J Intell Manuf
DOI 10.1007/s10845-010-0413-4

A chain-cluster based routing algorithm for wireless sensor networks

Feilong Tang · Ilsun You · Song Guo · Minyi Guo · Yonggong Ma

Received: 30 June 2009 / Accepted: 22 April 2010

Abstract Wireless sensor networks (WSNs) are an emerging technology for monitoring physical world. Different from the traditional wireless networks and ad hoc networks, the energy constraint of WSNs makes energy saving become the most important goal of various routing algorithms. For this purpose, a cluster based routing algorithm LEACH (low energy adaptive clustering hierarchy) has been proposed to organize a sensor network into a set of clusters so that the energy consumption can be evenly distributed among all the sensor nodes. Periodical cluster head voting in LEACH, however, consumes non-negligible energy and other resources. While another chain-based algorithm PEGASIS (power-efficient gathering in sensor information systems) can reduce such energy consumption, it causes a longer delay for data transmission. In this paper, we propose a routing algorithm called CCM (Chain-Cluster based Mixed routing), which makes full use of the advantages of LEACH and PEGASIS, and provide improved performance. It divides a WSN into a few chains and runs in two stages. In the first stage, sensor nodes in each chain transmit data to their own chain head node in parallel, using an improved chain routing protocol. In the second stage, all chain head nodes group as a cluster in a self-organized manner, where they transmit fused data to a voted cluster head using the cluster based routing. Experimental results demonstrate that our CCM algorithm outperforms both LEACH and PEGASIS in terms of the product of consumed energy and delay, weighting the overall performance of both energy consumption and transmission delay.

Keywords Wireless sensor network · Chain based routing · Cluster based routing · Mobile and ubiquitous computing

Introduction

The flexibility, self-organization, low-cost and rapid deployment of wireless sensor networks (WSN) are ideal characteristics to many new and exciting ubiquitous application areas such as data gathering, military, environment monitoring, intelligent control, traffic management, medical treatment, manufacture industry, antiterrorism and so on (Ming et al. 2005; Chao et al. 2006). WSNs will be the essential infrastructure for intelligent ubiquitous services by sensing and collecting the information of various scattered objects. Therefore, recent years have witnessed the rapid development of WSNs.

For any category of networks, routing is always at the center. Distinguishing from other wireless networks, e.g., ad hoc networks, most research efforts on WSN routing algorithms pay much attention to saving energy because in general it is impossible to replace or recharge batteries of sensor nodes. According to existing results, the operation states of a sensor node can be categorized as *transmitting*, *receiving*, *idle* and *sleep*. A sensor node in *transmitting* state consumes the most energy while in *receiving* or *idle* states consumes a little less energy. By comparison, the *sleep* state uses the least energy. Moreover, the energy consumption for data transmission is directly proportional to the square of a wireless transmission

---

F. Tang (✉) · M. Guo · Y. Ma
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: tang-fl@cs.sjtu.edu.cn

I. You
School of Information Science, Korean Bible University,
Seoul, South Korea

F. Tang · S. Guo
School of Computer Science and Engineering, The University of Aizu,
Fukushima 965-8580, Japan

distance. So, existing routing algorithms for WSNs are generally based on a multiple-hop approach.

Multiple-hop routing causes the *imbalanced energy consumption* problem, which means that some sensor nodes in a WSN use up their energy before others because they have to forward data for the latter, especially in a large sensing area.

To alleviate the imbalance of energy consumption, Heinzelman et al. (2000) proposed a routing protocol LEACH that divides a sensor network into a set of clusters. Each cluster votes a node as the head at the beginning of each voting round, and member nodes in a cluster send data to the cluster head, which fuses and then forwards them to the sink in one hop. The LEACH algorithm can somewhat balance and reduce the energy consumption but it induces a significant energy overhead in the head voting during each round.

In order to reduce such energy overhead, the PEAGSIS (Lindsey and Raghavendra 2002) algorithm proposed a fixed chain topology to avoid the periodic head voting. Its long chain topology, however, increases the delay of data transmission. In summary, both LEACH and PEAGSIS can not directly be applied to large-scale WSN based applications.

The objective of this paper is set to address the routing problem for large-scale WSNs, targeting the data gathering applications such as collecting the amount of water, electricity and gas usages in a building. Traditional method for gathering such data is completely manual. For example, a staff of a gas company records the gas meters room by room. It is very time-consuming and inconvenient. Contributions of this paper are described as follows.

(1) We propose a solution to gather the utility of water, electricity and gas in a remote way through WSNs. In our scheme, each water, electricity or gas meter is equipped with a sensor. Whenever a staff with a recorder (i.e., sink) appears in front of a building, sensors are waked up and transfer data (the utility information) in turn to the recorder, based on our proposed CCM routing algorithm. In such a WSN, sensor nodes in the same floor of a building organize as a chain structure, where each chain has the same topology due to the same floor structure.

