer which is the output layer in Fig. 2, as seen in Fig. 3. There is no weight on the connections between the hidden layer and the output layer. For instance, the connection of x_1 and x_2 is coupled and that of x_3 is uncoupled in the PCNN.

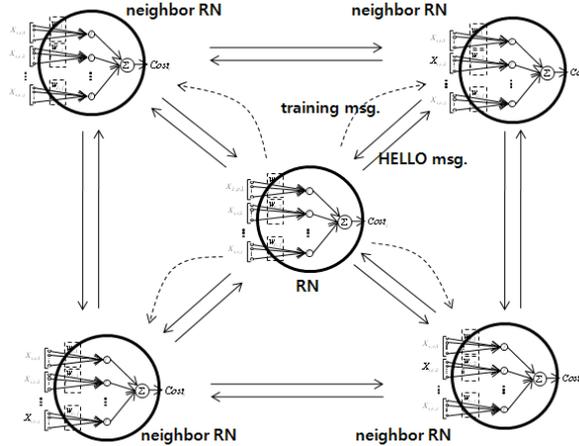


Fig. 3. Link cost function represented in PCNN.

From Fig. 3, our derived cost function can be formularized as Eq. (3).

$$Cost_i = \sum_j \log(X_{i,c,j} \cdot W_{c,j}^T) + \sum_k \log(x_{i,u,k} \cdot w_{i,k}^T), \quad (3)$$

where $X_{i,c,j}$ and $x_{i,u,k}$ are the j^{th} coupled(c) input vector set and the k^{th} uncoupled(u) input collected from the i^{th} neighbor RN respectively. $W_{c,j}$ and $w_{i,k}$ are the corresponding weight vector set and weight respectively. For instance, in our inputs, we have one coupled input vector set, $[x_{i,1}, x_{i,2}]$ and only one uncoupled input $x_{i,3}$. From Eq. (3), the optimal performance of the cost function depends on the proper weight vector.

3.2 Adaptation of Weights

To find an optimal weight vector, we imitate the training process of finding the optimal weights in the link cost function. Each weight in the cost function means the importance of each input for producing the cost function. Thus, each weight can be obtained by weight sensitivity with respect to the cost value. The weights are adapted to the direction in which the mean error between target PTSRs and estimated PTSRs of RNs is minimized. Here, the target PTSR means the ratio of the number of training packets transmitted successfully to the RN with the total number of training packets sent to the RN. Suppose that a RN sends five training packets for every 100 milliseconds. The RN received the training packets notifies the sending RN of a list of the received training packets using HELLO message. The sending RN receives HELLO message and measures the PTSR. And the estimated PTSR is the output of the cost function. Based on the NN training process with the rationale, the cost function is trained with the following steps:

- i. Set to 1s' to each weight as an initial weight vector and we call it old weight vector denoted as $W_{old} = \{w_{1,old}, w_{2,old}, \dots, w_{n,old}\}$, where n is the number of inputs.
- ii. Estimate PTSR for each RN which is the output of the cost function in Eq. (3).
- iii. Calculate mean square error between target PTSRs and estimated PTSRs of neighbor RNs as seen in Eq. (4).

$$E_{W_{old}} = \frac{1}{N} \sum_{i=1}^N (PTSR_i - \overline{PTSR}_{i,W_{old}})^2, \quad (4)$$

where $PTSR_i$ and $\overline{PTSR}_{i,W_{old}}$ are target PTSR and estimated PTSR using W_{old} of neighbor RN i respectively. The N is the number of neighbor RNs.

iv. Change each weight using the delta rule [8] with learning ratio η as in Eq. (5).

$$W_{k,new} \leftarrow W_{k,old} + \eta \cdot \frac{\partial E_{W_{old}}}{\partial W_{k,old}}, k = 1, 2, \dots, n. \quad (5)$$

v. We can obtain new weight vector denoted as $W_{new} = \{w_{1,new}, w_{2,new}, \dots, w_{n,new}\}$ and calculate mean square error, denoted as $E_{W_{new}}$, between target PTSRs and estimated PTSRs using W_{new} .

vi. Update $W_{old} \leftarrow W_{new}$ if. $E_{W_{new}} < E_{W_{old}}$.

vii. Repeat the steps from ii to vi until $E_{W_{new}} \geq E_{W_{old}}$.

