J Supercomput (2009) 47: 127–145
DOI 10.1007/s11227-008-0181-5

Adaptive location updates for mobile sinks in wireless sensor networks

Guojun Wang · Tian Wang · Weijia Jia · Minyi Guo · Jie Li

Published online: 21 March 2008

Abstract Mobile sinks can be used to balance energy consumption for sensor nodes in Wireless Sensor Networks (WSNs). Mobile sinks are required to inform sensor nodes about their new location information whenever necessary. However, frequent location updates from mobile sinks can lead to both rapid energy consumption of sensor nodes and increased collisions in wireless transmissions. We propose a new solution with adaptive location updates for mobile sinks to resolve this problem. When a sink moves, it only needs to broadcast its location information within a local area other than among the entire network. Both theoretical analysis and simulation studies

G. Wang (✉)
School of Information Science and Engineering, Central South University, Changsha, 410083, China
e-mail: csgjwang@mail.csu.edu.cn

G. Wang
e-mail: csgjwang@gmail.com

T. Wang · W. Jia
Department of Computer Science, City University of Hong Kong, Kowloon Tong, Kowloon, Hong Kong

T. Wang
e-mail: tianwang@cityu.edu.hk

W. Jia
e-mail: itjia@cityu.edu.hk

M. Guo
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China
e-mail: guo-my@cs.sjtu.edu.cn

J. Li
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan
e-mail: lijie@cs.tsukuba.ac.jp

show that this solution consumes less energy in each sensor node and also decreases collisions in wireless transmissions, which can be used in large-scale WSNs.

Keywords Wireless sensor networks · Mobile sinks · Routing protocol · Location update

1 Introduction

Recent advances in MEMS (Micro Electro Mechanical Systems)-based sensor technology, embedded computing, and low-power wireless communications have enabled the development of low-cost, multi-functional sensors, which are equipped with sensing, computation, and wireless communication units. Hundreds or even thousands of such sensors form Wireless Sensor Networks (WSNs). WSNs consist of a large number of low-cost and low-power sensor nodes which can cooperatively monitor their surrounding environments and these sensors send the sensed data to the data collecting center, the sink, for further processing. It can be used for a wide range of applications such as military surveillance, industry control, traffic control, and ambient conditions detection [1–3]. Being different from traditional wireless networks, sensor nodes in WSNs are severely constrained in resources such as energy, computation, communication, and memory, so the data gathering and routing protocols must be energy efficient in order to prolong the lifetime of the entire network [4–10].

At present, many routing protocols have been proposed for WSNs, most of which are designed for static nodes, assuming that both the sensor nodes and the sink do not change their location after deployment. Since geographic routing makes the routing decision at each node using only the location information of both the destination node and the neighbors of the forwarding node, it is quite suitable for WSNs. Recently, thanks to the advances in robotics [11], mobile sinks have been introduced into WSNs due to two main reasons. In the first place, mobile sinks have been utilized as mechanical data carriers to prolong the network lifetime. As we know that in static WSNs, the sensor nodes close to the sink will deplete their energy quickly because they have to forward messages originating from many other nodes, and thus shorten the lifetime of the entire network [12]. If the sink can move, the sensor nodes in the network can take turns to become the neighbors of the sink, and thus the energy can be consumed evenly among all the sensor nodes, and consequently, the lifetime of the entire network can be prolonged [13]. In the second place, some applications need the support of mobile sinks. For example, consider a forest patrol that makes periodic patrols in the forest. These patrols aim at preventing poaching and monitoring endangered wildlife. However, their efficacy is limited because they cannot cover the entire area assigned to them. This problem can be solved by deploying a wireless sensor network with mobile sinks, where each mobile sink functions as a patrol (see Fig. 1 for an illustration).

Generally speaking, sensor nodes do not have a priori knowledge of the location information of the mobile sinks. A naive approach to addressing the problem of communicating with a mobile sink whose location information is unknown is through flooding. While flooding ensures that the mobile sink receives data packets from

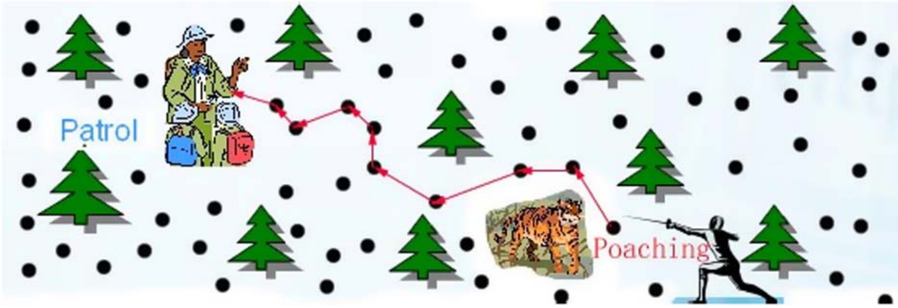


Fig. 1 An application that uses the mobile sink

sensor nodes, the data rate that can be supported in the network may be very low due to drastically increased collisions in transmitting flooding packets from sensor nodes [14]. Another approach is for the sink to consecutively inform its new location information to the sensors, such as DRP (Dynamic Routing Protocol) [15] and GRAB (GRAdient Broadcast) [16]. DRP takes a data-centric naming approach to enable intra-network data aggregation. GRAB targets at robust data delivery in an extremely large sensor network consisting of highly unreliable nodes and it uses a forwarding mesh instead of a single path, where the mesh's width can be adjusted on the fly for each data packet. But both DRP and GRAB require that the mobile sink needs to continuously propagate its location information throughout the entire network, so that all the sensor nodes get updated with the direction of sending future data reports. However, frequent location updates from the sink can lead to both large energy consumption of the sensor nodes and collisions in wireless transmissions [17].

To address this problem, we propose an Adaptive Local Update-based Routing Protocol (ALURP). When the sink node moves, it only needs to update its location information within a local area other than among the entire network, so it consumes less energy in each sensor node and also decreases the probability of collisions in wireless transmissions, and thus it can be used in large-scale WSNs.

The remainder of the paper is organized as follows. The proposed protocol is presented in Sect. 2. Section 3 gives some theoretical analysis. Section 4 presents our simulation studies. Section 5 concludes the paper.

2 Protocol design

As the ALURP protocol is an improvement of the LURP protocol which is our previous work [18], in this section, we first introduce LURP, and then propose ALURP.

2.1 Assumptions

(1) Sensors remain static but the sink can move freely in the network.
(2) Each node as well as the sink node knows the location of itself, the location of its 1-hop neighbors, by some localization algorithms [19].