(2) We design a chain-cluster based routing algorithm CCM, which organizes the sensor nodes as a set of horizontal chains and a vertical cluster with only chain heads. Data transmissions in CCM proceed in two stages: chain routing and then cluster routing. It has the advantages over both chain based routing and cluster based routing in terms of energy consumption, transmission delay and especially the energy × delay metrics.

The remainder of this paper is organized as follows. Section "Related work" reviews related work. Section "Reference ntework model" presents a reference network model for data gathering applications. Section "Chain-cluster based mixed routing algorithm" firstly analyzes how to improve the cluster-based routing protocol LEACH and chain-based routing protocol PEAGSIS and then proposes a mixed chain-cluster based routing algorithm CCM for large-scale WSNs. Section "Experiments and performance evaluation" reports the implementation and performance evaluation on our CCM algorithm. Finally, we conclude the paper with the discussion on our future work in section "Conclusions and future work".

Related work

There have been many research efforts on WSNs since 2000 (Younis and Fahmy 2004). In this section, we will review related routing protocols, with a focus on the hierarchical-based routing in WSNs.

Existing research results on WSNs are generally built on a flat architecture, where randomly distributed sensor nodes constitute a self-organizing network with a *sink* connecting to outside wired or wireless networks. Each of these scattered sensor nodes has the capability to collect data and route data back to the sink. For saving energy, each node can only transmit to and receive from its neighbor nodes by adjusting its radio power. The sensed data are sent back to the sink hop by hop, using various routing protocols (Akyildiz et al. 2002; Karlof et al. 2003).

Different routing protocols have different design considerations, but energy saving is always the common and the most important goal. From the perspective of self-organized network architecture, many routing protocols can be categorized as flat-based routing and hierarchical-based routing (Al-Karaki and Kamal 2004).

In the flat-based routing, all nodes are typically assigned an equal role. *Flooding* is a classical protocol to relay data in WSNs but with several serious deficiencies such as implosion, overlaping and resource blindness(Akyildiz et al. 2002; Heinzelman et al. 1999). In *flooding*, each node receiving a data or management packet broadcasts the packet to all of its neighbors, unless a maximum number of hops for the packet is reached or the destination of the packet is just the node itself. *Gossiping*, a derivation of flooding, sends data to only one neighbor which is randomly selected to avoid the implosion problem, but message propagation in Gossiping takes longer time (Akyildiz et al. 2002). As a family of the adaptive protocols, *SPIN* (Sensor Protocols for Information via Negotiation) addresses the deficiencies of classical flooding by considering resource adaptation and data negotiation among nodes. In SPIN, whenever a packet is available, a node only broadcasts a description instead of the packet itself and sends it only to the sensor nodes that express interest (Heinzelman et al. 1999; Kulik et al. 2002). *Directed Diffusion* (Intanagonwiwat 2001) is a data-centric and applicationaware paradigm in the sense that all data generated by sensor nodes are named by attribute-value pairs. The basic idea of this protocol is to combine the data coming from different sources (in-network aggregation) together to minimize the transmission loads by eliminating redundancy. *Rumor routing* (Braginsky and Estrin 2002) is a variation of the directed diffusion. It only routes the query messages to the nodes involved in a specified event rather than the entire network. The *CADR* (constrained anisotropic diffusion routing) protocol is a general form of the directed diffusion, with the concern to maximize information gain and to minimize latency and bandwidth. Based on the observation that the direction of routing always is towards the fixed external base-station, *MCFA* (minimum cost forwarding algorithm) (Ye et al. 2001) does not maintain a routing table, but the route with the least cost estimation from itself to the base-station instead.

To save energy and improve the routing efficiency and scalability, many hierarchical-based routing algorithms have been proposed. In such routing algorithms, different sensor nodes play different roles. Generally speaking, the nodes with higher battery energy can be used to aggregate and relay data while others are only used to sense targets. Hierarchical routing provides an efficient way to prolong the network lifetime by distributing energy consumption evenly, as well as performing data aggregation to decrease the transmitted data volume. *LEACH* (Heinzelman et al. 2000), *TEEN* (threshold sensitive energy efficient sensor network protocol) (Manjeshwar and Agrawal 2001) and *PEGASIS* (Lindsey and Raghavendra 2002) are the representatives of such hierarchical-based routing protocols.