During the training process, it is challenging to determine an optimal learning ratio. If we choose too large value of η , it causes high convergence speed but it has high possibility of missing the optimal weight values. Too small value of η is vice versa of too large value of η where convergence speed is too small but low possibility of missing the optimal weight values. We determine the learning ratio from exhaustive empirical experiment in the next section.

3.3 Compensation of Measurement Errors

One thing not to be overlooked is that the link cost function can be distorted for the case when metrics such as RSS, HMRR, hop distance, and PTSR are corrupted by environmental factors during the training process. The quality of data delivery degrades as the errors in the link cost function with the corrupted metrics increases. For example, RSS corrupted with interference or obstacle will likely produce a high RSS value during training the link cost function. In this case, the RN associated with that RSS could be chosen as the next-hop RN which may not be the optimal solution. To handle the problem, we employ the compensation methods such as the weighted moving average and autoregressive (AR). The weighted moving average process is usually employed to filter out the RSS measurement uncertainty and obtains the low frequency components of the measured RSS as seen Eq. (6).

$$\bar{R}(n) = (1 - \alpha)\bar{R}(n-1) + \alpha R(n), \quad (6)$$

where $R(n)$ is the n^{th} measured RSS value, $\bar{R}(n)$ is the average RSS value after the n^{th} measurement, and α is the smoothing factor. The $\bar{R}(n)$ becomes smoother as α gets smaller. We filter out the measurement error by Eq. (6) with $\alpha = 0.01$ as in [38].

HMRR and PTSR depend on channel contention between nodes and fluctuate dynamically in time. We employ an adaptive autoregressive process as a statistical compensation method. The measured value in a time window can be represented by the AR(p) model. Let $z_t, z_{t-1}, z_{t-2}, \dots, z_{t-(M-1)}$ denote the measured value at time $t, t-T, t-2T, \dots, t-(M-1)T$, where T is the measurement interval and M is the number of measurements in a time window. By the AR(p), the current value can be expressed as a linear aggregate of the previous values and uncorrelated normal noise with mean zero and variance σ_a^2 . Let μ denote the average value in a time window. Defining $\tilde{z}_t = z_t - \mu$, we have

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \dots + \phi_p \tilde{z}_{t-p} + a_t, \quad (7)$$

where ϕ_i 's ($i=1, 2, \dots, p$) are model parameters. If we define an autoregressive operator of order

p by $\phi(B) = 1 - \phi_1 B - \phi_1 B^2 - \dots - \phi_p B^p$, where B is the backward shift operator defined by $Bz_t = z_{t-1}$ (hence, $B^m z_t = z_{t-m}$), the AR model is written economically as

$$\phi(B)\tilde{z}_t = a_t. \quad (8)$$

By Yule-Walker equations, ϕ_i s can be obtained in terms of the autocorrelations of \tilde{z}_t 's [39]. That is, if we write

$$\phi = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix} \quad \rho_p = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_p \end{pmatrix} \quad P_p = \begin{pmatrix} 1 & \rho_1 & \cdots & \rho_{p-1} \\ \rho_1 & 1 & \cdots & \rho_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p-1} & \rho_{p-2} & \cdots & 1 \end{pmatrix}, \quad (9)$$

where ρ_i is the large lag- i autocorrelation of \tilde{z}_t . The parameters ϕ can be obtained by solving the following linear operations

$$\phi = P_p^{-1} \rho_p. \quad (10)$$

From Eq. (7), the variance of a_t is given by

$$\sigma_a^2 = E[\tilde{z}_t^2] \left(1 - \sum_{i=1}^p \rho_i \phi_i\right), \quad (11)$$

