Highly Scalable and Efficient Publish/Subscribe Protocols Using Geographic Information for Wireless Sensor Networks

Yuan Zheng , Jiannong Cao
Internet and Mobile Laboratory, Department of Computing
Hong Kong Polytechnic University,Kowloon, Hong Kong
{csyuazheng, csjcao}@comp.polyu.edu.hk

Abstract

Publish/Subscribe (Pub/Sub) systems have been used in traditional distributed computing applications to provide effective and efficient event services. Recently, the design of Pub/Sub systems are proposed for wireless sensor networks (WSNs). Some of the proposed Pub/Sub protocols address the issue of establishing event delivery path with low routing cost by using geographic information. However, they have not considered how to reduce redundant event delivery, which may cause high resource consumption and poor system scalability. In this paper, we propose protocols which address both issues. More specifically, our protocols use geographic information to save routing cost, and meanwhile reduce redundant event delivery by letting the subscribers share the event delivery paths. Three highly scalable and efficient Pub/Sub protocols for WSNs, namely Shortest Delivery Path (SDP), Shortest Delivery Path with Lower Delivery Overhead (SDP-LDO), and Lowest Delivery Overhead (LDO), are designed to achieve different performance goals. The results of theoretical and experimental evaluation show that the proposed protocols can significantly improve the resource efficiency and scalability of a Pub/Sub system compared to the previous solutions. Our protocols can also be used to achieve a good tradeoff between the costs of event subscribing and event delivery.

1. INTRODUCTION

A wireless sensor network (WSN) is composed of a collection of sensor nodes attached with sensing, computation, and wireless communication devices. The sensor nodes can cooperatively collect sensory data and transmit them to users of the applications over the WSN [1]. In the recent years, WSNs have attracted a lot of attentions from both industrial and academic communities. Many WSN-based applications have been explored in different areas, such as environment monitoring, traffic management, intelligent building management, battle-field surveillance, etc [2].

One of the important tasks of WSNs is to provide highly efficient and scalable event service. Pub/Sub systems [3] have been widely used in traditional distributed systems to provide event services. A Pub/Sub system encapsulates data into events and provides the services of event publications and subscriptions for asynchronous data exchange among the system entities. Several Pub/Sub systems have been developed in recent years in the context of WSNs [4]-[9].

Most existing Pub/Sub systems [4]-[8] employ flooding for event subscription and publication, which incurs high resource consumption and suffers from poor scalability. To avoid flooding, Diffusion Filters (DF) [9] uses geographic energy-aware routing (GEAR) [10] to establish independent event delivery paths to the subscribers with resource efficiency. However, DF did not consider the issue of reducing redundant event delivery, which may cause high cost and poor system scalability.

In this paper, we propose three pub/sub protocols which allow the subscribers to share the event delivery paths according to subscribers' geographic locations and consequently, effectively reduce the redundant event delivery. The three protocols, namely Shortest Delivery Path (SDP), Shortest Delivery Path with Lower Delivery Overhead (SDP-LDO), and Lowest Delivery Overhead (LDO), are designed to achieve different performance objectives. The results of theoretical and experimental evaluation show that the proposed protocols can significantly improve the resource efficiency and system scalability compared with the existing solutions. Our protocols can also achieve a good tradeoff between event subscribing overhead and event delivery overhead.

The remaining part of this paper is organized as follows. Section 2 discusses existing work on pub/sub systems in wireless sensor networks. Section 3 describes the system model and our three Pub/Sub protocols. Section 4 presents the theoretical analysis and experimental evaluation of the three proposed protocols. Finally, Section 5 concludes the paper with the discussion of our future work.

2. RELATED WORK

Existing works have addressed the Pub/Sub system design and deployment in WSNs. The related protocols and strategies are also proposed. However, no work has addressed how to reduce redundant event delivery to improve the system performance and save resources for the multiple subscribers.