Based on a 2-level hierarchical routing, the LEACH protocol that is designed to reduce global energy dissipation and to distribute energy consumption evenly across all nodes. In *LEACH*, a WSN is partitioned into clusters such that each cluster has a cluster head and a few member nodes. The heads receive data from the member nodes in their cluster, aggregate the data, and forward them to the sink. The even energy dissipation is achieved by randomly re-choosing the cluster heads at regular intervals. It leads to a 8 times improvement compared to the direct transmission protocol. While the *TEEN* protocol (Manjeshwar and Agrawal 2001) adopts the same clustering model used by *LEACH*, a cluster node in *TEEN* sends a hard threshold and a soft threshold to its members for reducing the delay of data transmission. As the hard threshold is reached, a new hard threshold is set up and sensed data will be sent out in the next slot. As a result, users can control the trade-off between energy efficiency and data accuracy. The *PEGASIS* protocol improves *LEACH* in the sense that it consumes less energy per round. All sensor nodes in a WSN form a chain, and each sensor node communicates with the closest neighbor. Moreover, sensor nodes in *PEGASIS* act as the chain head in turn. The sensed data are forwarded and fused node by node. Eventually, the chain head transmits the fused data to the sink. It was reported that *PEGASIS* outperforms *LEACH* by 1–3 times under different network sizes and topologies (Lindsey and Raghavendra 2002).

Reference network model

As mentioned in Section “Introduction”, this paper focuses on how to gather the utility information of water, electricity and gas usage via WSNs in an energy- and time-efficient way. For such applications, sensor nodes are evenly distributed in a 2-dimension area, and the sink, which is held by a company staff, is away from the area. Accordingly, we have the following assumptions for such WSNs.

- All sensor nodes can directly communicate with the sink;
- All sensor nodes are homogeneous and have the same initial energy supply;
- Radio channel is symmetric, i.e., the energy consumption for transmitting a message from one node to another is the same as on the reverse direction; and
- Energy consumption for a data transmission only depends on (1) the distance between a sender and a receiver and (2) the size of the data packet.

Figure 1 illustrates the architectural model of such a WSN within a 50 m × 50 m area. The network consists of 100 sensor nodes and is divided to 10 strips, each with 10 nodes distributed in a 5-meter apart. The sink is at the point (60, 25) i.e., it is 10 m away from the closest sensor node. The power of each sensor node can be adjusted so that it can communicate only with the direct neighboring nodes (Ramanathan and Hain 2000).

For the simplicity of energy analysis, we adopt a first-order radio model used in LEACH (Heinzelman et al. 2000). Energy consumptions in circuitry for running the transmitter or receiver and in radio amplifier for wireless communication are $E_{circuitry} = 50$ nJ/bit and $E_{amplifier} = 100$ pJ/bit/m^2, respectively. The value of $E_{amplifier}$ is directly proportional to the square of transmission distance. Therefore, we have the following formula, where k is the size of transmitted packets, and d is the distance between a transmitter and a receiver.

The energy for transmitting a packet is:

$$E_{transmit}(k, d) = E_{circuitry} \times k + E_{amplifier} \times k \times d^2 \quad (1)$$

The energy for receiving a packet is:

