Delay-Constrained Energy-Efficient Cluster-based Multi-Hop Routing in Wireless Sensor Networks

Trong-Thua Huynh, Anh-Vu Dinh-Duc, and Cong-Hung Tran

Abstract: **Energy efficiency is the main objective in the design of a wireless sensor network (WSN). In many applications, sensing data must be transmitted from sources to a sink in a timely manner. This paper describes an investigation of the trade-off between two objectives in WSN design: minimizing energy consumption and minimizing end-to-end delay. We first propose a new distributed clustering approach to determining the best clusterhead for each cluster by considering both energy consumption and end-to-end delay requirements. Next, we propose a new energy-cost function and a new end-to-end delay function for use in an inter-cluster routing algorithm. We present a multi-hop routing algorithm for use in disseminating sensing data from clusterheads to a sink at the minimum energy cost subject to an end-to-end delay constraint. The results of a simulation are consistent with our theoretical analysis results and show that our proposed performs much better than similar protocols in terms of energy consumption and end-to-end delay.**

Index Terms: **Cluster, end-to-end delay, energy consumption, multi-hop, trade-off.**

I. INTRODUCTION

ENERGY is the most crucial resource for wireless sensors, particularly in environments in which replacing or recharging a sensor's batteries is impossible. Therefore, energy efficient routing protocol is the main objective for wireless sensor networks (WSNs). However, in many current applications of WSNs, such as forest fire detection, data must be transmitted from sources to a sink within a limited amount of time for the data be useful. Thus, a trade-off exists between minimizing energy consumption and minimizing end-to-end delay.

Although many heuristic solutions to balancing delay and energy consumption in WSNs have been presented, their effectiveness is negligible because of their long convergence times [1]–[5]. Clustering is a technique that has been used very effectively to archive energy efficiency in WSNs [6]. In clustering, sensors select themselves as clusterheads based on probability values. Because of energy constraints, a sensor in a WSN can only communicate directly with other sensors that are within a small distance. To enable communication between sensors that are not within each other's communication range, the sensors form a multi-hop communication network. In the clustering approach, each cluster has a clusterhead that combines all of the sensing data from its members and forwards it to the sink. When the clusterhead and sink are far from each other, the direct communication between the clusterhead and sink increases the energy consumption of the clusterhead exponentially with distance [7].

Direct communication minimumizes delay but increases energy consumption. Multi-hop communication is energy efficient but increases delay [8]. In this paper, we present a new approach, called delay-constrained energy multi-hop (DCEM) for solving the aforesaid problem by considering the delay-energy trade-off in multi-hop routing between clusterheads.

The major contributions of this research are the following:

- We propose a clusterhead selection approach for each cluster to optimize two objectives: Minimization of energy consumption and minimization of end-to-end delay.
- We propose a new energy-cost function and a new end-to-end delay function for use in determining the lowest-cost route for data dissemination from clusterheads to a sink, subject to an end-to-end delay.
- We present an inter-cluster multi-hop routing algorithm that takes into consideration both energy consumption and end-to-end delay.
- We present the results of a simulation conducted to assess the performance of our protocol and a comparison of the results with those of conventional protocols. Performance was assessed in terms of the ability to determine the optimal hop-count value to achieve the best trade-off between minimizing energy consumption and minimizing end-to-end delay for a specific network size.

The remainder of the paper is organized as follows. In Section II, we describe proposed solutions to this problem and place our work in this context. In Section III, we present network and energy models. And in Section IV, we present details of the DCEM approach. The results of a simulation conducted to confirm the correctness of our theoretical analysis and a comparison with similar protocols are presented in Section V. Section VI concludes the paper.

II. RELATED WORKS

Several studies have been conducted to attempt, with varying degrees of success, to address the problem of energy-efficient delay-constrained routing in WSNs.

Clu-DDAS [9], proposed by Li *et al.*, is an energy-efficient distributed scheduling algorithm based on a cluster-based aggregation tree. The authors studied the minimum-latency aggregation schedule problem and proposed a collision-free transmission schedule for data aggregation for all sensors such that the delay for aggregated data to reach the sink is minimized. By con-

Manuscript received December 13, 2015.

T.-T. Huynh is with Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam, email: htthua@ptithcm.edu.vn.

A.-V. Dinh-Duc is with University of Information Technology, Vietnam National University, Ho Chi Minh City, email:vudda@uit.edu.vn.

C.-H. Tran is with Posts and Telecommunications Institute of Technology, in Ho Chi Minh City, Vietnam, email: conghung@ptithcm.edu.vn.

Digital object identifier 10.1109/JCN.2016.000081

structing a cluster-based data aggregation tree, this protocol permits concurrent and collision-free packet transmissions among different clusters. However, constructing distributed trees using a broadcasting technique generates more overhead.

Huynh *et al.* proposed the Energy*Delay multi-hop routing scheme to balance energy efficiency and network delay [10]. This routing algorithm is applied within a three-hop cluster for sensors within each cluster, while an energy-efficient construction algorithm is applied for clusterheads to construct energy-efficient chains from clusterheads to the sink. However, this algorithm is not sufficiently flexible for fixed three-hop clusters. These authors have also proposed another energy-efficient delay-aware routing algorithm for a multi-layer WSN [11], in which clusterheads at each layer are interconnected as in a de Bruijn graph model to reduce network delay and energy consumption, and increase system reliability. The performance of the algorithm in terms of delay and energy consumption was demonstrated experimentally.

In hybrid energy-efficient distributed clustering (HEED) [12], clusterheads are chosen periodically, based on a hybrid of the nodal remaining energy and a secondary parameter, such as nodal proximity to its neighbors or nodal degree. HEED can achieve a uniform clusterhead distribution across the whole network, but it must perform many iterations to accomplish this and therefore incurs high overhead.

Delay-bounded adaptive energy-constrained routing (DEAR) [13] is a multi-path routing protocol that considers in many parameters, such as reliability, delay, and energy consumption. This protocol allows packets to be continuously distributed across the network, even if the paths are going to crash. It balances the delay between the different paths by providing a polynomial-time algorithm for solving the multi-objective optimization problem. However, the energy savings and network delay efficiency achieved is limited because of the complexity of the algorithm.

In [14], the authors analyzed the trade-off between delay and energy consumption in data aggregation. They showed that a WSN suffers from high energy consumption without the use of a data aggregation method and suffers from high delay when a full aggregation method is used. In [15], the authors proposed a delay-energy aware routing protocol (DEAP) for heterogeneous sensor and actor networks. Energy saving is achieved by using the resources of actor nodes whenever possible. This involves using an adaptive energy management scheme to control the wake-up cycle of the sensor nodes, based on the delay experienced by the packets, and using geographical information for load balancing to achieve energy efficiency.