Low-level naming [4] is the first Pub/Sub system developed for wireless sensor networks. Low-level naming uses the Directed Diffusion (DD) [11] routing protocol for interest dissemination and gradient forwarding. However, DD uses subscription flooding, and therefore, causes very high resource consumption. Mires [5] system employs advertisement-driven mechanism, which also causes the high cost due to the advertisements and subscriptions flooding. DF[9], the extension of Low-level naming, uses GEAR[10] to deliver events rather than simply flooding as in Low-level naming. However, DF has not considered how to reduce the redundant subscription and event delivery, which incurs high resource consumption. In addition, DF allows only the sink node to disseminate subscriptions, which cannot satisfy the applications with distributed data requests.

Some previous works proposed the event delivery strategies to improve the reliability of Pub/Sub systems. The work in [6] uses a semi-probabilistic approach to forward events to randomly selected neighbour nodes. However, this approach requires all nodes to participate in the event delivery, which causes high resource cost. The system in [7] establishes multiple broker trees for event subscription and publication. The large number of the event delivery trees considerably increases the control overhead. The work [8] uses a local repairing method to recover the broken delivery path incurred by exhausted nodes. It also uses flooding to establish subscription trees and consequently, suffers from the high cost. Unfortunately, neither of the above strategies investigates how to let the nodes share event delivery paths to reduce the redundant event transmission.

3. SYSTEM MODEL

In this section, we discuss the system model used for our protocol design. First, we describe the cluster-based network model and based on the model, we present the broker-based Pub/Sub system architecture. Then, we describe the event model in our work.

A. Network Model

Following the WSN deploying models used in GAF[12], we assume that the wireless sensor network is deployed in a rectangular region, which is divided into grids of the identical size. The locations of the sensor nodes in the rectangular region obey the randomly independent and identical distribution. All the nodes are aware of their geographical locations. We assume that every grid has a unique ID which is composed of X and Y coordinates of the grid. A sensor node in a grid can acquire its grid id by using the information of its location, the size of the grid, and the border of the deploying region. All these information can be pre-loaded into the node before deployment or disseminated to the node after deployment.

The sensor nodes are grouped into clusters. The sensor nodes in one grid form a cluster and a sensor node belongs to only one cluster. Every cluster has an ID identical to its grid ID. There is only one cluster head in a cluster and the ID of the cluster head is identical to its cluster's ID. The member nodes in a cluster can communicate directly with their cluster head. In addition, a cluster head can directly communicate with their counterparts in the neighbour clusters. There are many papers on clustering [14] and we will not discuss them in this paper.

B. Broker-based Pub/Sub Model

We employ the broker-based pub/sub architecture where only the broker nodes disseminate subscriptions and deliver events. All cluster heads in the WSN act as the brokers by running the pre-loaded broker program. Thus, a broker's ID is just the cluster head ID. Client programs can run on any node for event publications and subscriptions. A client has a system-wide unique ID and can communicate with the broker in the same cluster. Throughout the paper, we call a broker connected with a publisher as a *publisher hosting broker* (PHB) and a broker connected with a subscriber as a *subscriber hosting broker* (SHB).

C. Event and Subscription Model

An event in our system is defined by a 3-tuple as follows.

Event = <Region, Key, Value>, where

- Region is the source region of the event, denoted by the publisher's grid.
- Key is the sensory type described by the event, such as temperature, light, pressure, etc.
- Value is the sensory result, such as 80 degree, 1 lumen, 1 Pa, etc.

A subscription in our system can be described by a 4-tuple <region, key, value, operation>. A broker handles only those events that can satisfy a subscription, meaning that the source region and the key of the received event are the same as the satisfied subscription, and the relation between the values of the event and the subscription satisfies the 'operation' in the subscription. For an example, event <A, temperature, 5 > can satisfy the subscription <A, temperature, 10, '<'>. Thus, a subscription received by a broker will be used as a filter to find the desired events.