(3) The sink has no energy constraints, but the sensor nodes have severe constraints on their energy supply. So, we only take the energy of the sensor nodes into account. Compared with communications, the energy consumed in computations is very little, so we only estimate the energy consumed in communications.

2.2 LURP protocol

When the network is deployed, the sink broadcasts its location information among the entire network, and then the sensor nodes can send their data to the sink. A main difference between the LURP protocol and the traditional geographic routing protocol is that the latter one directly forwards the data to the sink hop by hop. But in the LURP protocol, the sensor nodes first forward the data to a small area near the sink, more specifically, to a certain node in the small area and then this node forwards the data to the sink. That is, the routing process is divided into two stages. At the first stage, data packets are forwarded from the sensors to a destination area. At the second stage, the data packets are forwarded to the sink in the destination area.

Figure 2 shows these two stages with an example. The rectangular area represents the monitored field. *S* represents a sensor node. At the beginning, the sink resides in the Virtual Center (*VC*). When *S* has a data packet to be sent, it first forwards the data packet to a certain node in the small area centered by *VC* according to some

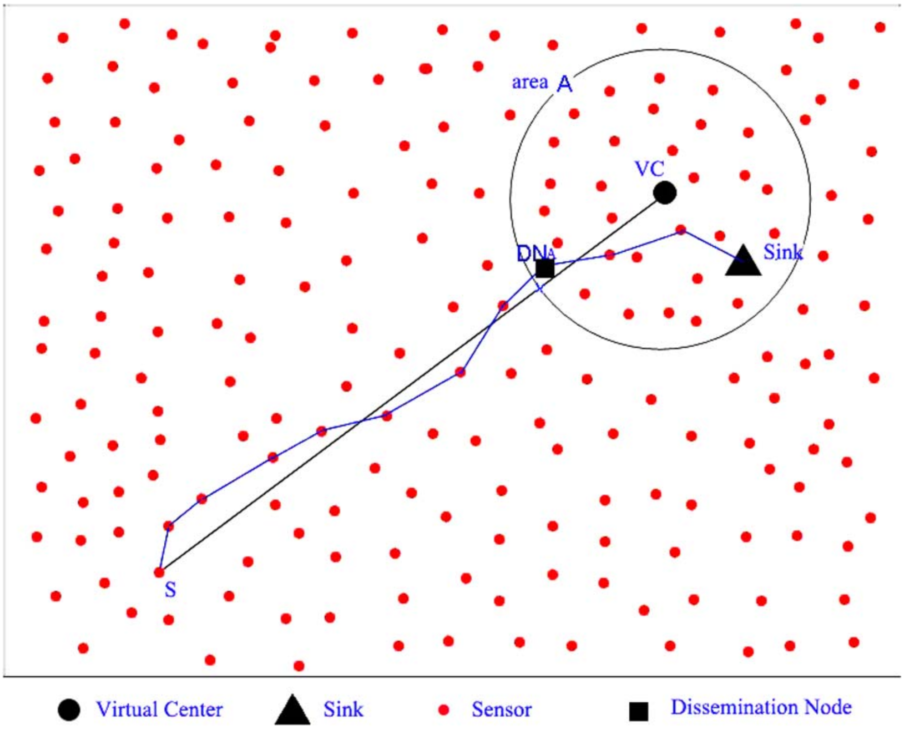


Fig. 2 Two stages of the LURP protocol

geocasting protocols, such as GFG [20], which is a protocol to forward data packet from one node to the nodes in a destination area. In this figure, that node is Dissemination Node (DN_A, where the subscript letter A denotes area A). When the data packet reaches DN_A, the second routing stage begins. The data packet is forwarded to the sink by some topology-based routing protocol instead of being flooded to the destination area as in GFG. More specifically, when the sink moves, as long as it is still within the destination area, it only needs to broadcast its location information within the destination area. During the broadcasting process, each sensor node in the area considers the node that sends the location information of the sink to itself as the next-hop node to the sink. After that, all the sensors in the area have built the routes to the sink.

As shown in Fig. 3, when the sink moves out of area A, it needs to broadcast its location information among the entire network, and accordingly, the new destination area, here area B, is built and the routing process repeats the way mentioned above. For example, node S will route the data packet along the new path shown in Fig. 4. There are three merits for doing so. First, when the sink moves, the sensors can keep continuous communications with the sink. Secondly, most of the time, the update of the sink's location information is restricted within a local area other than among the entire network, so it reduces the energy consumption in the network. Thirdly,