where $E[\tilde{z}_t^2]$ is the expectation of \tilde{z}_t^2 . For example, in case of AR(1) process, the model parameter and the variance of a_t are given as $\phi_1 = \rho_1$, and $\sigma_a^2 = E[\tilde{z}_t^2](1 - \rho_1 \phi_1)$. By the time-series theory, in the sense of the minimum mean square error, the optimal k -step ahead predictor of \tilde{z}_{t+k} is given by the conditional expectation, $\hat{z}_{t+k} = E[\tilde{z}_{t+k} | \tilde{z}_t, \tilde{z}_{t-1}, \dots]$. Therefore, the k -step ahead prediction at time t is give by

$$\mu + \hat{z}_{t+k} + e_k, \quad (12)$$

where e_k is the k -step ahead prediction error. In this paper, we compensate the measurement errors in HMRR and PTSR using 1-step ahead prediction method.

4. Performance Evaluation

We perform experiments for our method using the NS-2 [40] simulator. We use the log-normal shadowing model [41] to model radio propagation environment. The path loss exponent ranges from 2 to 8 for a large-scale outdoor environment. An RN sends a HELLO message for every 100 milliseconds. A training packet to measure PTSR is sent every 100 milliseconds. IEEE 802.11 [42] is used as the MAC layer and the transmission range of a node is 250m. The total simulation time is 360 sec.

4.1 Validation of Input Types

Before we model the link cost function using three inputs as mentioned in the previous section, we validate whether RSS and HMRR are correlated, using the input sensitivity explained in the our previous work [33]. The validation algorithm is summarized as follows.

- i. Train an NN with the inputs, and calculate the cost function for each node.
- ii. Vary one input at one time by adding small random value ranging from 0.01 to 0.1.
- iii. Calculate the cost function using the varied input obtained in ii.
- iv. Obtain the input sensitivity of each input, which is the mean difference between the

costs obtained from i and iii.

As seen in **Table 1**, when we vary input x_1 (RSS), the input sensitivity of x_1 is 0.45 and that of x_2 (HMRR) is 0.36 while that of x_3 (hop distance) is 0.09. Also, when varying input x_2 , we obtain similar result with when varying input x_1 . However, when input x_3 is varied, only x_3 has large sensitivity compared to x_1 and x_2 . From the results, we validate x_1 and x_2 are correlated, and x_3 is uncorrelated. Based on the identified input types, we can model our link cost function as in Eq.(13).

$$Cost_i = \log(X_{i,c} \cdot W_C^T) + \log(w_3 \cdot x_{i,3}), \quad (13)$$

where $X_{i,c}$ is the coupled input vector collected from the i^{th} neighbor RN, composed of $[x_{i,1}, x_{i,2}]$ and W_c is the corresponding weight vector composed of $[w_1, w_2]$.

Table 1. The result of the input sensitivities.

Varied input	Input sensitivity of x_1	Input sensitivity of x_2	Input sensitivity of x_3
x_1	0.45	0.36	0.09
x_2	0.31	0.40	0.06
x_3	0.14	0.13	0.34

To determine the optimal learning ratio (η) as explained in the previous section, we train the NNs repeatedly as varying the value of η by 0.1 increment in a single-hop transmission environment. We compare the performances of our method with those of NN method (Fully Connected NN, FCNN). For the construction of the single-hop distribution network, 20 RNs are randomly placed in the 500m \times 500m area. From experimental results, we can infer that PTSR decreases as η increases. For $\eta = 0.3$, PTSRs of PCNN and FCNN are the best, i.e., 0.979 and 0.792 respectively. The training process takes about 5-10 seconds by varying the learning ratio. The training time includes the packet transmission delay and CPU processing time. As seen in Eq. (13), the link cost function operates in constant time $O(1)$ and repeats $n \times 1/\eta$ times. Thus, CPU processing time depends on the number of inputs, n . In this case n is only three, which is very small compared to other NN applications with tens or hundreds of inputs. The CPU processing time for training the link cost function is very small which is negligible in our application. Therefore, the convergence speed is not issued in this experiment. Hereafter, using $\eta = 0.3$, we compare the performance of the three methods by varying the input values.