4. HIGHLY EFFICIENT AND SCALABLE PUB/SUB PROTOCOLS

In this section, we describes three protocols, including the shortest delivery path (SDP) protocol, the shortest delivery path with lower delivery overhead (SDP-LDO) protocol, and the lowest delivery overhead (LDO) protocol. First, we describe the preliminaries of the protocol design, including the subscription-reverse-path routing mechanism used in the Pub/Sub protocols. Then, we describe the three different proposed routing algorithms. In the designing of the three protocols, we aim at enabling different subscribers to share the event delivery paths as much as possible. In this way, the event delivery on the common part of the paths can serve all of the subscribers simultaneously and, therefore, the redundant event delivery can be eliminated. Our protocols adopt the subscription-reverse-path mechanism, which is discussed in the following sub-section. Thus, the paths for subscription dissemination determine the final event delivery paths and the SHBs should use overlapping paths as much as possible to propagate their subscriptions. According to the geographic information of the desired event publisher, the newly joining SHB can effectively discover the existing propagation path, which may already become a delivery path.

Then, the SHB can take full advantages of the existing path by appending itself to the existing path. We discuss the details in this section.

A. *Preliminaries*

The protocols proposed in this paper employ the subscription-reverse-path mechanism to perform the event delivery. First, a SHB propagates a subscription from a client to the target PHB via one or more propagation paths. An intermediate broker on a propagation path records the last-hop and the next-hop brokers in its local routing table. Then, the target PHB selects one propagation path and delivers the corresponding events reversely along the path to the SHB. Since all the intermediate brokers on the path have record the last-hop and next-hop brokers, they can properly forward the received events.

For the simplicity of the discussion, we first establish a rectangular coordinate system. Assuming that a PHB is located at the center of its grid, we set the PHB's location to be the origin and add four lines passing the origin point. One of the four lines is a vertical line, and then every acute angle generated by two neighbouring lines is 45 degrees, as shown in Fig.1. Starting from the vertical line, we name the four lines as X-Axis, leading-diagonal (L-diagonal), Y-Axis, and auxiliary-diagonal (A-diagonal), in the clockwise direction (Fig. 1). The grids passed by axes and diagonals are named axis and diagonal grids, respectively. The two axes together with the two diagonals divide the whole region into 8 parts, named as Part I to Part VIII in the clockwise direction, excluding the grids passed by the axis and diagonal lines. We also define the following directions.

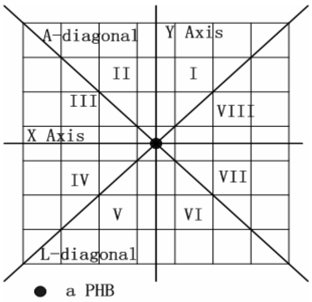


Fig.1 8 parts of a deploying region

- X-direction: the direction parallel to X-Axis and towards the target PHB.
- Y-direction: the direction parallel to Y-Axis and towards the target PHB.
- L-direction: the direction parallel to L-diagonal and towards the target PHB
- A-direction: the direction parallel to A-diagonal and towards the target PHB.

B. *The Shortest Delivery Path (SDP) protocol*

Using the SDP protocol, a SHB greedily propagates a subscription towards the target PHB to establish the shortest subscription delivery path which is the reverse shortest event delivery path. First, the SHB and the intermediate brokers greedily forward the subscriptions to the diagonal grids along the X-direction or Y-direction. Then, the brokers in the diagonal grids continue to forward the received subscriptions along the diagonal towards the PHB. Obviously, the delivery path on the diagonal line can be shared by different SHBs and the redundant event delivery can be reduced. Figure 2(a) and (b) show the examples of subscription and event delivery paths established by SDP, respectively.