In [16], the authors analyzed the energy-delay trade-off during the deployment of a WSN. They proposed a formal model for use in comparing the performance of the different protocols and algorithms. In [17], the authors divided energy-efficient routing into two subproblems. The first is how to construct efficient routing trees. The second is how to assign wake-up frequencies with multiple routing trees. The authors obtained a solution to the first problem using an optimization algorithm. In addition, they proved that the second problem was nondeterministic polynomial-time hard (NP-hard) and presented a polynomial-time approximation algorithm to solve it.

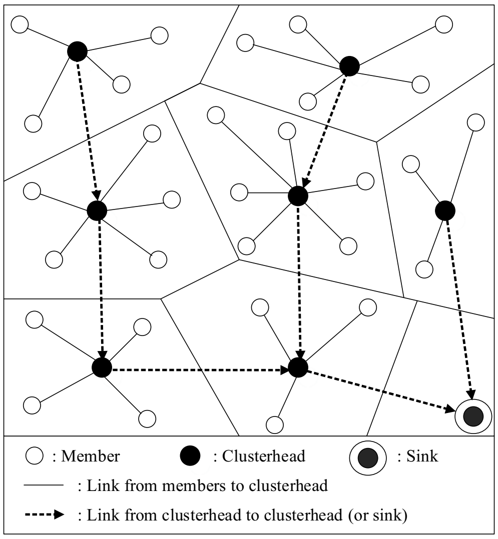


Fig. 1. Hierarchical wireless sensor network model.

In [8], the authors proposed data forwarding protocols for trade-off energy and delay that involve slicing the communication range of sensors into concentric circles. In [18], the authors proposed an energy-delay trade-off solution for intra-cluster routing in a WSN.

Akkaya and Younis [19] proposed a routing protocol that finds an energy-efficient path along which the end-to-end delay requirements of the connection are met. They assumed that the sensor nodes have a class-based, priority queuing mechanism. This mechanism can convert the delay requirements into bandwidth requirements. This approach, however, does not take into consideration the other delays that can occur due to channel contention at the medium access control (MAC) layer.

III. NETWORK AND ENERGY MODEL

A. Network Model

Consider a set of sensors dispersed in a field. We employ the hierarchical network model shown in Fig. 1 and make the following assumptions:

- All sensors are stationary, have similar capabilities, and have equal significance.
- All sensors are aware of their own residual energy and adapt their transmission power according to communication distances.
- Links are symmetric, and the radio signal has identical energy attenuation in all directions.
- Data exchanged between two communicating sensors that are not within each other's radio range are forwarded by other sensors.
- All sensors are capable of operating in forwarding (clusterhead) mode and sensing mode.
- The data sensed by adjacent nodes are correlative, so the clusterhead can combine the collective data to reduce the total data sent.

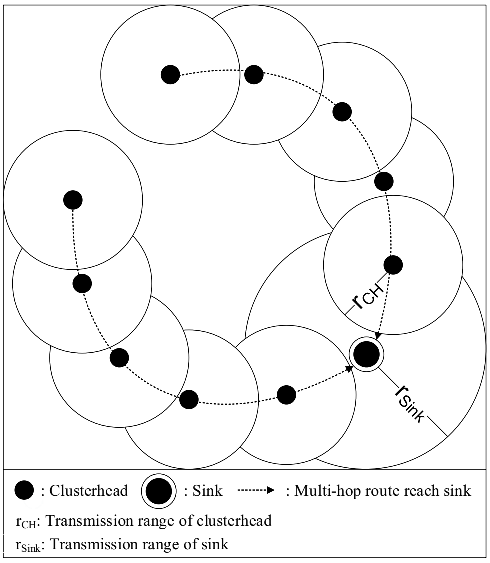


Fig. 2. Clusterheads can adjust their radii to communicate with both members and other clusterheads (or sink) in the multi-hop route to reach the sink.

In this hierarchical network model, sensor nodes are distributed in clusters. Each cluster selects a clusterhead that aggregates data from its members and sends the combined data to the sink in a multi-hop manner. The clusterheads also act as relays that forward packets to the sink from the other clusterheads. In addition, the sensor nodes (especially the clusterheads) are capable of adjusting their radii to reach adjacent nodes in the process of disseminating data to the sink, as shown in Fig. 2.

B. Energy Model

We use a simplified model for the radio hardware energy dissipation in [7]. To receive *l*-bit data, the energy spent for the radio is as follows:

$$E_{\text{Rx}}(l) = E_{\text{elec}} \times l \tag{1}$$

where E_{elec} is the electronic energy consumption factor.

It is assumed that the sensed data are correlated; thus, a clusterhead can combine the data gathered from its members into a single fixed-length packet. The clusterhead fuses *l*-bit data from *m* members to expend:

$$E_{\text{Fu}}(l,m) = m \times E_{\text{fuse}} \times l \tag{2}$$

where E_{fuse} is the data fusion factor.

The radio hardware energy consumption in transmitting *l*-bit data over a distance *d* is as follows:

$$E_{\text{Tx}}(l,d) = \begin{cases} l \times E_{\text{elec}} + l \times \epsilon_{fs} \times d^2, & \text{if } d < d_0 \\ l \times E_{\text{elec}} + l \times \epsilon_{mp} \times d^4, & \text{if } d \geq d_0 \end{cases} \tag{3}$$

where E_{elec} is the electronic energy consumption factor, ϵ_{fs} and ϵ_{mp} are the amplifiers required to maintain an acceptable signal-to-noise ratio, and $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ is the reference distance between transmitter and receiver.

IV. DCEM DETAILS

DCEM is a distributed clustering scheme that operates in consecutive rounds, each round of which is separated into two phases: Network organization and data transmission. The first stage's task is to establish a cluster network topology and build a multi-hop route. The second stage's task is to transmit data from source sensors to the sink via clusterhead-based multi-hop forwarding.

A. Network Organization

A.1 Cluster Setup

The algorithm begins with the neighbor discovery phase, which is initiated by the sink by broadcasting an advertisement (ADV) message to all nodes at a certain power level. Each node computes its approximate distance to the sink (d_{toSink}) according to the received signal strength.

Each node waits for an amount of time $\tau = 1/E$ before broadcasting an ADV(ID,E) message to its neighbors and collecting data from the neighbors, where *ID* is a nodal identifier and *E* is the nodal remaining energy. Each node compares its energy level with the energy level of the nodes from which it has received ADV messages. If a sensor node has less energy, it will cancel its timer and decide to be a cluster member (i.e., a non-clusterhead).