$$E_{receive}(d) = E_{circuitry} \times k \quad (2)$$

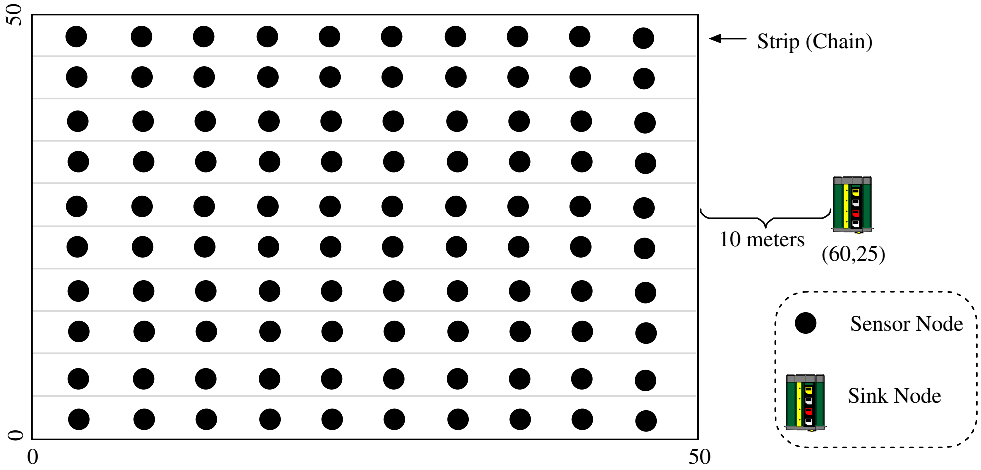


Fig. 1 A sensor network with 10 strips (chains), each with 10 sensor nodes

Chain-cluster based mixed routing algorithm

Analysis of the LEACH and PEGASIS algorithms

A simple method for data collection is that each sensor node directly transmits sensed data to the remote sink. However, data transmission in this method consumes too much energy and most nodes will drain of energy very quickly because the sink is far away. As a classical cluster-based routing algorithm, LEACH uses a randomization approach to distribute the energy load evenly among sensors through clustering. Each cluster randomly chooses a head node periodically, and other nodes in the cluster send data to the head node, which fuses received data and then directly transmits them to the sink in one hop. The LEACH protocol runs in phases. In the first phase, cluster heads are elected. In the second phase, the cluster head sets up its cluster by inviting others to join. In the third phase, the cluster head decides and notifies its member nodes a time schedule, which avoids conflicts caused by overlapped transmission. The cluster members then transmit data at their own time slots.

In LEACH, a cluster head spends more energy than its member nodes for data aggregation and forwarding. To prevent the imbalance in energy consumption, cluster heads are voted periodically. A new cluster head will be chosen in the next round if the residual energy of current cluster head becomes lower than a specified threshold. The policy that a cluster head is periodically generated evens the energy consumption. This approach is useful to balance energy load, resulting in a prolonged network lifetime by 15% on the average. On the other hand, LEACH has the following disadvantages.

- The voting process spends non-negligible additional energy and causes serious overhead in network traffic.
- Cluster heads need to notify member nodes in a broadcast way. Each member node detects the signal strength of different cluster heads and responds to the cluster head with the highest residual energy. This process increases the communication overhead.

To reduce the energy overhead for clustering, PEGASIS uses a greedy algorithm that organizes sensor nodes as a chain. The main idea is that each sensor node communicates only with its closest two neighbors at low power consumption, and serves as the chain head in turn.

At the beginning of each round, a chain head is voted and then it generates and passes a token. Only the node holding the token can send packets. In this way, packets are transferred in turn until they reach the chain head. Finally, the chain head sends data to the sink. Compared with LEACH, PEGASIS avoids the overhead for setting up clusters and uses less energy because the distance of packet transmission is reduced significantly. At the same time, sensor nodes act as the chain head in turn such that the energy load is evenly distributed among the sensor nodes in the network.

On the other hand, PEGASIS performs data fusion at every node in the chain except the two end chain nodes. Each node fuses its neighbor's data with its own data to a single packet and then transmit it to the next neighbor (if it is not the chain head). The chain head will be at a random position during each round. In a given round, PEGASIS uses a simple control token generated by the leader to initiate the data transmission from the end of the chain. The overhead is negligible since the token size is very small (Lindsey and Raghavendra 2002). As a result, the lifetime of WSNs using PEGASIS increases by around 100% compared to LEACH. Unfortunately, PEGASIS significantly induces a much longer data transmission delay because of large number of hops in a long chain.

Table 1 Notations

Symbol	Description
N	The number of sensor nodes in a WSN
n	The number of sensor nodes in a strip (i.e., chain)
L	The length and width of the WSN
h	The height of each strip
k = L/h	The number of strips (chains)
chain (i)	The ith chain ($1 \leq i \leq k$)
S(i, j)	The jth sensor node in the ith chain ($1 \leq i \leq k$ and $1 \leq j \leq n$)
data (i, j)	Data sensed by the node S(i, j)
E(i, j)	Current residual energy of the node S(i, j)
r	The number of rounds
E_i(r)	Total energy consumption of the ith node after the rth round
E	Average energy consumption of each round
D	Network delay in a test

The CCM algorithm

In this section, we present our mixed routing algorithm CCM that combines the advantages of PEGASIS with low energy consumption, and LEACH with short transmission delay. Our motivation is to improve the routing performance in terms of the energy × delay metrics (Lindsey et al. 2001) that is important in many time-critical applications.

Let N sensor nodes be distributed in a 2-dimension area with the size of L(m) × L(m). We divide this area as a set of strips, as shown in Fig. 1. Each strip has a height of h(m) and there are k = L/h strips in total in the sensor network. We consider n sensor nodes in each strip, which constitute a chain in a self-organized manner in the first stage of our CCM algorithm. All symbols used in this paper are described in Table 1.

Our CCM algorithm works in two phases:*chain based routing* and *cluster based routing*, which are discussed in details as follows.

Phase 1: Chain based routing In this phase, sensor nodes in each chain transmit their data to their own chain head using the chain based routing. Each sensor node is located in a 2-dimension location with coordinate (x, y). In our target environments (see Fig. 1), sensor nodes are evenly distributed in a symmetrical building so that we can use the 2-dimension serial number (e.g., S(1, 2)) of a sensor node as its 2-dimension coordinate. A sensor node can transmit data only to its closest two neighboring nodes in the same chain. Our chain based routing works in the following steps.