4.2 Quality of Data Delivery

In this subsection, we evaluate the performance of our link cost function in terms of the following factors:

- **Data delivery ratio-** the ratio of the number of data packet successfully received at the GW to the total number of data packets sent by sinks.
- **End to end delay-** the time to deliver data from a sink to the GW.
- **Energy efficiency-** the number of packets delivered to the GW for each unit of energy spent by the network

Our experiment is performed for the distribution network with randomly deployed 120 RNs and 20 sinks in the 1000m \times 1000m area. The number of the mobile sinks is set to 20% of the number of sinks and the mobility model of the mobile sink is set to random waypoint [43]. The sink sends one data message every second at constant bit rate (CBR).

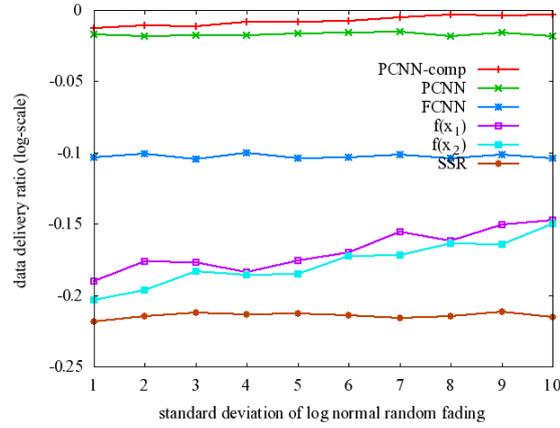


Fig. 4. Data delivery ratio with varying RSS by adding log normal random fading with $N(0, \sigma_1)$.

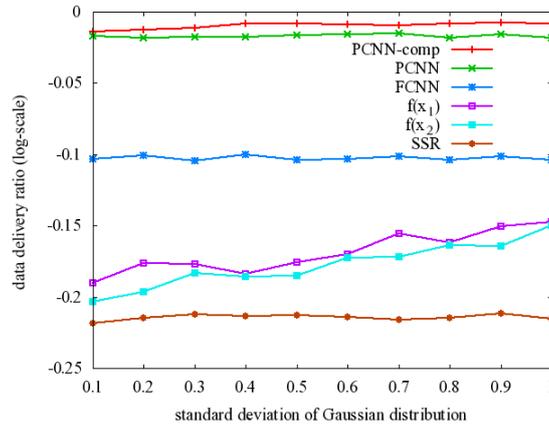


Fig. 5. Data delivery ratio with varying HMRR using Gaussian distribution with $N(0, \sigma_2)$.

Fig. 4 and **Fig. 5** show the data delivery ratio between a sink and the GW. The data delivery ratio is mostly affected by the signal strength and the degree of collision. In the figures, PCNN-comp and PCNN represent PCNN using and not using our compensation method for the measurement errors. SSR [21] is one of the conventional methods using x_3 . The results indicate that methods using the NNs deliver more data than the conventional methods by 47% and 42% as varying σ_1 and σ_2 . The link cost function using the PCNN improves the performance by about 21% and 23%, compared to that using the FCNN. It is because our link cost function considers various input metrics affecting the data delivery ratio and correlation between the inputs to obtain the cost function for determining the next-hop RN. In addition, the error compensation method improves the data delivery ratio of our link cost function by about 6%.

Fig. 6 and **Fig. 7** show the average delay of data delivery from a sink to the GW. The end to end delay is mainly affected by the link-layer retransmission of a data packet and the number of hops between a sink and the GW. As the probability of collision increases, the probability of retransmission increases. As seen in the figures, the link cost functions using the NNs show shorter end to end delay than the conventional method by about 38% and 33% respectively. It is because our link cost function considers HMRR reflecting the degree of collision in our link cost function for determining the next-hop RN. Besides, the

link cost function using the PCNN improves the delay by about 16% and 12%, compared to those using the FCNN with varying σ_1 and σ_2 . The error compensation method improves the end to end delay of our link cost function by about 4%.