The clusterhead candidates are the set of sensor nodes that send ADV messages and then either do not receive any ADV messages or have higher energy than the energy in the ADV messages they receive. It is possible for two nodes with the same energy level to be in communication range of each other. To address this situation, a trade-off for energy and delay (TED) is used to establish a balance between energy consumption and end-to-end delay by adjusting the value of the parameter α based on the remaining energy of the clusterhead and the value of the parameter β based on distance from the clusterhead to the sink. The TED is calculated for sensor *i* from (4) for the clusterhead candidates only. α and β are controlling parameters. α is used to adjust the dependence of the remaining energy of the clusterhead candidates, and β is used to adjust the distance between the clusterhead candidates and the sink. The values of α and β lie in the range of [0, 1] and $\alpha + \beta \neq 0$.

$$TED_i = \left(\frac{E_i}{E_{\text{total}}}\right)^{\alpha} + \left(\frac{1}{d_{(i,s)}}\right)^{\beta}. \tag{4}$$

In (4), E_i is the remaining energy of clusterhead candidate *i*, E_{total} is the cumulative energy of the other clusterhead candidates received from ADV messages, and $d_{(i,s)}$ is the distance from clusterhead candidate *i* to the sink.

Each clusterhead candidate *i* waits for an amount of time $\omega = 1/TED_i$ before making an announcement that it is a final clusterhead. All clusterhead candidates that receive a final clusterhead announcement cancel their TED timers to become the member nodes for the current round. After the cluster setup procedure is finished, all clusterheads broadcast time division multiple access (TDMA) message to allocate time slots to their cluster members.

A.2 *Calculating the End-to-End Delay*

The link delay $D(i,j)$ is a measure of the delay a packet experiences when traversing a link from node *i* to node *j*. By definition, a link delay $D(i,j)$ includes the queuing delay d_Q per node, the transmission delay d_T, and the propagation delay d_P. In other words:

$$D(i,j) = (d_Q + d_T + d_P) \tag{5}$$

where $d_T = l/\psi$ and $d_P = d_{ij}/\gamma$; *l* is the packet size (bits), ψ is the link bandwidth (bps), d_{ij} is the length of physical link from clusterhead *i* to clusterhead *j*, and γ is the propagation speed in the medium (m/s). The value of d_Q can be calculated using rules related to queue theory. The nodal queue is considered to be of type M/M/1 [20]. In this type of queue, the input is of Poisson type, the output is an exponential random variable, and the amount of service is *1*. The queuing delay d_Q in this queue is calculated based on the following equation:

$$d_Q = \frac{1}{\mu - \lambda} \tag{6}$$

where μ is the service rate, which is an exponential stochastic variable, and λ is the rate of entry for new packets, which is a Poisson stochastic variable.

An end-to-end delay, denoted by $D_{\text{ete}}(x,s)$, is the time elapsed between the departure of a collected data packet from a source *x* and its arrival at the sink *s*. By definition, the end-to-end delay $D_{\text{ete}}(x,s)$ of the route from clusterhead *x* to sink *s* is defined as:

$$\begin{aligned} D_{\text{ete}}(x,s) &= \sum_{i,j\in\{x,U,s\}} D(i,j) \\ &= \sum_{i,j\in\{x,U,s\}} \left((\frac{1}{\mu-\lambda}) + \frac{l}{\psi} + \frac{d_{ij}}{\gamma} \right) \end{aligned} \tag{7}$$

where μ, λ, ψ, and γ are constants that are assumed to be the same for all clusterheads; *l* is the packet size (bits); ψ is the link bandwidth (bps); d_{ij} is the length of the physical link from clusterhead *i* to clusterhead *j*; γ is the propagation speed in the medium (m/s); and *U* is the set of intermediate nodes from clusterhead *x* to sink *s*.

A.3 *Calculating the Link and Route Costs*

We define the following cost function for a link between clusterhead nodes *i* and *j*.

$$\begin{aligned} cost_{ij} &= \sum_{\Theta\in\{Rx,Fu,Tx\}} E^{ij}_{\Theta} + \rho \times cost(E^i_{Re}) \\ &= (E^i_{Rx} + E^i_{Fu} + E^{ij}_{Tx}) + \rho \times cost(E^i_{Re}) \end{aligned} \tag{8}$$

where E^i_{Rx}, given by (1), is the energy that clusterhead *i* spends receiving data from members; E^i_{Fu}, given by (2), is the energy that clusterhead *i* spends in fusing data from *m* members; E^{ij}_{Tx}, given by (3), is the energy spends transmitting data from clusterhead *i* to clusterhead *j*; and ρ is the nodal remaining energy factor.

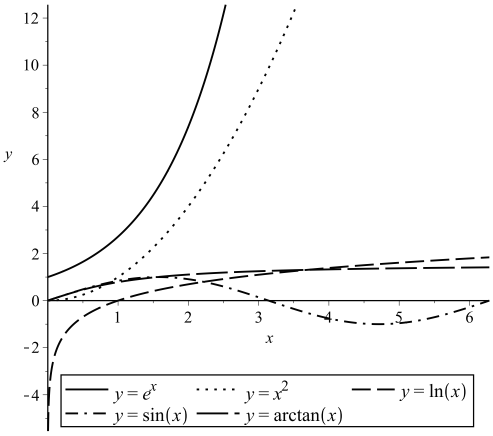


Fig. 3. Variation of the elementary functions.

The $cost(E^i_{Re})$ is cost function that takes into consideration the remaining energy of sensors for the energy balance among sensors. Therefore, the function $cost(E^i_{Re})$ is based on the principle in which small changes in remaining energy of sensors can result in large changes in value of cost function. Exponential function $f(x) = exp^{\frac{1}{x^2}}$ is the type of function that can satisfy this principle [21]. Replacing *x* by E^i_{Re}(the remaining energy of sensor *i*), we have the following cost function:

$$cost(E^i_{Re}) = exp^{\left(\frac{1}{(E^i_{Re})^2}\right)}. \tag{9}$$

The following illustrates why the function $f(x) = e^{\frac{1}{x^2}}$ is chosen to balance the energy consumption among sensor nodes and maximizes network lifetime.

According to [22], among the elementary functions such as x^{α}, e^x, *ln(x)*, *sin(x)*, *arctan(x)*,$\cdots$, the function e^x is the sharpest fluctuating function when *x* changes in a small interval as illustrated in Fig. 3. Moreover, according to the aforementioned principle, we need to find a function *f(x)* that satisfies two conditions as follows:

(i) When *x* is decreasing to 0 then *f(x)* is increasing to $+\infty$.
(ii) The function *f(x)* is the sharpest increasing function when *x* is decreasing to 0.