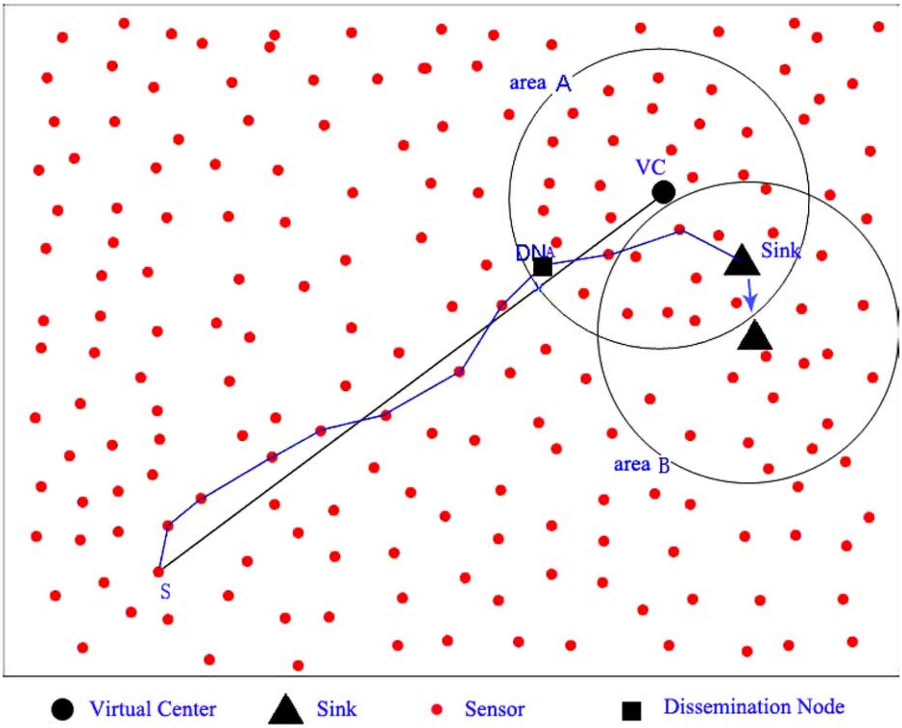


Fig. 3 Switch between two destination areas

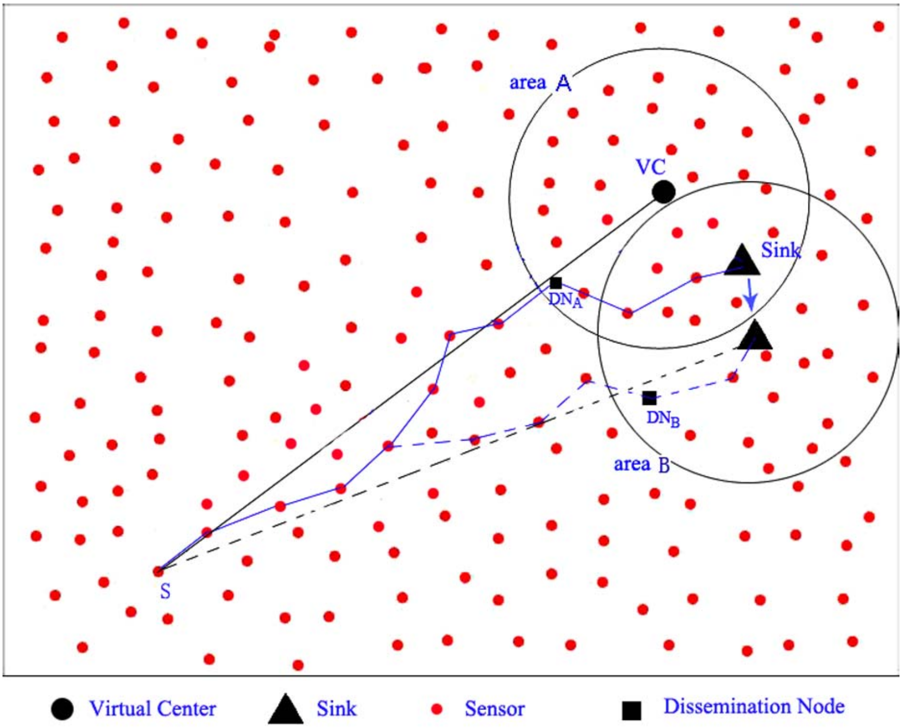


Fig. 4 The new routing path after the sink switches from area A to area B

it greatly decreases the collisions in wireless transmissions through confining the network area for updating the sink's location information.

In addition, when the sink moves out of a destination area, it rebuilds a new destination area and broadcasts its location information among the entire network. For example, as shown in Fig. 3, when the sink moves out of destination area A, it rebuilds a new destination area B. But, if we take the delay of broadcasting into account, some sensors which are far away from the sink may not receive the updated location information of the sink immediately. In this case, these sensors will still send their data packets to DN_A and then DN_A forwards the data packets to the sink as it is aware of the new location information of the sink. The pseudo code of the LURP protocol is shown in Fig. 5.

2.3 Remark on the size of the destination area

The size of the destination area can be expressed by its radius R, which is an important parameter in this protocol. On the one hand, if it is too small and the sink continuously switches from one destination area to another, the sink has to frequently update its location information among the entire network, so it will consume too much energy of the sensor nodes. On the other hand, if it is too large, the local update cost of the sink's location information will increase. It is easy to understand that the

The pseudo code from the sink perspective

(1) Construct a destination area using the sink itself as the Virtual Center *(VC)* and broadcast its location information to all the sensor nodes in the entire network. At the same time, sensor nodes in the destination area build the routes to the sink.
(2) Periodically check the routing topology at the sink until it changes due to mobility of the sink.
(3) **if** the sink is still within the destination area **then**
    (3.1) The sink broadcasts its location information in the destination area;
    (3.2) The sensor nodes in the destination area rebuild the routes to the sink;
    (3.3) **goto** (2)
    else goto (1). //The sink is outside the destination area

The pseudo code from the sensor nodes perspective

1) The packets from a data source are routed towards the point *VC* using the geographic routing protocol until a certain dissemination node *DN* is met in the destination area;
2) The packets are forwarded from *DN* to the sink using the topology-based routing in the destination area.

Fig. 5 The pseudo code of the LURP protocol

value of R is relevant to the size of the network. R should be set larger as the size of the network increases because the value of R determines the ratio between the times of updating the sink's location information among the entire network, and the times of updating the sink's location information within a local area. As the size of the network becomes larger, the cost of updating the location information among the entire network increases. In this case, the value of R should be set large enough to decrease the probability of flooding the sink's location information among the entire network.

2.4 ALURP protocol

The LURP protocol can save a large amount of energy through restricting the update range of the sink's location information compared with those protocols that need to flood the location information in the entire network. But once the radius R of the destination area has been set, the sink has to flood its location information to all the nodes in that area which may be not necessary most of the time. As shown in Fig. 6, area A is the destination area. When the sink moves within that area, it can further restrict the update range of its location information. The area can be reduced to a circle area using *VC* as its center and the distance between *VC* and the sink as its radius. We call this reduced area "adaptive area" because its radius can vary as the sink moves. In Fig. 6, the adaptive area is denoted by area B. We call the improved protocol with adaptive area Adaptive Local Update-based Routing Protocol (ALURP).