Selection of chain head node Sensor nodes in the chain (i) are marked as S(i, 1), S(i, 2),…, and S(i, n). To even the energy consumption of sensor nodes in a chain, every node in the chain acts as the chain head in turn. In our CCM algorithm, the node S(i, j) is assigned as the chain head node during the jth round($1 \leq j \leq n$). In this way, CCM does not need to spend energy for voting the chain head.

S(1,1) → S(1,2) → **S(1,3)** ← S(1,4) ← S(1,5)

Fig. 2 Data transmission in the chain (1)

Data transmission in a chain Our CCM schedules the data transmission through an improved token mechanism. At the beginning of each round, the chain head node generates two tokens and then transmits them to two end nodes, i.e., the 1th and nth nodes, respectively. The two end nodes in a chain transmit data to their individual neighboring nodes in parallel. The latters fuse the received data with their own data and transmit them to the next neighboring node, respectively. The data are transmitted in an alternative way until all data are transmitted to the chain head node. In the third round, for example, the node S(1, 3) is selected as the chain head node as shown in Fig. 2. At the beginning of the third round, S(1, 1) and S(1, 5) respectively receive a token from the chain head S(1, 3) so that they can transmit their data and the token to node S(1, 2) and S(1, 4) in parallel. After receiving the token and data (1, 1), node S(1, 2) fuses its own data with the data (1, 1), and then sends the fused one with the token to the chain head S(1,3). At the same time, S(1,4) and S(1,5) send their data to the chain head S(1,3) as well. Finally, the chain head S(1,3) fuses all data and destroys the two tokens. Sensor nodes in other chains transmit their data in the same way.

Since there are k chains in a WSN, data transmissions in different chains may interfere with each other due to the radio conflict. If sensor nodes have CDMA functionalities, or distance between neighboring chains is parted far enough without any radio conflict, sensor nodes in different chains

```
Chain Based Routing:
  the sensor node S(i,j) is assigned as the chain head in the $j^{th}$ round;
  S(i,j) generates 2 tokens and sends them to s(i,1) and s(i,n) respectively;
  let x=1 and y=n;
  repeat
     if (x<j)     // data transmission in the left side of S(i,j)
     {     S(i,x) fuses received data(i,x-1) (x>1) and its own data(i,x);
           S(i,x) transmits the fused data and the token to its neighboring s(i,x+1);
           x=x+1;    }
     if (y>j)     // parallel data transmission in the right side of S(i,j)
     {     S(i,y) fuses received data(i,y+1)(y<n) and its own data(i,y);
           S(i,y) transmits the fused data and the token to its neighboring s(i,y-1);
           y=y-1;      }
  until (x=j) and (y=j)
```

Fig. 3 The chain based routing in the chain (i) (Phase 1)

can transmit data simultaneously. Otherwise, sensor nodes in different chains have to been transmitted in a sequential way.

We illustrate the chain based routing in the chain (i) ($1 \leq i \leq k$) during the jth round in Fig. 3, which is the phase 1 of our CCM routing algorithm. Note that the second-dimension number of sensor nodes in a chain increases one by one from the left to the right in a chain and CCM selects the jth node S(i,j) as the chain head node during jth round. In Fig. 3, S(i, x) ($x < j$) denotes the nodes on the left of S(i, j), while the S(i,y) ($y > j$) represents the nodes on the right of S(i, j). A chain based routing starts from the two end nodes, i.e., S(i, 1) and S(i, n), and propagates data to the chain head S(i, j) hop by hop. Each node fuses the data from its precedent node with its own data, except the two end nodes. After the completion of the chain routing, CCM goes to phase 2.

Phase 2: Cluster based routing As described above, the jth node S(i, j) in the chain(i) ($1 \leq i \leq k$) works as the chain head during the jth round of the chain based routing. After every chain head receives and fuses data from its own chain, these chain heads form a new cluster while all other nodes in each chain enter the *sleep* state. Cluster based routing includes the following steps.

Voting cluster head All chain head nodes group as a cluster after the completion of chain based routing. A cluster head is chosen based on its residual energy at the beginning of cluster based routing. Each node in the cluster broadcasts its residual energy to other cluster members. After receiving such messages, each node in the cluster compares its own residual energy with other node's residual energy. If any node finds that its residual energy is higher than any other else, it applies for working as the cluster head by advertising to other nodes in the cluster. Otherwise, it simply drops voting messages. A possible conflict is that two or more nodes have the exact same residual energy. In this case, the node that firstly issued the advertisement will be a cluster head.