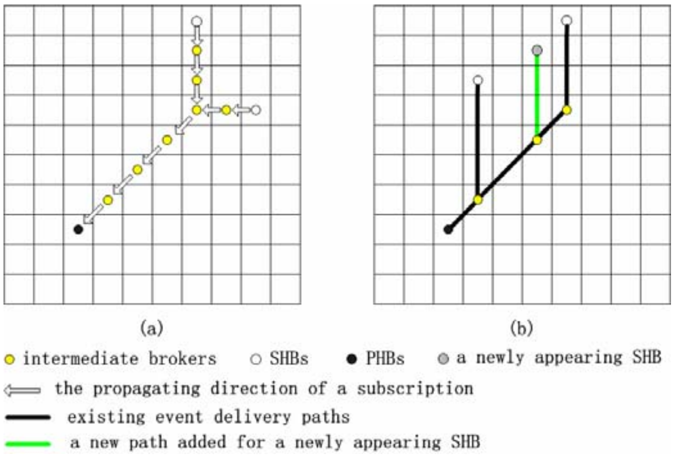


Fig. 2 examples of the subscription and event delivery paths established by SDP

When a broker receives a new subscription *S*, it first checks its routing table. If an entry in the routing table can satisfy *S* or the broker is the target PHB, the broker send the reply back to the sender of *S*; otherwise, the broker employs Algorithm 1, shown in Fig. 3, to determine the next-hop broker and then, forward *S* to the selected next-hop broker.

```
Algorithm 1:
Input::
 S      //the new subscription received by this broker
Output::
 B      //the neighbour brokers which S should be forwarded
PseudoCode:
 if this broker is in Part I, II, V, or VI then
        B ← the neighbor broker in Y-direction.
 else if this broker is in Part III, IV, VII, or VIII then
        B ← the neighbor broker in X-direction.
 else if this broker is in a L-diagonal grid then
        B ← the neighbor broker in L-direction
 else if this broker is in an A-diagonal grid then
        B ← the neighbor broker in A-direction
 else if this broker is in an axis grid then
        B ← the neigbor broker towards the target PHB
            along the axis passing this broker's grid.
 end if
 return B.
```

Fig.3 Algorithm 1

C. *The Shortest Delivery Path with Lower Delivery Overhead (SDP-LDO) protocol*

Using the *SDP-LDO* protocol, the SHBs also greedily forward the subscriptions to find the shortest paths between a PHB and a SHB. However, the brokers forward the subscriptions not only along the X-direction or Y-direction, but also along the L-direction and A-direction. This allows different SHBs to share longer event delivery paths than SDP with a little increase of the subscription overhead. Figure 4(a) and (b) show the examples of subscription and event delivery paths established by the SDP-LDO protocol, respectively.

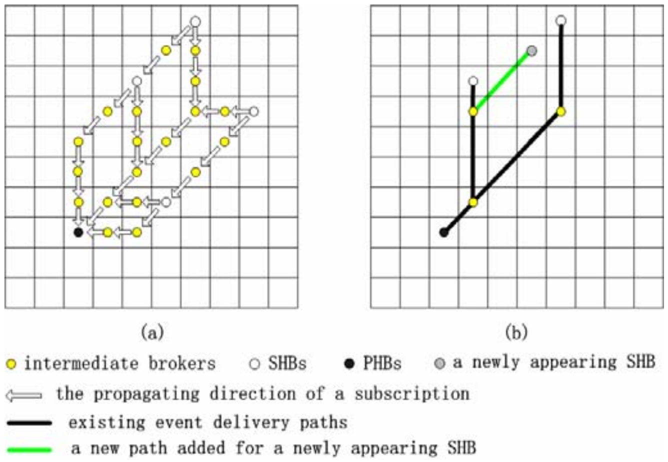


Fig. 4 Examples of the subscription and event delivery path established by SDP-LDO

When a broker receives a new subscription *S*, it first checks its routing table. If a routing entry in the table can satisfy *S* or the broker is the target PHB, the broker sends the reply back to the sender of *S*; otherwise, the broker employs Algorithm 2, shown in Fig. 5, to determine the next-hop broker, and then forward *S* to the selected next-hop broker.

```
Algorithm 2:
Input::
 S      //the new subscription received by this broker
Output::
 B      //the neighbour brokers which S should be forwarded
PseudoCode:
if this broker is in Part I, II, V, or VI then
    B ← the neighbor brokers in Y-direction.
    if this broker is in Part I or V then
       B ← B ∪ the neighbor broker in L-direction
    end if
    if this broker is in Part II or VI then
       B ← B ∪ the neighbor broker in A-direction.
    end if
else if this broker is in Part III, IV, VII or VIII then
    B ← the neighbor broker in X-direction.
    if this broker is in Part III or VII then
       B ← B ∪ the neighbor broker in A-direction
    end if
    if this broker is in Part IV or VIII then
       B ← B ∪ the neighbor broker in L-direction
    end if
else if this broker is in a L-diagonal grid then
       B ← the neighbor broker in L-direction
else if this broker is in an A-diagonal grid then
       B ← the neighbor broker in A-direction
else if this broker is in an axis grid then
       B ← the neigbor broker towards the target PHB
       along the axis passing this broker's grid.
end if
return B.
```