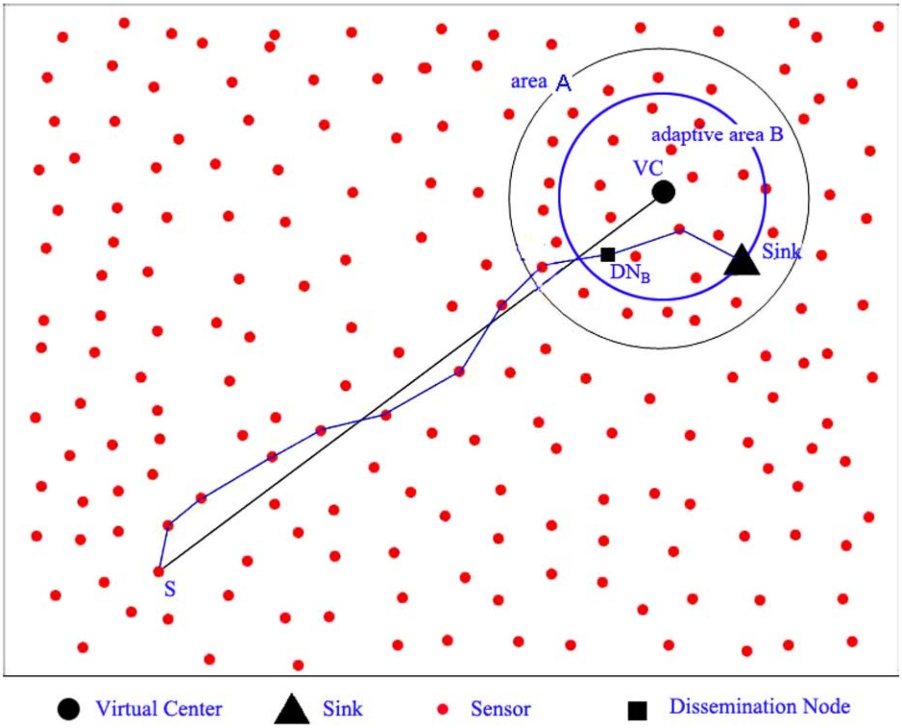


Fig. 6 An example of ALURP protocol

Initially, the sink is at the point *VC* and the radius of the adaptive area can be regarded as 0. As the sink moves farther away from the point *VC*, the radius of the adaptive area increases and the sink just needs to update its location information in this adaptive area when it changes routing topology due to its mobility. The data source can send the packet toward the point *VC* until the packet meets any node in the adaptive area who knows where the sink is. We also call that node Dissemination Node (*DN*) as defined in LURP. After that, the node *DN* forwards the packet to the sink.

However, when the sink moves in the direction closer toward the point *VC*, that is, when the radius of the adaptive area shrinks, a problem arises. The problem is that when the adaptive area is reduced, the nodes in the former adaptive area, which is bigger than the current one, but not in the current one still keeps the location information of the sink which has become obsolete. The reason is that the location of the sink has been changed but it does not inform the nodes outside the current adaptive area. So the data source will still send the packet to the former *DN* which will route the packet to a wrong place since the sink has moved to another place. Taking Fig. 7 as an illustration, before the sink moves closer toward the point *VC*, the adaptive area is area B and the node DN_B is the *DN* of area B. But after the sink's movement, the adaptive area has been reduced which is denoted by area C. Then the sink only updates its location information in area C while from the perspective of

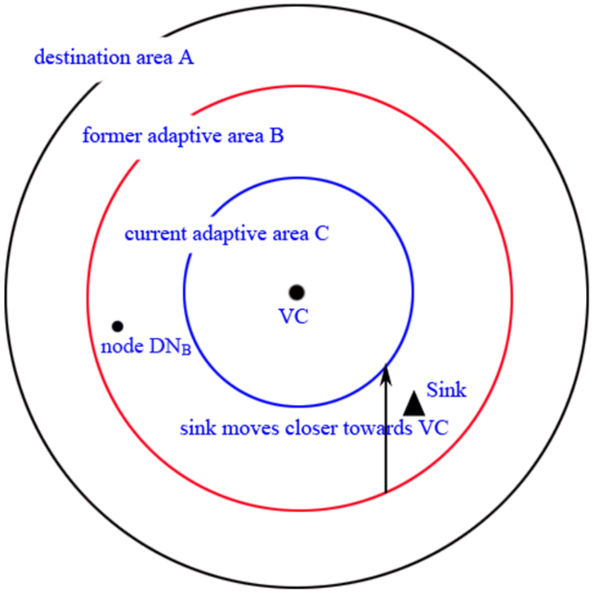


Fig. 7 The problem occurs when the sink moves closer towards *VC*

node DN_B, it does not know the new location of the sink and will route the packet to the location it reserved formerly. To resolve this problem, the sink has to inform the nodes in the former adaptive area, but not those in the current adaptive area to flush the location information of the sink. Compared with LURP, which needs to update the location information of the sink into the whole destination area, this mechanism further decreases the update range and it will further reduce the energy consumption. The pseudo code of the ALURP protocol is shown in Fig. 8.

3 Performance analysis

3.1 Analysis model

In this section, we evaluate the performance of ALURP through mathematical analysis. Similar to the analysis model in [17], we consider a squared sensing field whose side length is L and in which N sensor nodes are randomly distributed and the sensors send their sensed data to the sink. Assume that the velocity of the sink is v and its location will change m times during the period of time T. The radius of the destination area is denoted by R and the period of time consumed by the sink to move out of a destination area is denoted by t. We also assume that the communication overhead to flood within an area is proportional to the number of sensor nodes in the area.

3.2 Cost of updating the location information of the sink

During the period of time T, assume that node S has data packets to be sent to the sink. As the sink's location has changed m times, the maximum location update cost

The pseudo code from the sink perspective

---

(1) Construct a destination area using the sink itself as the Virtual Center (*VC*) and broadcast its location information to all the nodes in the entire network. At the same time, the nodes in the destination area build the routes to the sink.
(2) Periodically check the routing topology at the sink until it changes due to mobility of the sink.
(3) **if** the sink moves out of the destination area **then**

 goto (1);

(4) Construct a new adaptive area and broadcast its location to the nodes in the adaptive area and the nodes in the adaptive area rebuild the routes to the sink;
(5) if the sink moves in the direction closer toward the point *VC* **then**

The sink informs the nodes in the former adaptive area but not in the current adaptive area to flush the topology information of the sink;

(6) **goto** (2). //The sink is still within the destination area

The pseudo code from the sensor nodes perspective

---

(1) The packets from a data source are routed toward the point *VC* using the geographic routing protocol until a certain dissemination node *DN* is met in the adaptive area;
(2) The packets are forwarded from *DN* to the sink using the topology-based routing in the destination area.

Fig. 8 The pseudo code of the ALURP protocol

is:

$$E_1 = mnh + \left(\frac{T}{t}\right) \times Nh, \tag{1}$$

where n is the total number of sensors in the destination area and it can be calculated by the formula below:

$$n = \pi R^2 \times \frac{N}{L^2}. \tag{2}$$

$\frac{N}{L^2}$ in (2) denotes the density of the network. In (1), nh is the maximum local update cost of the sink's location information as actually the sink just needs to update the location information to the nodes in the adaptive area where the nodes in that area is less than n. Besides the local update cost, the total cost for updating the sink's location information includes $(\frac{T}{t}) \times Nh$ which is the cost for updating the sink's location information among the entire network as the sink switches from one destination area to another for $(\frac{T}{t})$ times. When putting (2) into (1), we derive (3):

$$E_1 = m\frac{\pi R^2}{L^2} \times Nh + \left(\frac{T}{t}\right) \times Nh. \tag{3}$$