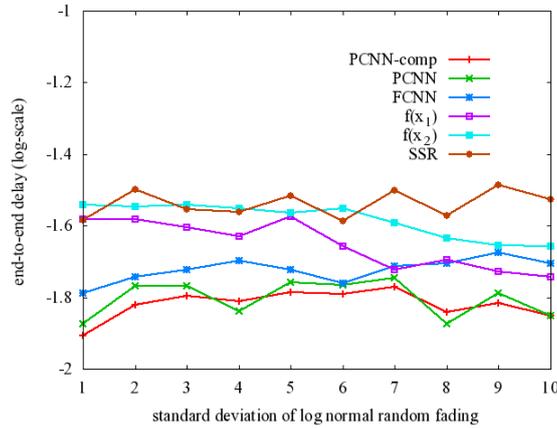


Fig. 6. End to end delay with varying RSS by adding log normal random fading with $N(0, \sigma_1)$.

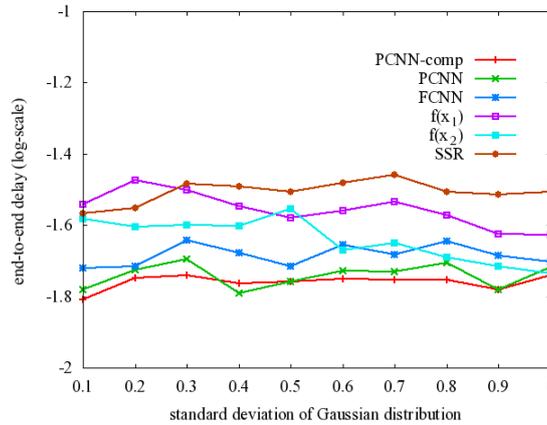


Fig. 7. End to end delay with varying HMRR using Gaussian distribution with $N(0, \sigma_2)$.

Energy efficiency(ξ) is calculated as seen in Eq. (14). The energy efficiency means the number of packets delivered to the GW for each unit of energy spent by the network in communication events.

$$\xi = \frac{P_{src} r}{e_{total} t}, \quad (14)$$

where r and t are delivery rate and total number of transmissions, respectively. e_{total} is the total amount of energy consumed by the network for each packet and it is obtained from $e_{total} = e_{tx} + e_{rx}$ where e_{tx} and e_{rx} are the amount of energy required by a node to transmit and to receive a packet, respectively. p_{src} is the number of packets sent by the data source. **Fig. 8** and **Fig. 9** show energy efficiencies in the environment varying RSS and HMRR. The link cost function using the PCNN improves the energy efficiency about 10% and 47%, compared to those using the FCNN and conventional methods. It is because our link cost function is

designed to improve the packet delivery ratio and it reduces the energy consumed per packet delivery.

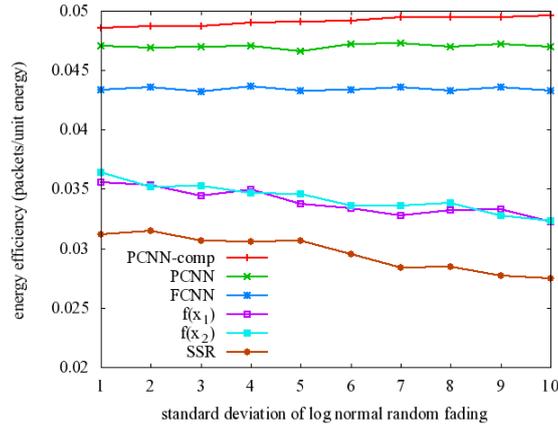


Fig. 8. Energy efficiency with varying RSS by adding log normal random fading with $N(0, \sigma_1)$.