Fig. 5 Algorithm 2

D. The Lowest Delivery Overhead (LDO) protocol

Using the LDO protocol, the SHBs and the intermediate brokers forward the received subscriptions to all of their neighboring brokers. Thus, a newly joining SHB can find the nearest broker on the existing event delivery path, which can forward the desired events to the SHB. Although the LDO protocol cannot ensure the shortest delivery path between a PHB and a SHB, it can greatly reduce the redundant event delivery by allowing different SHBs to share event delivery paths as long as possible. Figure 6(a) and (b) show the examples of subscription propagation and event delivery paths established by LDO protocol, respectively.

When a broker receives a new subscription *S*, it first checks its routing table. If a routing entry in the table can satisfy *S* or the broker is the target PHB, the broker sends the reply back to the sender of *S*; otherwise, the broker will forward *S* to all of the neighbouring brokers except the sender of *S*.

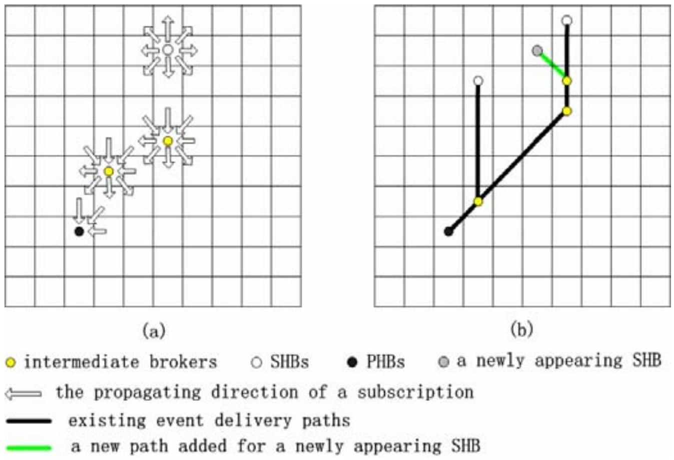


Fig. 6 Examples of the subscription propagation and event delivery paths established by LDO

5. Performance Evaluation

We have performed the theoretical analysis and experimental evaluation on the performance of our protocols. Due to the limit in space, for the theoretical analysis, we only briefly describe the findings and omit the details.

We have theoretically shown that the SDP and SDP-LDO protocols can establish the shortest delivery paths between every pair of SHB and PHB, while SDP-LDO has lower delivery overhead than SDP. This shows that SDP and SDP-LDO can achieve lowest delivery latency. In addition, LDO achieves the lowest delivery overhead among the three proposed protocols.

We have also proved the following theorem.

Theorem 1: In a region of L*L grids, if the PHB is at the central grid of the region, the expected length of delivery paths established by SDP and LDO-SDP is less than 1/3L+0.09.

This result of the expected length of the delivery paths shows that our protocols can achieve attractive scalability.

Now, we discuss the setup of the simulations and the performance metrics used for evaluation. We describe the evaluation results and compare our protocols with the existing solutions, including GEAR and an enhanced random delivery strategy.

In our experimental evaluation, we used two deployment regions with different sizes of 11*11 and 21*21 grids, respectively. The PHB is in the top-left grid. In the two regions, the numbers of subscribers are 2i|i=1, 2, 3,…, 10. The subscribers are randomly deployed into the regions and at most one subscriber in one grid. The results in the performance figures are the average values of the results from 10,000 experiments using the same parameters. In our simulation, a packet contained only one subscription, one subscription reply or one event. We also simulated GEAR and an enhanced random strategy with the same parameters. Using GEAR, a broker greedily forwards the subscription to the neighbour broker which is geographically nearest to the PHB. Using the enhanced random strategy, a broker randomly selects the next-hop broker from the neighbours which are closer to the PHB than the current broker.