It is easy to know that the value of t is related to the size of the destination area which is denoted by R and the velocity of the sink which is denoted by v. The larger the destination area is and the smaller the sink's velocity is, the longer period of time the sink takes to move out of a destination area. And the value of m is related to the value of T and the value of v. The larger the values of T and v are, the larger the value of m is. For the simplicity of analysis, we assume that the value of t is proportional to the value of R and is inverse proportional to the value of v, and the value of m is proportional to the values of T and v. Based on these considerations, we have (4) and (5):

$$t = \alpha \times \frac{R}{v}, \tag{4}$$

$$m = \beta \times T v, \tag{5}$$

where α and β are two constants. When putting (4) and (5) into (3), we get (6):

$$E_1 = NhTv\left(\frac{\beta\pi R^2}{L^2} + \frac{1}{\alpha R}\right). \tag{6}$$

It is easy to get that when:

$$R = \sqrt[3]{\frac{L^2}{2\pi\alpha\beta}}, \tag{7}$$

E_1 reaches the minimum, which is consistent with the remark in Sect. 2.3. It shows that there exists an appropriate value of R that can minimize the location update cost. We can also conclude that the larger the size of the network is, the larger the value of R is. On the other hand, for the Flooding-based Location Update Protocol (FLUP) [15, 16], as every time the location of the sink changes, the sink needs to broadcast its location information to all the sensors among the entire network, so the total update cost for FLUP is set as:

$$E_2 = mNh. \tag{8}$$

This is so because its location changes m times during the period of time T and every time the overhead is Nh. While replacing m of (5) into (8), we have:

$$E_2 = \beta T v N h. \tag{9}$$

To compare ALURP with FLUP when putting (7) into (3), we have:

$$\frac{E_1}{E_2} = \left(2^{1/3} + 2^{-2/3}\right)\left(\frac{\pi}{L^2\alpha^2\beta^2}\right)^{1/3}. \tag{10}$$

As π, α and β are all constants, let α and β are equal to 1 for a simple numerical illustration. Table 1 demonstrates that the cost for the mobile sink to update the location information using ALURP is much less compared with FLUP and the larger the value of L is, the smaller the value of $\frac{E_1}{E_2}$ is. Therefore, we can draw a conclusion that ALURP greatly decreases the energy cost in large-scale networks compared with FLUP.

Table 1 A numerical illustration of $\frac{E_1}{E_2}$

L (m)	10	50	100	500
$\frac{E_1}{E_2}$	0.596	0.204	0.128	0.0439

Table 2 The simulation parameters

Communication radius	30 m
Density of the network	$0.003/\text{m}^2$
Velocity of the mobile sink	10 m/s
Size of the data packet	525 Bytes
Eelec	50 nj/bit
ε_{fs}	$10\ \text{pj/bit/m}^2$

4 Simulation studies

From the performance analysis in Sect. 3, it is clear that the cost for the mobile sink to update the location information using ALURP is decreased compared with FLUP. However, the two-stages routing mechanism used by ALURP makes its path length a little longer than that of FLUP. When the destination area becomes smaller, the path length using ALURP is relatively shorter; while the destination area gets larger, it will be longer. Meanwhile, the size of the destination area is related to the factor mentioned in Sect. 2.3. In this section, we compare the energy consumption among ALURP, LURP, and FLUP, and the effect of the values of R and v in ALURP through simulations.

4.1 Simulation model

We compare ALURP with FLUP and LURP in simulations using C++. The simulation scenario is within a squared sensing field of side length L, in which N sensor nodes are randomly placed, and a mobile sink moves with fixed velocity to one of its neighboring nodes at random. After arriving at the node, the sink randomly chooses another neighboring node toward which it moves. There are some sensor nodes that send data packets to the sink. To simplify the model, we suppose that the length of a data packet is equal to that of the packet for updating the sink's location information. The energy consumed by communications obeys the First Order Radio energy model [5]. The simulation parameters in the model are shown in Table 2.

4.2 Simulation results

Figure 9 shows the average energy consumption of a data packet sent to the sink using ALURP, LURP, and FLUP, respectively, considering only the communication cost instead of that of updating the location information of the sink, when R is 100 m (which is the same in Fig. 10 and Fig. 11). It is clear that when the size of the network is enlarged from 1000 m $\times$ 1000 m to 5000 m $\times$ 5000 m, the energy consumption using ALURP and LURP are both slightly more than that of FLUP, although it is not

Fig. 9 The comparison of communication energy

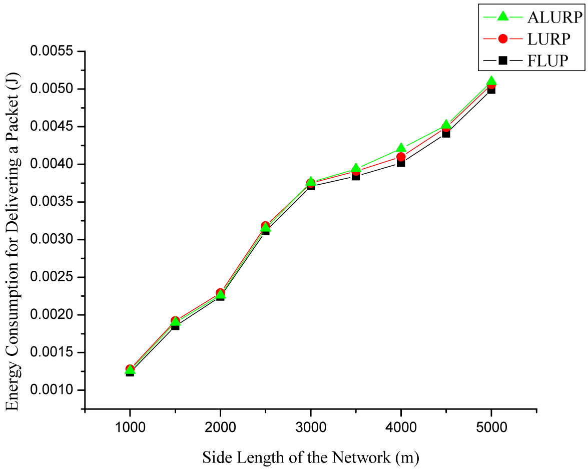
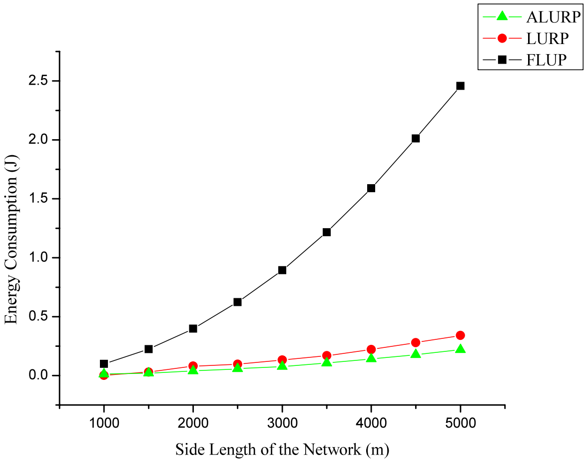


Fig. 10 Energy comparison of updating the location information of the mobile sink

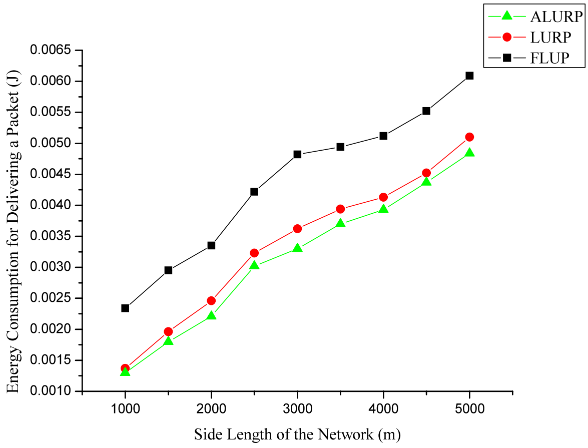


very notable. The reason is that the length of the path of ALURP is a little longer than that of FLUP.