Therefore, we chose the function $f(x) = e^{\frac{1}{g(x)}}$, where the function $g(x) = x^{\alpha}$ is decreasing sharply to 0 when *x* is decreasing to 0. That is, the second condition *(ii)* is satisfied. Fig. 4 illustrates the fluctuation of the function $f(x) = e^{\frac{1}{x^{\alpha}}}$ comparing with that of the function $f(x) = e^{\frac{1}{sin(x)}}$, where α = 2 is preferred to the larger values for reducing computation time in each sensor node. As shown in Fig. 4, the function $f(x) = e^{\frac{1}{x^{\alpha}}}$ is fluctuating sharper than the function $f(x) = e^{\frac{1}{sin(x)}}$ especially with *x* in range of [0, 1].

To calculating the cost function for a route from clusterhead node *x* to the sink *s*, we define the following equation:

$$Cost(x,s) = \sum_{i,j\in\{x,U,s\}} cost_{ij} \tag{10}$$

where U is set of intermediate nodes from clusterhead x to sink *s*.

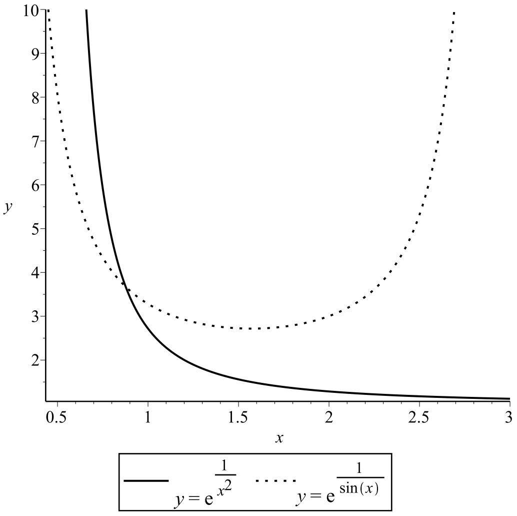


Fig. 4. Variability of the power function x^α and the trigonometric function *sin(x)* in combination with the exponential function.

A.4 *Inter-Cluster Multi-Hop Routing Algorithm*

Our optimization problem is finding the lowest cost route (most energy efficient) from a clusterhead node *x* to the sink *s* such that the end-to-end delay along that route does not exceed a delay constraint Δ. The constrained minimization problem is:

$$\min_{R_k \in R'(x,s)} Cost(R_k) \tag{11}$$

where R_k is the kth route, $R^{'}(x,s)$ is the set of routes from clusterhead node *x* to the sink *s* for which the end-to-end delay is bounded by Δ, given by:

$$D_{\text{ete}}(R_k) \leq \Delta, R_k \in R^{'}(x,s). \tag{12}$$

By considering the optimization problem above, we propose the algorithm shown in Algorithm 1 to find *k*-least cost routes that meet the end-to-end delay constraint.

The algorithm calculates the $cost_{ij}$ (line 3) for each link from clusterhead *i* to clusterhead or sink *j* based on the cost function defined in (8). Then, it calculates the number of probable routes from clusterhead node *x* to the sink *s* (line 4) using depth-first search (DFS) algorithm [23]. In line 5, the algorithm uses the *k*-shortest path [24] to find *k*-least cost route based on (8), (9), and (10). After determining the least-cost route R_k (initial *k*=1), the algorithm calculates the end-to-end delay $D_{\text{ete}}(R_k)$ for that route using (7). Then, it checks whether this end-to-end delay can satisfy specified threshold value Δ or not. If so, R_k is chosen (*SeR*, lines 9 and 10), and if not, R_k will be removed and added to the *NoSa* (lines 7 and 13). Line 7 will remove least-cost routes that do not satisfy the delay bound Δ.

Algorithm 1 *Algorithm for finding k-least cost routes that meet the end-to-end delay constraint*

```
1:  $SeR = \oslash$;   ▷ SeR is the selected route to disseminate data from clusterhead x to the sink s.
2:  $NoSa = \oslash$;  ▷ NoSa is set of routes that does not satisfy the delay bound $\Delta$.
3:  Calculate $cost_{ij}, \forall i, j \in C$;  ▷ C is set of clusterhead nodes, j can be sink.
4:  Calculate K(x,s);   ▷ K(x,s) is number of probable routes from clusterhead node x to the sink s.
5:  Find k-least cost routes k-SR(x,s,k);  ▷ k-SR(x,s,k) are k least cost routes from clusterhead x to sink s.
6:  while ($k \neq K(x,s)$) do   ▷ initial k =1.
7:      $R_k$ = k-SR(x,s,k) $\setminus NoSa$;   ▷ $R_k$ is the $k$th least-cost route.
8:      Calculate $D_{\text{ete}}(R_k)$ from (7);
9:      if $D_{\text{ete}}(R_k) \leq \Delta$ then
10:         $SeR = R_k$;
11:         break;
12:     else
13:         $NoSa = NoSa \cup R_k$;
14:         $k = k + 1$;
15:     end if
16: end while
17: Return SeR;
```

A.5 *Convergence and Complexity of Algorithm*

We verify the convergence of the algorithm provided that it always finishes within a finite time and the computational complexity is a polynomial function.

Theorem 1: *If* $\exists$ *K(x,s) routes from clusterhead x to sink s,* $\forall$ *1* $\leq$ *k* $\leq$ *K(x,s), the Algorithm 1 either finds k-least cost routes that meet the end-to-end delay constraint or no routes within a finite time.*

Proof: If no routes from clusterhead to the sink exist, the algorithm stops immediately after line 5. If so, *k-SR(x,s,k)* is found by K-shortest path algorithm as proved in [24]. Then, $\forall$ 1 $\leq$ *k* $\leq$ *K(x,s)*, if $\exists$ R_k | $D_{ete}(R_k) \leq \Delta$, the algorithm will stop with *SeR*=R_k (line 9) that satisfies the delay requirement. If no, it stops and there is no route exist that meets the end-to-end delay constraint (*k = K(x,s), SeR* = $\oslash$). That means the data will not be disseminated to the sink thereafter. □

Theorem 2: *The execution time of the algorithm for finding the route between a given clusterhead x and the sink s is O(n).*

Proof: The DFS algorithm [23] has proved that its computational complexity is *O(N)* where *N* is the number of nodes. In line 6, the **While** loop has the complexity *O(cK)* $\approx$ *O(K)*, where *K* is the number of clusterheads (*K* $\ll$ *N*). Clearly, at line 5, the computational complexity of *Cost(x,s)* given by (8), (9), and (10) is fixed by *O(1)* because it is performed in a finite time. Similarly, at line 8, the computational complexity of $D_{ete}(x,s)$ given by (7) is also fixed by *O(1)*. Furthermore, the set of steps in the algorithm 1 is organized in the sequence (non-nested) form, and the complexity of the algorithm 1 is *O(N)* + *O(K)* $\times$ *O(1)* $\approx$ *O(n)*. As a result, the computational complexity of the algorithm 1 is a polynomial function. This is fully suited to implementing for a distributed algorithm with a finite number of sensor nodes *n*. □

B. *Data Transmission*

Once the inter-cluster multi-hop routing is created, data transmission begins. Each member turns off the radio until it is allocated transmission time, and then sends the sensing data to the clusterhead during its time. The clusterhead keeps its receiver on to receive the data from the nodes in the cluster. After all the data has been received, the clusterhead fuses all data into a single packet to reduce redundancy and transmission energy, and then sends data to the other clusterhead which forwards the received packet so that it reaches the sink. After a certain time, the next round begins with network setup phase again.