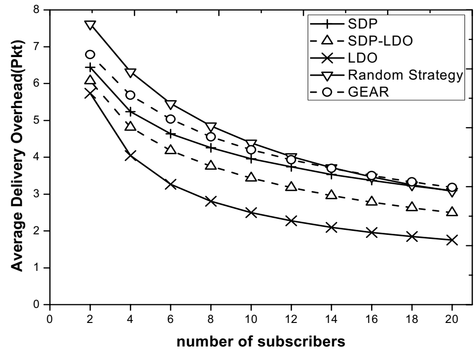
In the simulations, we use the following two metrics to measure the performance of our three different routing protocols.

Average Delivery Overhead is the average number of packets transmitted for delivering one event by a subscriber.

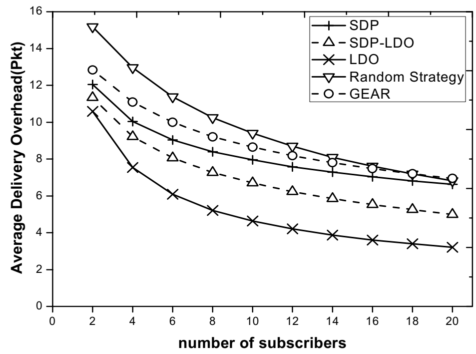
Average Subscribing Overhead is the average number of packets transmitted for establishing event delivery paths divided by a SHB. Simulation Results and Analysis

Fig.7 shows the results of average delivery overhead of different protocols with different simulation parameters. The average delivery overhead becomes larger when the region size increases. In both grid configurations, SDP has the largest delivery overhead among our protocols whereas LDO has the least overhead among the three protocols (see Fig. 7(a)(b)). The performance of SDP-LDO is a little worse than SDP but better than LDO. All three protocols outperform GEAR and the random strategy. The performance of SDP is closer to the random strategy and GEAR along with the increase of the numbers of the subscribers, while SDP-LDO and LDO are 14% better than GEAR. More specifically, when the number of subscribers becomes 20, SDP-LDO and LDO in the both grid configurations outperform GEAR about 30% and 45%, respectively. In addition, we find that the performance of our protocols becomes better with the increase of subscribers. This demonstrates the good scalability of our protocols.

Fig.8 shows the simulation results of average subscribing overhead of different protocols with different simulation parameters. The subscribing overhead increases with the increase of the region size. In both grid configurations, we can find that SDP has the lowest subscribing overhead, although SDP has a higher event delivery overhead. LDO, however, has a much higher subscribing overhead than other protocols (Fig. 8(a)(b)). Obviously, LDO suffers from a high cost of finding the nearest suitable broker. SDP-LDO uses a marginally higher cost to achieve a significantly lower event delivery overhead compared with other protocols. The subscribing overheads of the three protocols drop along with the increase number of the subscribers. This also demonstrates a good scalability of our protocols.