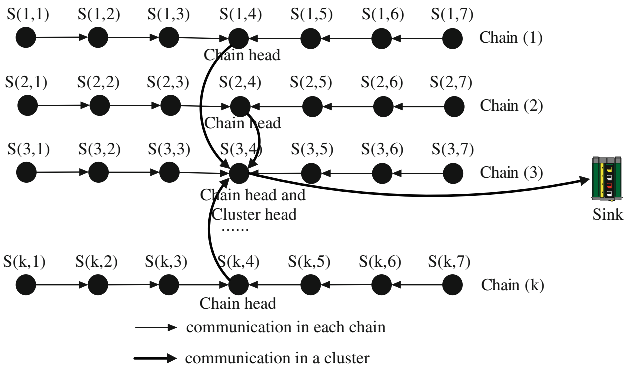


Fig. 4 Data transmission in chains and a cluster

```
Cluster Based Routing:
  While (t< $t_{expired}$)
  {     each cluster member S(x,j) broadcasts its residual energy E(x,j);
        S(x,j) listens vote messages from other members S(y,j)
     if ( E(y,j)>E(x,j) )
           S(x,j) drop out of the voting;
  }
  the survived node wins the voting and advertises the time schedule;
  other cluster members send data to the cluster head;
  the cluster head fuses all data and sends the fused data to the sink;
```

Fig. 5 The cluster based routing (phase 2)

Data transmission in the cluster The cluster head assign individual TDMS time slot for each member node so that the members can transmit their sensed data to the cluster head in their own time slot. The cluster head node fuses all data and sends the fused data to the sink directly. As shown in Fig. 4, after the 4th round in phase 1, nodes S(1, 4), S(2, 4), … and S(k, 4) form a cluster. If sensor node S(3, 4) is selected as the cluster head because it has the highest residual energy level, other nodes send their data to S(3, 4), which fuses and transmits the data to the sink.

Cluster based routing algorithm is illustrated in Fig. 5, where the parameter $t_{expired}$ is a timeout for cluster head voting. After the jth round chain routing, k chain head nodes S(x, j) ($1 \leq x \leq k$) forms a cluster. A cluster head is chosen based on its current residual energy.

Experiments and performance evaluation

In order to evaluate the performance of our CCM routing algorithm, we developed a simulation system using the tool *SWANS* (scalable wireless ad-hoc network simulator). In the simulation system, we tested two groups of sensor networks, 100 sensor nodes distributed in different sizes of 50 m × 50 m and 100 m × 100 m grids like in Fig. 1, respectively. The size of each packet was set to 2 k bits. The time for transmitting such a packet is considered as 1 unit of delay. A sufficient number of rounds were conducted such that each node in a chain works as the chain head and cluster head at least once.

Table 2 Parameters for DT, LEACH, PEGASIS and CCM

Algorithms	Number of clusters	Number of nodes in a cluster	Number of chains	Number of nodes in a chain	Number of root nodes
DT	/	/	/	/	100
LEACH	5	20	/	/	5
PEGASIS	/	/	1	100	1
CCM	1	5	5	20	1

We use the metrics *consumed energy* × *delay* to evaluates various routing algorithms, which is appropriate for our target application environments, as described in Section "Introduction". The average energy consumption per round can be estimated as

$$E = \sum_{i=1}^{N} E_i(r)/r \tag{3}$$

To compare our algorithm with related proposals, we tested the average energy consumption (E) and the delay (D) for a group of algorithms, including direct transmission(*DT*), *LEACH*, *PEGASIS* and our *CCM*. The simulation settings for each algorithm are summarized as follows.

- *DT* Each sensor node directly transmits its own data to the sink in one hop at high transmission power because of the long wireless transmission distance.
- *LEACH* We divided the network as five 20-node clusters. In each cluster, one node works as the cluster head while other 19 member nodes are under the control of the cluster head. The cluster head fuses the data from its own cluster and then forwards the fused data to the sink.
- *PEGASIS* All sensor nodes transfer data to one chain head.
- *CCM* The network was evenly divided into 5 chains. Each chain consists of 20 sensor nodes.

Table 2 lists the parameters of the four algorithms in details. We define the sensor node that directly relays data to the sink by one hop as the *root node*. Since the distance between the root node and the sink is much longer than that between two neighboring sensor nodes in our target environments, the *root node* consumes much more energy for data relay than others. As a result, DT is the most energy-consuming since all nodes work as root nodes, while other algorithms can distribute root nodes among all sensor nodes.