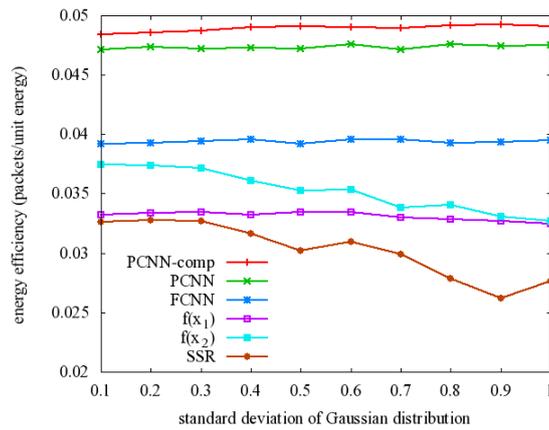


Fig. 9. Energy efficiency with varying HMRR using Gaussian distribution with $N(0, \sigma_2)$.

From the above results, we conclude that the data forwarding method using our link cost function improves the energy efficiency and quality of data delivery in terms of data delivery ratio and end to end delay. The distribution system adopting our method can deliver aggregated data from a sink to the GW reliably and promptly.

5. Conclusion

We developed the link cost function for the path management method in WSN. The contributions of this paper are stated as follows:

We designed the data forwarding method to deliver data from a sink to the GW.

We derived the cost function using the concept of PCNN structured by the input types (correlated input type and uncorrelated input type). The input types can be identified using the input sensitivity as shown in the experimental section.

We developed the training technique for finding optimal weights in the link cost function.

We showed the feasibility of our method from comparison of our method with the other conventional methods. From the comparison, we can conclude that the performance of our method is better than those of the conventional methods with perspective of the quality of data delivery and energy efficiency.

In our link cost function, we extracted three input features to customize for the distribution network in WSN, and improved the quality of data delivery and energy efficiency with the inputs.

There are two approaches for efficient data delivery; one is power control and the other is the link cost function. In this paper, we focus on the development of link cost function to solve the data delivery problem. Since the power control approach can be one of solutions to solve the problem, we will consider the approach as another research direction. Also, we need to extract more input features to meet the requirements and characteristics of applications and systems such as ITS (Intelligent Transportation System) and wireless mesh network.

References

- [1] D. Culler, D. Estrin and M. Srivastava, "Overview of sensor networks," *IEEE Computer*, vol.37, no.8, pp. 41-49, Aug. 2004. [Article \(CrossRef Link\)](#).
- [2] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol.11, no.6, pp. 54-61, Dec. 2004. [Article \(CrossRef Link\)](#).
- [3] 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, Dec. 2004. [Article \(CrossRef Link\)](#).
- [4] D. Niculescu, "Communication paradigms for sensor networks," *IEEE Communications Magazine*, vol.43, no.3, pp. 116-122, March 2005. [Article \(CrossRef Link\)](#).
- [5] Y. Sankarasubramaniam, O.B. Akan, and I.F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," in *Proc. of ACM Annual Int. Conf. on Mobile Computing and Networking (MOBICOM)*, Annapolis, pp. 177-188, June 2003. [Article \(CrossRef Link\)](#).
- [6] H. Chen, C.K. Tse and J. Feng, "Minimizing effective energy consumption in multi-cluster sensor networks for source extraction," *IEEE Trans. on Wireless Communications*, vol.8, no.3, pp. 1480-1489, March 2009. [Article \(CrossRef Link\)](#).
- [7] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks," *IEEE Trans. on Networking*, vol.16, no.1, pp. 51-62, Feb. 2008. [Article \(CrossRef Link\)](#).
- [8] S. Haykin, "Neural Networks: A Comprehensive Foundation," Prentice-Hall, 1998. [Article \(CrossRef Link\)](#).
- [9] M. Chen, S. Gonzalez, and V.C.M. Leung, "Applications and design issues for mobile agents in wireless sensor networks," *IEEE Wireless Communications*, vol.14, no.6, pp. 20-26, Dec. 2007. [Article \(CrossRef Link\)](#).
- [10] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proc. of ACM Annual Int. Conf. on Mobile Computing and Networking (MOBICOM)*, Atlanta, Sep. 2002, pp. 148-159. [Article \(CrossRef Link\)](#).
- [11] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Trans. on Wireless Communications*, vol.4, no.5, pp. 2579-2590, Sept. 2005. [Article \(CrossRef Link\)](#).
- [12] E. L. Lloyd and X. Guoliang, "Relay node placement in wireless sensor networks," *IEEE Trans. on Computers*, vol.56, no.1, pp. 134-138, Jan. 2007. [Article \(CrossRef Link\)](#).
- [13] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "High-throughput path metric for multi-hop wireless routing," *ACM Wireless Networks*, vol.11, no.4, pp. 419-434, July 2005. [Article \(CrossRef Link\)](#).