Figure 9 compares the energy consumption for updating the location information of the sink among ALURP, LURP, and FLUP, during the period of time from the sink's construction of a new destination area moving out of the area. As pointed out in Sect. 3.2, updating the location information of the sink using ALURP consumes much less energy compared with that using FLUP. This becomes more significant as the size of the network increases. Compared with LURP, ALURP also consumes less energy since the update area becomes smaller.

Figure 11 shows the comparison of energy consumption when the cost of communications and the cost of updating the location information of the sink are both considered. It can be seen that when the size of the network is enlarged from 1000 m × 1000 m to 5000 m × 5000 m, the energy consumption of these three solutions tend to increase because of the increase of the average distance between sensor

Fig. 11 The total energy comparison among ALURP, LURP and FLUP



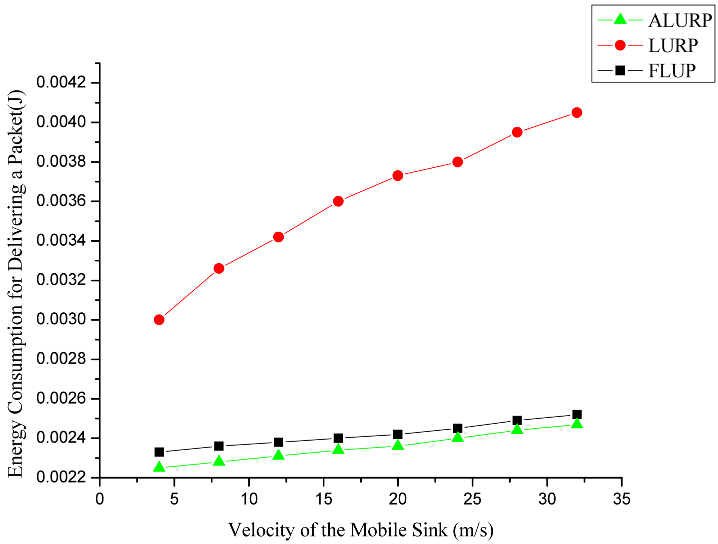


Fig. 12 Energy consumption over the velocity of the mobile sink

nodes and the sink. However, it is obvious that the energy consumption using ALURP is less than that using either LURP or FLUP with the same size of the network.

Figure 12 shows the average energy consumption of a data packet sent to the sink (including energy of communication and updating the location information of the sink), as the velocity of the mobile sink varies, in the network size of 2000 m $\times$ 2000 m. It can be seen that the energy consumption of all of the protocols tends to increase as the velocity of the sink increases. The reason is that the frequency of updating the location information of the sink increases as the velocity of the sink increases. That can also be seen in formulae (6) and (9) in Sect. 3, which show that the energy consumption is in proportion to the velocity of the sink.

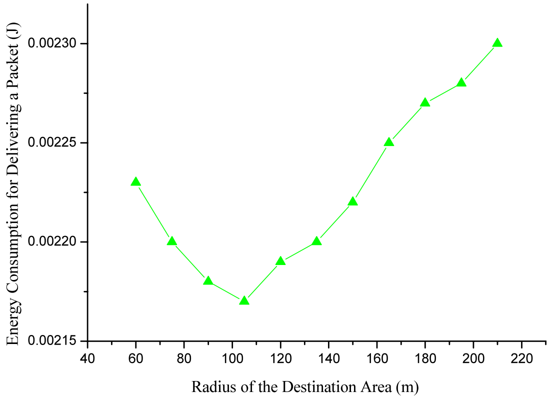


Fig. 13 Energy consumption over the radius of the destination area

However, energy consumed in FLUP obviously exceeds that of ALURP and LURP. Compared with LURP, the energy consumed by ALURP is also smaller than that of LURP. So, ALURP is more effective in the environments where the velocity of the sink is high compared with FLUP.

Figure 13 shows the average energy consumption of a data packet sent to the sink as R varies in the network size of 2000 m × 2000 m. When R increases gradually from 60 m, energy consumption tends to decrease. And when R is around 105 m, the energy consumption in ALURP reaches the minimum. After that, it starts to increase as R becomes larger. The reason is that when the value of R is very small, the sink will quickly move out of the destination area, resulting in the sink's location information being updated among the entire network. On the other hand, if the value of R is too large, the cost of updating the location information of the sink within a local area is larger. So as analyzed in Sect. 2.3, there exists a balanced point of the value of R, which is 105 m as shown in Fig. 13.

Figure 14 shows the comparison between ALURP and LURP. Before making the comparison, we first give the definition of "sojourn-hops." The "sojourn-hops" denotes how many hops the sink needs to move out of the destination area. For example, when the sojourn-hops are 10, it means that when a new destination area has been built, the sink has to visit at least 10 nodes in that area and then move out of the area. To do this, we simply modify the moving pattern of the sink. We set a counter on the sink. When a new destination area is built, the value of the counter is set to be 0. When the sink arrives at a node, the value will be added by 1. If the value does not exceed the value of sojourn-hops, the sink only selects the nodes in the destination area as the next moving target. From the figure, we can see that energy consumption using ALURP is less than that using LURP and when the value of sojourn-hops becomes larger, the differences between them become more obvious. The reason is that the value of sojourn-hops implies the "stabilization" of the sink. There are some applications where the moving pattern of the sink is "relatively" stable, i.e., the sink keeps moving but its track is in a relatively small range for a long time. As the example shows in Fig. 1, the patrol will always do some local manipulations such as searching for poachers. In this circumstance, the patrol will sojourn in a small area for a long

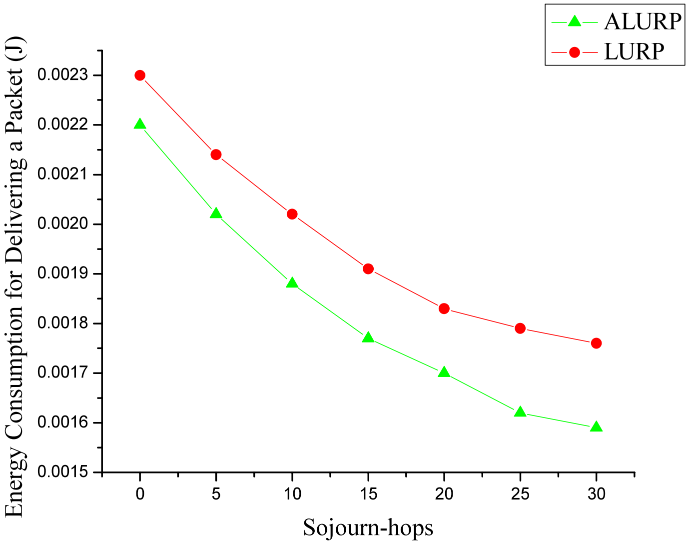


Fig. 14 Energy consumption over the sojourn-hops

time. So the ALURP is more effective than LURP especially in the circumstance where the moving pattern of the sink is relative stable.