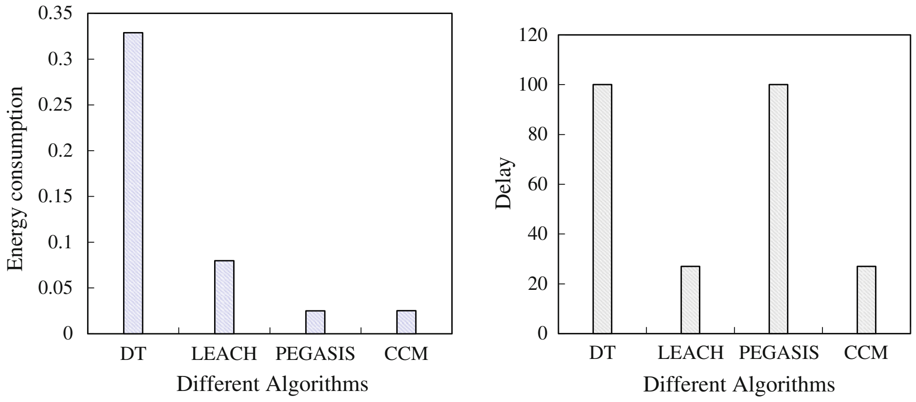
Performance evaluation and comparison among DT, LEACH, PEGASIS and CCM are shown in Fig. 6, where sensor nodes were evenly distributed within a 50 m×50 m area. From Fig. 6a, we can find that the energy consumption of our CCM is almost the same as PEGASIS, but is only 31.5% of LEACH. In Fig. 6b, the delay of our CCM is almost the same as LEACH, but is only 27% of PEGASIS. These results show that our CCM algorithm merges the advantages of PEGASIS (low energy consumption) and LEACH (low transmission delay). Fig. 6c further validates this result, where our CCM is 280% and 260% better than LEACH and PEGASIS in terms of *energy* × *delay* (i.e., $E \times D$) metrics, respectively. DT always performs worse than LEACH, PEGASIS and CCM in terms of energy consumption, delay or *energy* × *delay* metrics. In summary, LEACH is an appropriate solution to time-critical applications, PEGASIS is more energy-efficient but with a longer delay, while our CCM can meet both requirements for a prompt-response and energy-saving applications.

When the network area was set to 100 m×100 m, performance results for DT, LEACH, PEGASIS and CCM are illustrated in Fig. 7. The energy consumption of our CCM algorithm is a little higher than PEGASIS, but is still 22% of LEACH, as shown in Fig. 7a. The differences in the delay for the four algorithms are similar to Fig. 6b and thus we omit it here. Regarding to the energy×delay metrics, our CCM is 353% and 196% better than LEACH and PEGASIS, respectively, which is shown in Fig. 7b. Finally, we can conclude that our CCM improves the performance of LEACH and PEGASIS significantly in terms of energy×delay under different sizes of WSNs.

Conclusions and future work

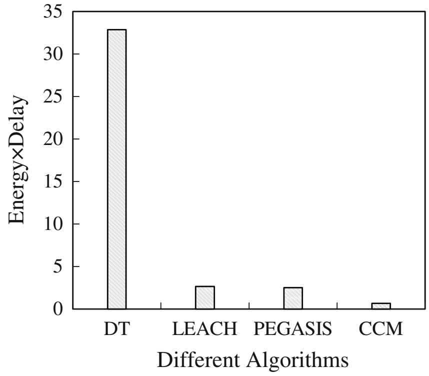
We have presented a mixed chain-cluster based routing algorithm CCM for data gathering. It organizes the sensor network as a set of horizontal chains and a vertical cluster and routes data in two phases. In the first phase, sensor nodes in each chain send data to the chain head, using chain based routing. In the second phase, all the chain heads form a cluster and send the data, which are fused from their own chains, to a voted cluster head. Finally, the cluster head further fuses data and transmits them to the remote sink. CCM exploits the full advantages of LEACH and PEGASIS and to some extent alleviates their weakness, i.e., extra energy overhead for voting cluster heads periodically in LEACH, and long transmission delay in PEGASIS. The experiments show that our algorithm CCM is much better than LEACH and PEGASIS in terms of energy×delay metrics.