Because members only need to send the sensing data to the clusterhead, the energy consumption of each member *j* is:

$$E_{\text{mem}}(j) = l \times E_{\text{elec}} + l \times \epsilon_{fs} \times d^2(j), \tag{13}$$

where $d(j)$ is distance from member *j* to its clusterhead.

Because the clusterhead needs to fuse all intra-cluster data from its members and forward the fused data to other clusterheads, its energy consumption is:

$$E_{\text{CH}}(i) = E_R(i) + E_F(i) + E_S(i), \tag{14}$$

$$E_R(i) = l \times E_{\text{elec}} \times (size_{\text{CH}}(i) + relays), \tag{15}$$

$$E_F(i) = size_{\text{CH}}(i) \times E_{fuse} \times l, \tag{16}$$

$$E_S(i) = \begin{cases} l \times (E_{\text{elec}} + \epsilon_{fs} \times d^2) \times (1 + relays), & \text{if } d < d_0 \\ l \times (E_{\text{elec}} + \epsilon_{mp} \times d^4) \times (1 + relays), & \text{if } d \geq d_0 \end{cases} \tag{17}$$

where $E_R(i)$ is the energy of clusterhead *i* spent to receive all intra-cluster data, $E_F(i)$ is the energy of clusterhead *i* spent to fuse all intra-cluster data, $E_S(i)$ is the energy of clusterhead *i* spent to transmit *l*-bit data to other clusterhead or sink, $size_{\text{CH}}(i)$ denotes the number of member nodes that belong to the clusterhead *i*, relays is the times of relay, *d* is the distance from clusterhead *i* to its next hop.

Then, the total energy consumption for each round is:

$$E_{\text{total}} = \sum_{i=1}^{K} E_{\text{CH}}(i) + \sum_{j=1}^{N-K} E_{\text{mem}}(j) \tag{18}$$

where *K* is the number of clusterheads and *N* is the number of sensors in the network.

V. SIMULATION RESULTS

We simulate a clustered WSN for 100 nodes in a field with dimensions 100 m × 100 m. Sink is located at (50, 50), the data message size is 30 bytes, λ = 3, μ = 6, initial energy of node is 1 Joule, E_{elec} = 50 nJ/bit, ϵ_{fs} = 10 pJ/bit/m^2, ϵ_{mp} = 0.0013 pJ/bit/m^4, E_{fuse} = 5 nJ/bit, ψ = 40 bps, γ = 50 m/s.

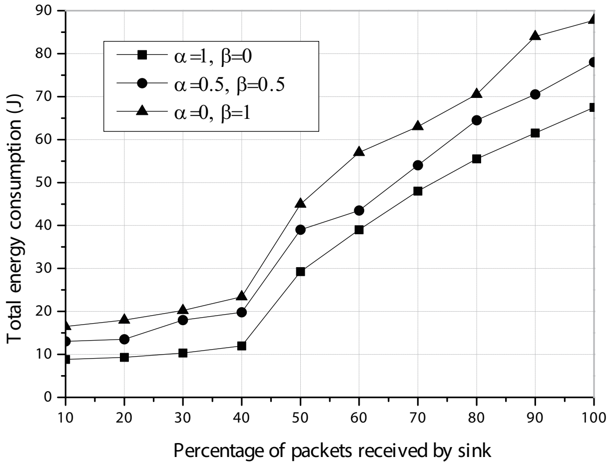


Fig. 5. Effects of α and β on energy consumption.

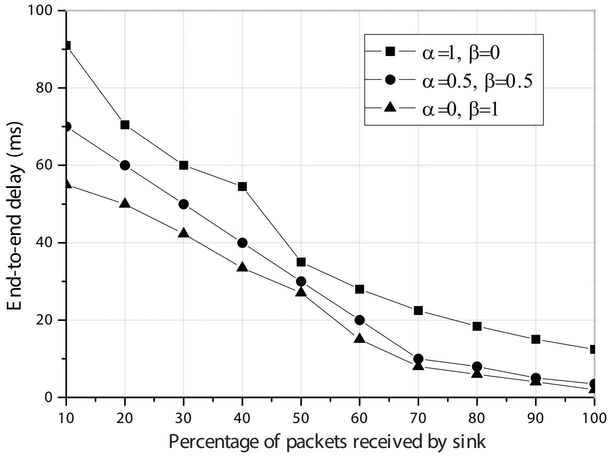


Fig. 6. Effects of α and β on the end-to-end delay.

To see the effect of α and β on DCEM, we set values of α and β to 0 and 1, respectively and measure the end-to-end delay and energy consumption. When α = 0 and β = 1, then variation in the values of TED in (4) is due to the β. Hence, it indicates that end-to-end delay is more important for a given application. On the other hand, when α = 1 and β = 0, then variation in the values of TED is due to the α, which indicates that energy consumption is more important for the given application compared to end-to-end delay. In this experiment, we remove the delay constraint so that the evaluation of the energy consumption and end-to-end delay depends simply on α and β. In Fig. 5, we plot the expected total energy consumption associated with percentage of packets received by sink. As seen, the energy spent in data dissemination decreases as α increases, respectively (α = 0, 0.5, 1). It means that, the more α increases, the better energy efficiency is. In Fig. 6, we plot the expected end-to-end delay associated with percentage of packets received by sink. As seen, the end-to-end delay decreases as the distance $d_{(i,s)}$ increases given that the delay is inversely proportional to $d_{(i,s)}$. Indeed, as the distance between any pair of consecutive forwarders increases, the times that a data packet will be forwarded decreases and hence the end-to-end delay decreases. It means that, the more β increases, the less end-to-end delay is.