- [14] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. of ACM Annual Int. Conf. on Mobile Computing and Networking (MOBICOM)*, Philadelphia, pp. 114-128, Sept. 2004. [Article \(CrossRef Link\)](#).
- [15] M. Z. Zamalloa, K. Seada, B. Krishnamchari, and A. Helmy, "Efficient geographic routing over lossy links in wireless sensor networks," *ACM Trans. on Sensor Networks*, vol. 4, no. 3, pp. 12:1-12:33, May 2008. [Article \(CrossRef Link\)](#).
- [16] S. Lindsey and C. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proc. of IEEE Aerospace Conf.*, pp. 1125-1130, March 2002. [Article \(CrossRef Link\)](#).
- [17] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, vol.17, no.8, pp. 1333-1344, Aug. 1999. [Article \(CrossRef Link\)](#).
- [18] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. of IEEE Wireless Communications & Networking Conf. (WCNC)*, Orlando, pp. 350-355, March 2002. [Article \(CrossRef Link\)](#).
- [19] M. Younis, M. Youssef and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proc. of IEEE/ACM Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Fort Worth, pp. 129-136, Oct. 2002. [Article \(CrossRef Link\)](#).
- [20] M. Youssef, M. Younis and K. Arisha, "A constrained shortest-path energy-aware routing algorithm for wireless sensor networks," in *Proc. of IEEE Wireless Communications & Networking Conf. (WCNC)*, Orlando, pp. 794-799, March 2002. [Article \(CrossRef Link\)](#).
- [21] B.K. Szymanski and G.G. Chen, "Computing with time: from neural networks to sensor networks," *The Computer Journal, The British Computer Society*, vol.51, no.4, pp. 511-522, July 2008. [Article \(CrossRef Link\)](#).
- [22] M. Mangeas, A.S. Weigend and C. Muller, "Forecasting electricity demand using nonlinear mixture of experts," *World Congress on Neural Networks*, vol. 2, pp. 48-53, 1995. [Article \(CrossRef Link\)](#).
- [23] D.M. Bates., "Nonlinear Regression Analysis and Its Applications," Wiley, 1998. [Article \(CrossRef Link\)](#).
- [24] D. Cohen, J. Shawe-Taylor, "Feed-forward networks-a tutorial," in *Proc. of a Meeting on Neural Computing*, pp.1-13, 1989. [Article \(CrossRef Link\)](#).
- [25] L. Mussone, "A review of feed-forward neural networks in transportation research," *Elektrotechnik und Informationstechnik*, vol. 116, no. 6, pp. 360- 365, 1999. [Article \(CrossRef Link\)](#).
- [26] H. Mori, K. Itou, H. Uematsu and S. Tsuzuki, "An artificial neural net based method for predicting power system voltage harmonics," *IEEE Transactions on Power Delivery*, vol. 7, no. 1, pp. 402-409, 1992. [Article \(CrossRef Link\)](#).
- [27] D.E. Duckro, D.W. Quinn and S.J. Gardner, "Neural network pruning with Tuckey Kramer multiple comparison procedure," *Neural Computation*, vol. 14, no. 5, pp. 1149-1168, 2002. [Article \(CrossRef Link\)](#).
- [28] J.L. Noyes, "B3.5 Handbook of Neural Computation," 1997. [Article \(CrossRef Link\)](#).
- [29] Y.Y. Chung, M.T. Wong and N.W. Bergmann, "High-speed neural network based classifier for real-time application," in *Proc. of Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on*, pp. 506-509, 1998. [Article \(CrossRef Link\)](#).
- [30] A. Rajapakse, K. Furuta and S. Kondo, "Identification of nonlinear dynamic models with partially connected neural networks trained using orthogonal least square estimation," *Transactions of the Institute of Electrical Engineerings of Japan, Part C*, vol. 3, pp. 335-343, 1999. [Article \(CrossRef Link\)](#).
- [31] I.K. Sethi, "Entropy nets: from decision trees to neural networks," in *Proc. of the IEEE*, vol. 78, pp. 1605-1613, 1990. [Article \(CrossRef Link\)](#).
- [32] J.R. Quinlan, "C4.5 Programs for machine learning," Morgan Kaufmann Publisher, 1993. [Article \(CrossRef Link\)](#).