Fig. 6 Performance comparison among DT, LEACH, PEGASIS and CCM in the 50 m × 50 m network



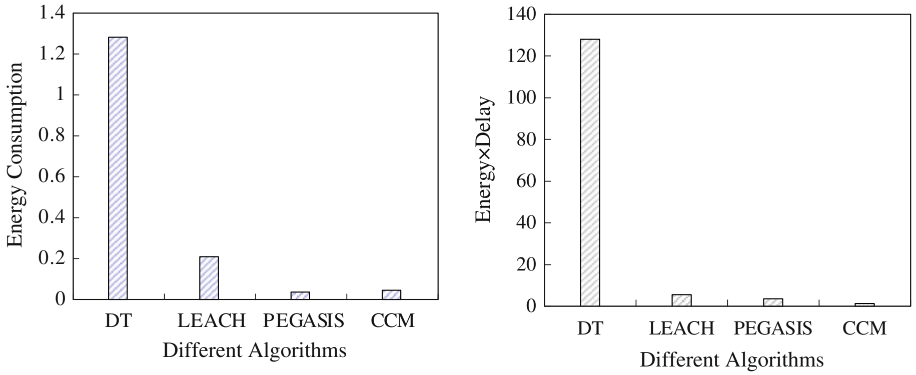
(a) Average energy consumption during a round.

(b) Transmission delay.



(c) Energy × delay during a round.

Fig. 7 Performance comparison among DT, LEACH, PEGASIS and CCM in the 100 m × 100 m network



(a) Average energy consumption during a round.

(b) Energy × delay during a round.

As part of our future work, we are going to investigate how to optimize the procedure of cluster head voting by incorporating both the priorities and the residual energy into cluster head selection to further improve the performance of our routing algorithm.

Acknowledgements Feilong Tang would like to thank The Japan Society for the Promotion of Science (JSPS) and The University of Aizu (UoA), Japan, for providing the excellent research environment during his JSPS Postdoctoral Fellowship (ID No. P 09059) Program in UoA, Japan. Thanks are also given to Dr. Cho-Li Wang in The University of Hong Kong and Professor Zixue Cheng in UoA, Japan, for their precious helps.

This work was supported by the National Natural Science Foundation of China (NSFC) (Grant Nos. 60773089 and 60725208), and the National High Technology Research and Development Program (863 Program) of China (Grant Nos. 2006AA01Z172 and 2008AA01Z106).

References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine, 40*(8), 102–114.

Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A Survey. *IEEE Wireless Communications, 11*(6), 6–28.

Braginsky, D., & Estrin, D. (2002). Rumor routing algorithm for sensor networks. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002)* (pp.22–31), Sep 23–28, 2002, Georgia, USA.

Chao, H. Y., Chen, Y. Q., & Ren, W. (2006). A study of grouping effect on mobile actuator sensor networks for distributed feedback control of diffusion process using central voronoi tessellations. *International Journal of Intelligent Control and Systems, 11*(2), 185–190.

Heinzelman, W.R., Chandrakasan, A., & Balakrishnan, H. (2000). LEACH: Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of Hawaii International Conference on System sciences* (pp. 3005–3014), 4–7 Jan, 2000, Maui, Hawaii.

Heinzelman, W. R., Kulik, J., & Balakrishnan H. (1999). Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5*th*ACM International Conference on Mobile Computing and Networking(MobiCom 99)* (pp. 174–185), 15–20 Aug, 1999, Seattle, WA, USA.

Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000). Directed Diffusion: A scalable and roubst communication paradigm for sensor networks. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002)* (pp. 56–67), Sep 23–28, 2002, Georgia, USA.

Karlof, C. Li, Y., & Polastre, J. (2003). ARRIVE: An architecture for robust routing in volatile environments. *University of California at Berkeley, Technical report UCB-CSD-03-1233*.

Kulik, J., Heinzelman, W. R., & Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks, 8*(2), 169–185.

Lindsey, S., & Raghavendra, C. (2002). PEGASIS: Power-efficient gathering in sensor information systems. In *Proceedings of IEEE Aerospace Conference*, Vol. 3 (pp. 1125–1130), March 2002.

Lindsey, S., Raghavendra C. S., & Sivalingam K. (2001). Data gathering in sensor networks using the energy*delay metric. In *Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing* (p. 188), Apr 23–27, 2001, San Francisco, USA,

Manjeshwar, A., & Agrawal, D. P. (2001). TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 15th Parallel and Distributed Processing Symposium* (IPDPS-01) (pp. 2009–2015), Apr 23–27, 2001, San Francisco, USA.

Ming, Y. U., Aniket, M., & Wei, S. U. (2005). An environment monitoring system architecture based on sensor networks. *International Journal of Intelligent Control and Systems, 10*(3), 201–209.

Ramanathan, R., & Hain, R. (2000). Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings Infocom 2000* (pp. 404–413), March 26–30, 2000, Tel-Aviv, Israel.

Ye, F., Chen, A., Liu, S., & Zhang, L. (2001). A scalable solution to minimum cost forwarding in large sensor networks. In *Proceedings of the tenth International Conference on Computer Communications and Networks (ICCCN 2001)* (pp. 304–309), July 7–14, 2001, Vancouver, British Columbia, Canada.

Younis, O., & Fahmy, S. (2004). HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing, 3*(4), 366–379.