In Section IV, we have proposed a new energy-cost function to determine the least-cost route for data dissemination from clusterheads to the sink. In this simulation, we show the primacy of the cost function proposed in (8), (9), and (10) compared with the previous cost functions. In [25], instead of using the consumed energy e_{ij} as the cost function in [26], when a packet is transmitted between node *i* and node *j*, the link cost is essentially equivalent to function $cost_{ij} = e_{ij}/E_i$, where e_{ij} is the energy consumed to transmit data from node *i* to node *j*, E_i is the remaining energy of node *i*. We compare the network lifetime using different cost functions which are $cost_{ij} = e_{ij}$,

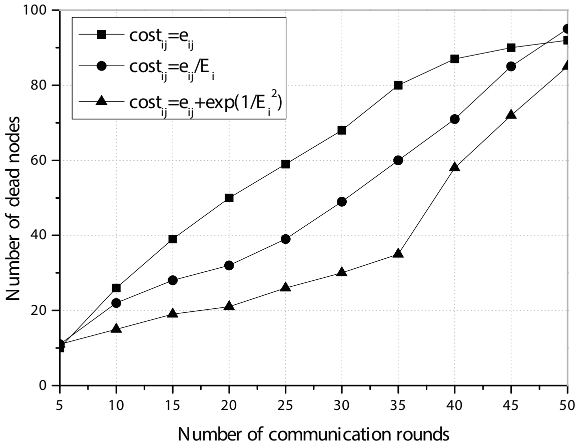


Fig. 7. Number of dead nodes over time.

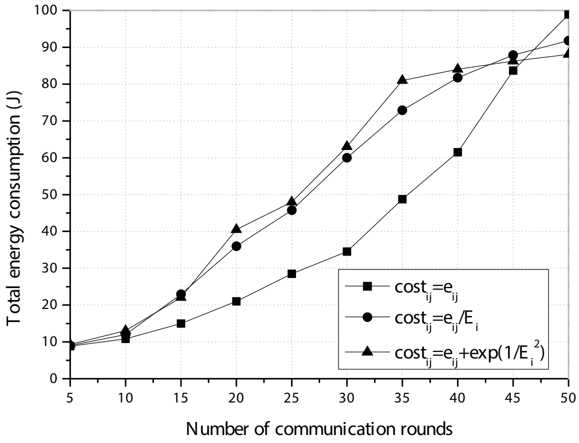


Fig. 8. Total energy consumption over time.

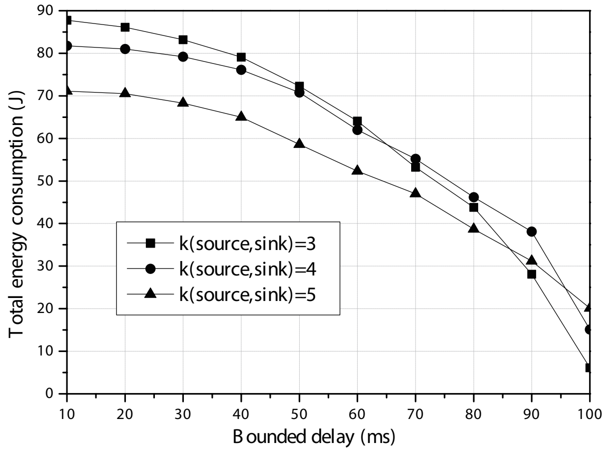


Fig. 9. Energy consumption variation with delay constraint Δ.

$cost_{ij} = e_{ij}/E_i$ and $cost_{ij}$ proposed in (8) and (9). We evaluate the number of dead nodes through each round (dead node is the node that spent more than 95% its energy). As seen in Fig. 7, the line represented by the equation $cost_{ij} = e_{ij} + exp^{\left(1/E_i^2\right)}$ shows that the number of dead nodes increases slowly in the first rounds but increases rapidly in the last rounds. Whereas, number of dead nodes in lines represented by equations $cost_{ij} = e_{ij}$ and $cost_{ij} = e_{ij}/E_i$ increases steadily over time. In Fig. 8, the line represented by the equation $cost_{ij} = e_{ij}+exp^{\left(1/E_i^2\right)}$ shows that the total consumed energy increases steeply in the first rounds but increases gradually in the last rounds. Whereas, total consumed energy in lines represented by equations $cost_{ij} = e_{ij}$ and $cost_{ij} = e_{ij}/E_i$ increases steadily over time. These results are explained by the exponential function of the nodal remaining energy $cost(E^i_{Re})$ ((9)) that we applied in the cost function ((8), (10)). This exponential function varies markedly as the nodal remaining energy has a small change. Thus, it balances the energy consumption among sensor nodes. In fact, if using $cost_{ij} = e_{ij}$, the function $cost_{ij}$ simply depends on the distance between the two nodes *i* and *j* regardless of the nodal remaining energy. However, if using $cost_{ij} = e_{ij}/E_i$, the nodal remaining energy will have a significant effect on the cost function (weight of the nodal remaining energy E_i is equivalent to that of the e_{ij}). Whereas, the function $cost_{ij} = e_{ij} + exp^{\left(1/E_i^2\right)}$ considers the remaining energy of the sensor nodes E_i as an addition parameter, i.e., E_i takes account of a smaller weight than e_{ij}. This makes the remaining energy of the sensor nodes to be more balanced.

For 100 m × 100 m network size and 100 sensor nodes, we change number of data forwarders *k* by adjusting the transmission range of clusterheads r_{CH} (Fig. 2) to see how energy consumption varies with delay constraint Δ. As seen in Fig. 9, energy consumption decreases as the value of Δ increases and vice versa. However, for *k* = 3 (number of hops is *k*+1), energy consumption decreases smoothly as delay increases. For *k* = 4 or 5, the corresponding decrease is not as smooth as in the case *k* = 3.

To gain more insight regarding the behavior of energy consumption and delay metrics with respect to the number of data forwarders, we consider the following plots where both E_{total} ((18)) and $D_{\text{ete}}(x, s)$ ((7)) are plotted on the same figure. Figs. 10–12 show how energy consumption and end-to-end delay vary depending on the number of data forwarders, which helps WSN application designers obtain an idea about the optimal number of hops that could be used to trade-off energy consumption with end-to-end delay. Similar to the first experiment, in this experiment, we also remove the delay constraint so that the evaluation of the trade-off energy consumption and end-to-end delay simply depends on α and β. In Fig. 10, for α = 1 and β = 0, a source could use the *k* = 3 (4 hops) as a good candidate to minimize both metrics. In Figs. 11 and 12, for (α = 0.5 and β = 0.5) or (α = 0 and β = 1), either *k* = 2 or *k* = 3 is also the good choice.