- [33] S. Kang and C. Isik, "Partially connected feedforward neural networks structured by input types," *IEEE Trans. on Neural Networks*, vol.16, no.1, pp.175-184, Jan. 2005. [Article \(CrossRef Link\)](#).
- [34] J.M. Zurada, A. Malinowski, U. Shiro, "Perturbation method for deleting redundant input of perceptron networks," *Neurocomputing*, vol. 14, no. 2, pp. 177-193, 1996. [Article \(CrossRef Link\)](#).
- [35] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. on Wireless Communications*, vol.2, no.4, pp. 736-748, July 2003. [Article \(CrossRef Link\)](#).
- [36] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha and S. Banerjee, "Diagnosing wireless packet losses in 802.11: separating collision from weak signal," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM)*, Phoenix, pp. 735-743, April 2008. [Article \(CrossRef Link\)](#).
- [37] K. Whitehouse, C. Karlof and D. Culler, "Practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41-52, 2007. [Article \(CrossRef Link\)](#).
- [38] S. Woon, N. Golmie and Y. A. Sekercioglu, "Effective link triggers to improve handover performance," in *Proc. of IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Finland, pp. 1-5, Sep. 2006. [Article \(CrossRef Link\)](#).
- [39] G. E. P. Box, G. M. Jenkins and G. C. Reinsel, "Time series analysis: Forecasting and control," Holden-Day, 1976. [Article \(CrossRef Link\)](#).
- [40] The Network Simulator, ns-2. [Article \(CrossRef Link\)](#).
- [41] T. S. Rappaport, "Wireless Communications, principles and practice," Prentice Hall, 1996. [Article \(CrossRef Link\)](#)
- [42] IEEE Std. 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification: higher speed physical layer (PHY) extension in the 2.4GHz Band," 1999. [Article \(CrossRef Link\)](#)
- [43] G. Resta, and P. Santi, "An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks," in *Proc. of ACM Workshop On Principles Of Mobile Computing (POMC)*, France, pp. 44-50, Oct. 2002. [Article \(CrossRef Link\)](#).



Yujin Lim received a B.S. and a M.S. degree in computer science, and a Ph.D. degree in computer science from Sookmyung Women's University, Seoul, Korea in 1995, 1997, and 2000 respectively. From 2000 to 2002 she worked as a research faculty at the department of mechanical and information engineering in the University of Seoul, Seoul, Korea. She worked as a research staff at the department of computer science in the University of California Los Angeles from 2002 to 2003. She worked for Samsung Advanced Institute of Technology as a senior research engineer from 2003 to 2004. Since 2004, she is currently an assistant professor in department of information media, University of Suwon. Her current research interests include ad hoc and sensor networks, mesh networks, vehicular ad hoc network, and routing protocols over wireless environments.



Sanggil Kang received the M.S. and Ph.D. degrees in Electrical Engineering from Columbia University and Syracuse University, USA in 1995 and 2002, respectively. He is currently an associate professor in Computer Science at INHA University, Korea. His research interests include Semantic Web, Artificial Intelligence, Multimedia Systems, Inference Systems, etc.