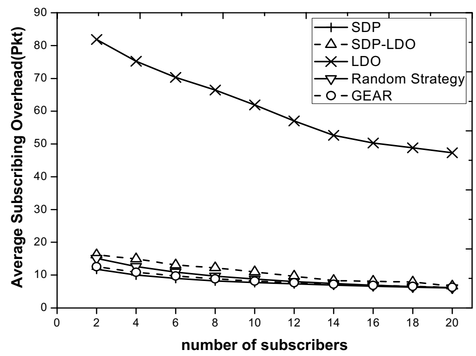


(a)11*11 grids

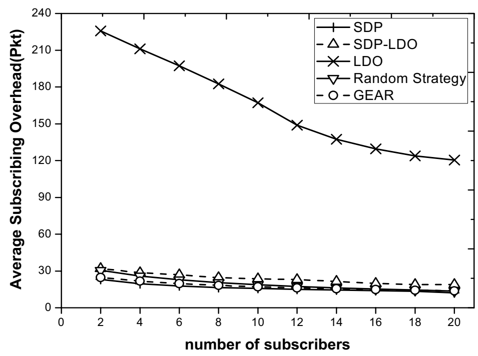


(b) 21*21 grids

Fig. 7 average delivery overhead of different protocols



(a) 11*11 grids



(b) 21*21 grids

Fig. 8 average subscribing overhead of different protocols

The three protocols have different performance advantages. The SDP protocol has the lowest subscribing cost because it just propagates the subscriptions in one direction. However, this method cannot help a SHB find a delivery path with lowest delivery overhead. Thus, if the short-term event subscription frequently occurs, for example, real-time vehicle information system can employ SDP to save subscribing overhead. On the other hand, the LDO protocol can establish the shortest delivery path, and then achieve the lowest event delivery overhead. However, a broker using the LDO protocol needs to propagate the subscriptions in more directions and, therefore, the subscribing overhead is very high. Thus, if there are not many new subscriptions and these subscriptions are valid for a long time, we can use the LDO protocol to reduce event delivery overhead. Thus, the LDO protocol is suitable to the lowly dynamic long-term applications, for example, long-term eco-systems or wild environment monitoring systems. The SDP-LDO protocol can reduce the event delivery overhead with a slightly higher subscribing overhead. So, it can achieve a good tradeoff between event subscription and delivery overhead and can be widely used in various WSN-based applications.

6. CONCLUSIONS

In this paper, we have proposed highly efficient and scalable event delivery protocols for Publish/Subscribe systems in WSNs. The protocols use the geographic information to reduce routing cost and, meanwhile, significantly reduce the event delivery overhead by allowing different subscribers to share the event delivery paths. Our protocols are designed to achieve different performance objectives. We have theoretically analyzed and experimentally evaluated the performance of the protocols. The results show that, in comparison with the existing works, our protocols can significantly improve the resource efficiency and enhance the scalability of Pub/Sub systems for WSNs. The analysis and evaluation results also show that the proposed protocols can achieve a good tradeoff between the costs of event subscription and delivery overhead.

ACKNOWLEDGEMENT

This work is partially supported by University Grant Council of Hong Kong under the CERG grant B-Q937 (PolyU 5105/05E), the Hong Kong Polytechnic University under the ICRG grant G-YE57.

REFERENCES

[1] I. Akyildiz, *at el*. A survey on sensor networks. In IEEE Communication Magazine, 40(8):102-114, 2002.
[2] K. Akkaya, M. Younis. A Survey of Routing Protocols in Wireless Sensor Networks. In Elsevier Ad Hoc Network Journal, 3(3): 325-349, 2005.
[3] P.Th. Eugster, *at el*. The many faces of Publish/Subscribe. In ACM Computing Surveys Vol.35(2): 114 - 131, 2003.
[4] J. Heidemann, *at el*. Building Efficient Wireless Sensor Networks with Low-Level Naming. In ACM Intl Symposium on Operating System Principles (SOSP), 2001.
[5] E. Souto, *at el*. Mires: a publish/ subscribe middleware for sensor networks. In ACM Personal and Ubiquitous Computing, 10(1): 37-44, December 2005.
[6] P. Costa, *at el*. Publish-Subscribe on Sensor Network: A Semi-probabilistic Approach. In IEEE Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2005.
[7] A. Boukerche, *at el*. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems(MSWiM),October. 2004.
[8] V.J. Marathe, T. Herman. The Design of an Interval Routing Enabled Publish/Subscribe Communications Protocol for Ad Hoc Sensor Networks. In Midwest Society of Programming Languages and Systems Workshop, November 2002.
[9] J. Heidemann, *at el*. Diffusion Filters as a Flexible Architecture for Event Notification in Wireless Sensor Network. Technical Report ISI-TR-556, USC/Information Sciences Institute, 2002.
[10] Y. Yu, *at el*. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, 2001.
[11] C. Intanagonwiwat, *at el*. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 2000.
[12] Y. Xu, *at el*. Geography-informed Energy Conservation for Ad Hoc Routing. In ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), 2001.
[13] K. Akkaya, M. Younis. A Survey of Routing Protocols in Wireless Sensor Networks. In Elsevier Ad Hoc Network Journal, 3(3): 325-349, 2005.