In addition, we evaluate the performance of the DCEM protocol and compare it with generalized low-energy adaptive clustering hierarchy (Gen-LEACH) in [7] and Multihop-HEED in [12]. By simulation, we run 10 experiments that were performed in 50 rounds (each round is 1 second). Each experiment is assigned a distinctive end-to-end delay constraint (we set the bounded delay Δ from 10 ms to 100 ms for experiments, respectively). The results are shown via Figs. 13 and 14.

In Fig. 13, the result is the average value of 10 experiments. In Gen-LEACH, each node *i* elects itself to become a cluster-

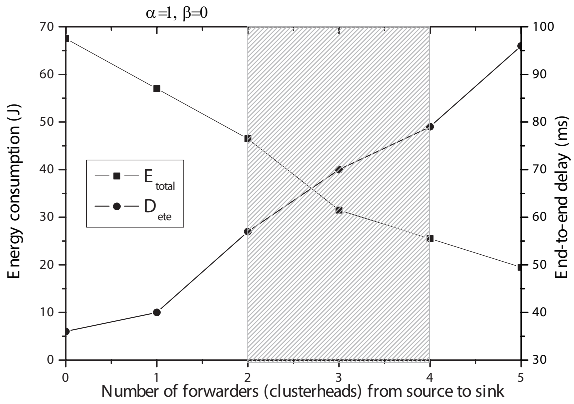


Fig. 10. Trade-off between energy consumption and end-to-end delay; $\alpha = 1$, $\beta = 0$.

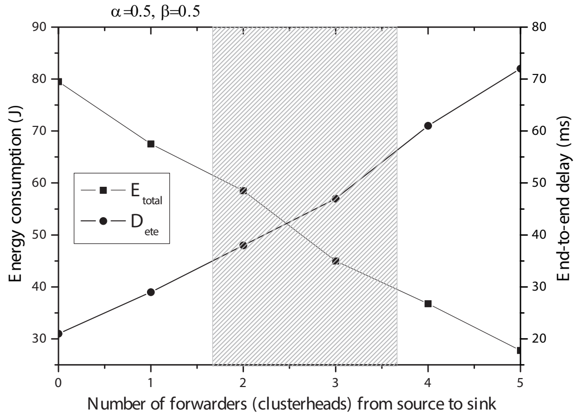


Fig. 11. Trade-off between energy consumption and end-to-end delay; $\alpha = 0.5$, $\beta = 0.5$.

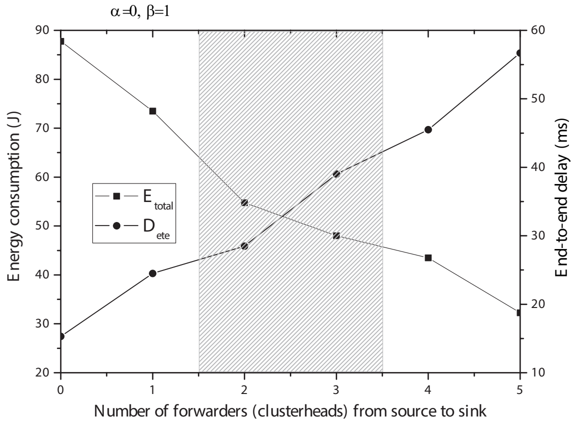


Fig. 12. Trade-off between energy consumption and end-to-end delay; $\alpha = 0$, $\beta = 1$.

head with probability $CH_{\text{prob}}(i) = (E_i/E_{\text{total}} \times k, 1)$, where E_i is the remaining energy of node *i*, and $E_{\text{total}} = \sum_{i=1}^{N} E_i$. For Multihop-HEED, the optimal number of clusterheads k_{opt} is computed for using it as an initial percentage of clusterheads. This may result in slower death of sensor nodes. Gen-LEACH and Multihop-HEED are organized for multihop networks; however, neither of them take interest in the end-to-end delay constraint. Thus, sensor nodes just send data to the sink following the established time slot in the first phase (cluster setup phase) regardless of the end-to-end delay requirement of the application. Therefore, the total energy consumed by the data transmission for DCEM is significantly less than that for both Gen-LEACH and Multihop-HEED. This results in faster death of sensor nodes after each round for both Gen-LEACH and Multihop-HEED compared with DCEM as shown in Fig. 13.

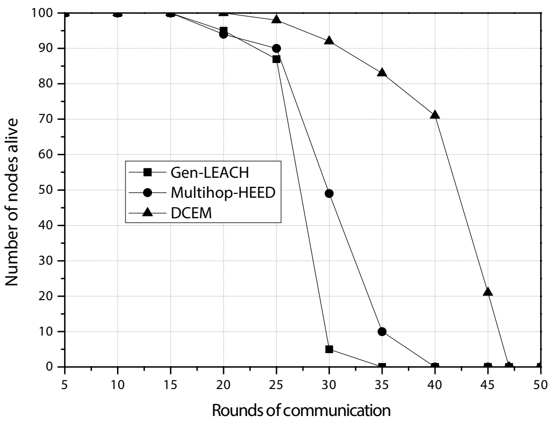


Fig. 13. Performance of Gen-LEACH, Multihop-HEED, and DCEM on number of nodes alive with respect to given delay constraint.

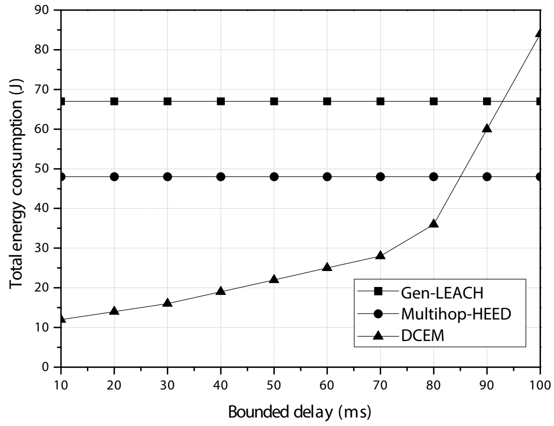


Fig. 14. Performance of Gen-LEACH, Multihop-HEED, and DCEM on total energy consumption with respect to different delay constraints.

In Fig. 14, the total energy consumption for both Gen-LEACH and Multihop-HEED is constant for any values of the bounded delay Δ (48 J for Multihop-HEED, 67 J for Gen-LEACH). Whereas, for DCEM, the total energy consumption increases as the bounded delay Δ increases. Particularly, when the $\Delta \geq 70$ ms, the total energy consumption increases rapidly.