5 Conclusions

In this paper, we proposed an adaptive local update-based routing protocol in wireless sensor networks with a mobile sink. The proposed protocol, ALURP, greatly saves the energy for wireless sensor networks and makes the sink keep continuous communications with sensor nodes by confining the destination area within a local area for updating the sink' location information as the sink moves. Compared with protocols that need to continuously propagate the sink's location information among the entire network, ALURP greatly decreases the cost of updating the sink's location information and decreases the collisions in wireless transmissions. In addition, when the sink moves out of its destination area, those sensors which are far away from the sink can still communicate with the sink without receiving the new location information of the sink. Therefore, the proposed protocol reduces the delay and energy consumption, and thus it is suitable for large-scale and delay-sensitive wireless sensor networks. Theoretical analysis and simulation studies show that ALURP is efficient in resolving the above issues.

Our future work is to consider multiple mobile sinks in wireless sensor networks. According to different applications, it can be further divided into two scenarios. Under the first scenario, the sensors only need to disseminate their data packets to any one of the mobile sinks and we will consider how to balance the energy consumption among the mobile sinks. Under the second scenario, the sensors need to disseminate their data packets to all the sinks, and we will consider how to efficiently construct the data dissemination tree when the sensors send their data packets to various mobile sinks.

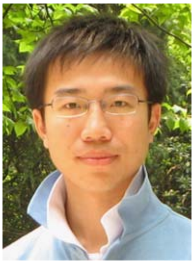
Acknowledgements This work has been partially supported by the Hunan Provincial Natural Science Foundation of China under Grant No. 07JJ1010, the National Natural Science Foundation of China under Grant Nos. 60740440032 and 60533040, the Hong Kong CERG Grant No. 9041129 (CityU 113906) and CityU Strategic Grant No. 7002102, and the National High-Tech Research and Development Plan of China (863 Plan) under Grant Nos. 2006AA01Z202 and 2006AA01Z199.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. Comput Netw 38(4):393–422
2. Estrin D, Govindan R, Heidemann J, Kumar S (1999) Next century challenges: scalable coordination in sensor networks. In: Proceedings of ACM MobiCom 1999, Seattle, WA. ACM Press, New York. pp 263–270
3. Pottie GJ, Kaiser WJ (2000) Wireless integrated network sensors. Commun ACM 43(5):51–58
4. Heinzelman WR, Kulik J, Balakrishnan H (1999) Adaptive protocols for information dissemination in wireless sensor networks. In: Proceedings of ACM MobiCom 1999, Seattle, WA. ACM Press, New York, pp 174–185
5. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of 33rd annual Hawaii international conference on system sciences, Hawaii, USA, pp 20–30
6. Intanagonwiwat C, Govindan R, Estrin D (2000) Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceedings of ACM MobiCom 2000, Boston, MA, pp 56–67
7. Wang T, Wang G, Guo M, Jia W (2008) Designing hash-area-based data dissemination protocol in wireless sensor networks. J Central South Univ Technol (English Edition), to appear
8. Jia W, Wang T, Wang G, Guo M (2007) Hole avoiding in advance routing in wireless sensor networks. In: Proceedings of the 2007 IEEE wireless communications & networking conference (WCNC 2007), Hong Kong, pp 3522–3526
9. Wang G, Cao J, Wang H, Guo M (2007) Polynomial regression for data gathering in environmental monitoring applications. In: Proceedings of the 2007 IEEE global telecommunications conference (GLOBECOM 2007), Washington, DC, USA, pp 1307–1311
10. Wang G, Wang H, Cao J, Guo M (2007) Energy-efficient dual prediction-based data gathering for environmental monitoring applications. In: Proceedings of the 2007 IEEE wireless communications & networking conference (WCNC 2007), Hong Kong, pp 3516–3521
11. LaMarca A, Brunette W, Koizumi D, Lease M, Sigurdsson S, Sikorski K (2002) Making sensor networks practical with robots. In: LNCS, vol 2414. Springer, New York, pp 152–166
12. Al-Karaki JN, Kamal AE (2004) Routing techniques in wireless sensor networks: a survey. IEEE Wirel Commun 11(6):6–28
13. Gandham SR, Dawande M, Prakash R, Venkatesan S (2003) Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In: Proceedings of IEEE GLOBECOM 2003, San Francisco, CA, pp 377–381
14. Shah D, Shakkottai S (2007) Oblivious routing with mobile fusion centers over a sensor network. In: Proceedings of IEEE INFOCOM 2007, Anchorage, AK, USA, pp 1541–1549
15. Coffin D, Hook DV, McGarry S, Kolek S (2000) Declarative ad-hoc sensor networking. In: SPIE integrated command environments, pp 109–120
16. Ye F, Zhong G, Lu S, Zhang L (2005) Gradient broadcast: a robust data delivery protocol for large scale sensor networks. ACM Wirel Netw (WINET) 11(2):285–298
17. Ye F, Luo H, Cheng J, Lu S, Zhang L (2002) A two-tier data dissemination model for large-scale wireless sensor networks. In: Proceedings of ACM MobiCom 2002, Atlanta, GA, USA, pp 148–159
18. Wang G, Wang T, Jia W, Guo M, Chen HH, Guizani M (2007) Local update-based routing protocol in wireless sensor networks with mobile sinks. In: Proceedings of the 2007 IEEE international conference on communications (ICC 2007), Glasgow, Scotland, UK, pp 3094–3099
19. Niculescu D (2004) Positioning in ad hoc sensor networks. IEEE Netw 18(4):24–29

20. Seada K, Helmy A (2004) Efficient geocasting with perfect delivery in wireless networks. In: Proceedings of IEEE WCNC 2004, Atlanta, GA, pp 2551–2556