VI. CONCLUSION

In this research, we have proposed a new distributed clustering approach to determine the best clusterhead for each cluster in WSNs in order to trade-off energy consumption and end-to-end delay. The regular nodes join clusters where clusterheads are elected by TED value in relation to both energy consumption and end-to-end delay. We have also proposed a new cost function for the inter-cluster multi-hop routing algorithm based on the new proposed delay model. Hence, we have provided a multi-hop routing algorithm from clusterheads to sink with a minimum energy cost that is subject to an end-to-end delay constraint. Using simulation, we have shown the outstanding performance of our proposal by comparing with other protocols. We have also indicated the optimal parameter values to trade-off between energy consumption and end-to-end delay in a specific network size. In the subsequent work, we will further improve this protocol to find the optimal number of hops for the general case.

REFERENCES

[1] X. Zhang and L. Zhang, "Optimizing energy-latency trade-off in wireless sensor networks with mobile element," in *Proc. IEEE ICPAD*, 2010.

[2] Y. Jin and D. Wei, "Latency and energy - consumption optimized task allocation in wireless sensor networks," in *Proc. IEEE WCNC*, 2010.

[3] H. Liming, "Energy-efficient multi-path routing with short latency and low overhead for wireless sensor networks," in *Proc. IEEE/ACIS SNPD*, 2007.

[4] H. Oh and K. Chae, "An energy-efficient sensor routing with low latency, scalability in wireless sensor networks," in *Proc. IEEE MUE*, 2007.

[5] A. Allirani and M. Suganthi, "An energy sorting protocol with reduced energy and latency for wireless sensor networks," in *Proc. IEEE IACC*, 2009.

[6] O. Boyinbode, H. Le, and M. Takizawa, "A survey on clustering algorithms for wireless sensor networks," *Int'l J. Space-Based Situated Comput.*, vol. 1, no. 2–3, pp. 130–136, 2010.

[7] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application specific protocol architecture for wireless sensor network," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, 2002.

[8] H. Ammari, "On the energy-delay trade-off in geographic forwarding in always-on wireless sensor networks: A multi-objective optimization problem," *Comput. Netw*, vol. 57, pp. 1913–1935, 2013.

[9] Y. Li *et al.*, "An energy efficient distributed algorithm for minimum latency aggregation scheduling in wireless sensor networks," in *Proc. IEEE ICDCS*, 2010.

[10] T. T. Huynh and C. S. Hong, "An energy* delay efficient multi-hop routing scheme for wireless sensor networks," *IEICE Trans.Inform. Syst.*, vol. E89-D, pp. 1654–1661, 2006.

[11] T. T. Huynh *et al.*, "Energy efficient delay-aware routing in multi-tier," in *Proc. IEEE ATC*, 2013, pp. 439–444.

[12] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 660–669, 2004.

[13] S. Bai *et al.*, "DEAR: Delay-bounded energy-constrained adaptive routing in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2012.

[14] L. Wuyungerile *et al.*, "Tradeoff between delay and energy consumption of partial data aggregation in wireless sensor networks," in *Proc. ICMU*, 2010.

[15] A. Durresi *et al.*, "Delay-energy aware routing protocol for sensor and actor networks," in *Proc. IEEE ICPADS*, 2005.

[16] T. Moscibroda *et al.*, "Analyzing the energy-latency trade-off during the deployment of sensor networks," in *Proc. IEEE INFOCOM*, 2006.

[17] R. Cohen and B. Kapchits, "Energy-delay optimization in an asynchronous sensor network with multiple gateways," in *Proc. IEEE SECON*, 2011, pp. 98–106.

[18] A. Shahraki *et al.*, "A new approach for energy and delay trade-off intra-clustering routing in WSNs," *Comput., Math. Appl.*, vol. 62, no. 4, pp. 1670–1676, 2011.

[19] K. Akkaya and M. Younis, "Energy-aware routing of time-constrained traffic in wireless sensor networks," *J. Commun. Syst.*, vol. 17, no. 6, pp. 663–687, 2004.

[20] D. Gross, *Fundamentals of Queuing Theory*, J. Wiley & Sons, 2008.

[21] Anfeng Liu *et al.*, "Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks," *Elsevier Comput. Netw.*, vol. 5, no. 7, pp. 1951–1967, May 2012.

[22] James Stewart, *Calculus: Concepts and Contexts*, Thomson, 2004.

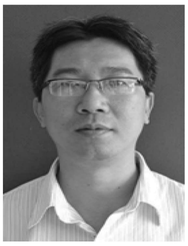
[23] Sedgewick, Rober, *Algorithms in C++: Graph Algorithms (3rd ed.)*, Pearson Education, 2002.

[24] Ernesto de Queiros Vieira Martins *et al.*, "The K shortest paths problem," CISUC, Research Report, 1998.

[25] Chang-Soo Ok *et al.*, "Distributed energy balanced routing for wireless sensor networks," *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 125–135, Aug 2009.

[26] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks," in *Proc. IEEE Radio and Wireless Conference*, (Colorado Springs, CO), 1998, pp. 55–58.

Trong-Thua Huynh was born in Vietnam in 1977. He received his B.Sc. in Computer Science from University of Science Ho Chi Minh city and M.Sc. in Communication Engineering from Kyung Hee University, South Korea in 1999 and 2005, respectively. His current research interests include embedded systems, communication technology and wireless sensor networks. He is currently pursuing his Ph.D. in Computer Science from Ho Chi Minh City University of Technology, Vietnam.



Anh-Vu Dinh-Duc is an Associate Professor at the University of Information Technology - Vietnam National University at Ho Chi Minh City where he has served as Vice-Rector, R&D and External Relations since 2012. He also leads the UIT-VLSI Design group at the Faculty of Computer Engineering. His research interests include WSN, Design Automation of Embedded Systems, Hardware/Software Verification, VLSI CAD, and Reconfigurable Architectures. Anh-Vu Dinh-Duc received the Master and Ph.D. degrees in Microelectronics from the Institute National Polytechnique de Grenoble (INPG), France in 1998 and in 2003, respectively. Anh-Vu Dinh-Duc currently serves as a Program/Organizing Committee Member of several ACM and IEEE conferences. He is a valued Member of the IEEE.

Cong-Hung Tran was born in Vietnam in 1961. He received the B.E. in Electronic and Telecommunication Engineering with First Class Honors from Ho Chi Minh University of Technology in Vietnam, 1987. He received the B.E. in Informatics and Computer Engineering from Ho Chi Minh University of Technology in Vietnam, 1995. He received the M.E. Degree in Telecommunications Engineering course from Postgraduate department Hanoi University of Technology in Vietnam, 1998. He received Ph.D. at Hanoi University of technology in Vietnam, 2004. His main research areas are B-ISDN performance parameters and measuring methods, QoS in High speed networks, MPLS. He is, currently, Associate Professor Ph.D. of Faculty of Information Technology II, Posts and Telecoms Institute of Technology in Ho Chi Minh, Vietnam.