Guojun Wang received his BSc degree in Geophysics, MSc degree in Computer Science, and PhD degree in Computer Science, from the Central South University, in 1992, 1996, 2002, respectively. He is currently a Professor (2005–) in the Department of Computer Science and Technology, the Central South University, Changsha, P. R. China, 410083. He is also a vice head of the Department, and the director of the Institute of Trusted Computing, formerly the Institute of Mobile Computing, in the University. He was a Research Fellow (2003–2005) in the Hong Kong Polytechnic University, Hong Kong. He was a Visiting Researcher (2006–2007) in the University of Aizu, Japan. His research interests include mobile computing, trusted computing, and software engineering. He has published more than 120 technical papers and books/chapters in the above areas. He is an associate editor or an editorial board member of some international journals including Security and Communication Networks, Journal of Security and Telecommunications, Journal of Computer Systems, Networking, and Communications, and International Journal of Multimedia and Ubiquitous Engineering. He has served as a reviewer for many international journals such as IEEE TC, IEEE TPDS, IEEE TWC, and IEEE TVT. He has also served as a program chair, program vice-chair, publication chair, publicity chair, session chair, and program committee member for many international conferences such as GLOBECOM, ICC, WCNC, AINA, HPCC, IWCMC, EUC, ISPA, ICYCS, and TrustCom. He is a senior member of the China Computer Federation (2005–), the chair of the YOCSEF Changsha of the China Computer Federation (2007–2008), a member of several Technical Committees of China Computer Federation, including Fault Tolerant Computing, Pervasive Computing, Software Engineering, E-Government and Office Automation (2007–), a YOCSEF member of the China Computer Federation (2001–), and a member of the Hunan Provincial Association of Computers (1994–). He was listed in Marquis Who's Who in the World (2007).



Tian Wang received his BSc and MSc degrees in Computer Science from the Central South University in 2004 and 2007, respectively. Currently, he is a research assistant in the City University of Hong Kong. His research interests include designing energy-efficient routing protocols in wireless sensor networks.

Weijia Jia is an Associate Professor of Computer Science at the City University of Hong Kong (CityU). He received BSc, MSc, from Center South University, China and M. Applied Sci. and PhD from Polytechnic Faculty of Mons, Belgium, all in Computer Science. He joined German National Research Center for Information Science (GMD) in Bonn (St. Augustine) from 1993 to 1995 as post-doc. In Aug. 1995, he joined Department of Computer Science, CityU, as an assistant professor. In 2006, he was awarded HK$11 millions from the Innovation & Technology Fund of the HKSAR Government for a project entitled “Digital Network Platform Technology and Equipment for Ubiquitous Communications”. The project is the No. 1 in size at CityU in 2006 with intention of design and implementation of ubiquitous communication platform and soft switch to harness Internet with 3G, WiFi,WiMAX, ad-hoc and PSTN networks. His research interests include wireless communication and networks, distributed systems, multicast and (pioneer) anycast QoS routing protocols for Internet. In these fields, he has about 300 publications in international journals, books/chapters and refereed international conference proceedings. He (with Wanlei Zhou) has published a book “Distributed Network Systems” by Springer where the book contains extensive research materials and implementation examples. He has received the best paper award in a prestige (IEEE) conference. He (with J. Chen et al.) has proposed an improved algorithm for well-known Vertex Cover and set-packing NP-hard problems with time bounds of O($kn + 1.2852k$) and O($(5.7k)kn$) respectively. The both results stand on the current best time-bound (as at Oct. 2006) for the fixed-parameterized intractable problems. He is the Chair Professor of Central South University, Changsha, China, Guest Professor of University of Science and Technology of China and Jinan University, Guangzhou, China. He has served as the editor and guest editor for international journals and PC chairs and PC members/keynote speakers for various IEEE international conferences. He is the member of IEEE and has been listed in Marquis Who's Who (VIP) in the World (2000-2006).

Minyi Guo received his PhD degree in Computer Science from the University of Tsukuba, Japan. Before 2000, Dr. Guo had been a research scientist of NEC Corp., Japan. He is now a chair professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He was also a visiting professor of Georgia State University, USA, Hong Kong Polytechnic University, University of Hong Kong, National Sun Yet-Sen University in Taiwan, China, University of Waterloo, Canada and University of New South Wales, Australia. He is also a Professor of the University of Aizu, Japan. Dr. Guo has published more than 150 research papers in international journals and conferences. Dr. Guo has served as general chair, program committee or organizing committee chair for many international conferences. He is the founder of International Conference on Parallel and Distributed Processing and Applications (ISPA), and International Conference on Embedded and Ubiquitous Computing (EUC). He is the editor-in-chief of the Journal of Embedded Systems. He is also on the editorial board of Journal of Pervasive Computing and Communications, International Journal of High Performance Computing and Networking, Journal of Embedded Computing, Journal of Parallel and Distributed Scientific and Engineering Computing, and International Journal of Computer and Applications. Dr. Guo's research interests include parallel and distributed processing, parallelizing compilers, Pervasive Computing, Embedded Software Optimization, molecular computing and software engineering. He is a senior member of IEEE, a member of the ACM, IPSJ and IEICE.

Jie Li received the B.E. degree in computer science from Zhejiang University, Hangzhou, China, in 1982, and the M.E. degree in electronic engineering and communication systems from China Academy of Posts and Telecommunications, Beijing, China, in 1985. He received the Dr. Eng. degree from the University of Electro-Communications, Tokyo, Japan in 1993. From 1985 to 1989, he was a research engineer in China Academy of Posts and Telecommunications, Beijing. From April 1993, he has been with the Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Japan, where he is a Professor. His current research interests are in mobile and ubiquitous multimedia computing and networking, OS, network security, distributed and parallel computing, modeling and performance evaluation of information systems, and their applications. He received the best paper award from IEEE NAECON '97. He is a senior member of IEEE and ACM, and a member of IPSJ. He has served as a secretary for Study Group on System Evaluation of the Information Processing Society of Japan (IPSJ), and on the many editorial boards such as the IPSJ (Information Processing Society of Japan) Journal, IEEE Transactions on Vehicular Technology, and the International Journal of High Performance Computing and Networking, etc. He is also serving on Steering Committees of the SIG of System EVAluation (EVA) of IPSJ, the SIG of DataBase System (DBS) of IPSJ, and the SIG of MoBiLe computing and ubiquitous communications of IPSJ. He has also served on the program committees for several international conferences such as IEEE ICDCS, IEEE INFOCOM, IEEE GLOBECOM, and IEEE MASS.