Computer Networks xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A holistic sensor network design for energy conservation and efficient data dissemination ☆

Zehua Zhou [a], Xiaojing Xiang [b], Xin Wang [c,*], Jianping Pan [d]

[a] State University of New York at Buffalo, Buffalo, NY, United States
[b] Microsoft Corporation, Redmond, WA, United States
[c] State University of New York at Stony Brook, Stony Brook, NY, United States
[d] University of Victoria, Victoria, BC, Canada

ARTICLE INFO

Article history:
Received 6 November 2009
Received in revised form 3 May 2010
Accepted 4 August 2010
Available online xxxx
Responsible Editor: X. Lin

Keywords:
Wireless sensor networks
Protocol architecture
Cross-layer design
Data dissemination
Energy conservation
Node scheduling

ABSTRACT

One important application of wireless sensor networks is to track mobile elements in the field, where the sensor network generally consists of a large number of sensor nodes deployed in an unattended environment and the sensor nodes are unlikely to be recharged. Redundant sensor nodes are often deployed to increase network robustness and extend network lifetime. There is a big challenge to distribute information and conserve energy in resource-constrained, densely populated sensor networks. Target and inquirer mobility brings further challenges to large-scale sensor networks. Frequent location updates for multiple inquirers and targets can drain the limited on-board energy excessively. We propose a holistic system design across media access control, network and application layers to optimize the performance of large-scale sensor networks by taking into account the interactions and tradeoffs between different design objectives. Our system includes a two-level node activity scheduling scheme for energy conservation and a scenario-aware data dissemination scheme to efficiently distribute query and event data and handle inquirer and target mobility. Furthermore, we analyze the system delay and introduce an adaptive scheduling scheme to reduce the initial sensing delay. The simulation results show that our system can save more than six times of the energy with reduced transmission delay and increased data delivery ratio.

1. Introduction

Recent advances in micro-electro-mechanical systems have enabled the development and deployment of large-scale sensor networks. One important application of wireless sensor networks is to track mobile elements, where the sensor network generally consists of a large number of sensor nodes [1,2] deployed in an unattended environment to be monitored and controlled. To lower the cost, the computation, communication and energy capacities in sensor nodes are often limited, and the sensor nodes are unlikely to be recharged in field. Sensor nodes are prone to failures; hence they are often densely deployed to improve network reliability and lifetime. In addition, the topology of sensor networks can change quickly, especially when nodes fail in operation (e.g., due to running out of on-board battery). Further, although the sensor nodes in a monitoring system are often stationary, the targets to be monitored (e.g., tanks in a battle field, vehicles on the road, and so on) and the inquirers that collect information (e.g., people on foot or in a car) of a sensor network are often mobile. Target mobility causes a frequent change of

☆ A preliminary version of this paper appeared in IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), Niagara-Falls, Buffalo, NY, June, 2006. This work is supported by US NSF under grant numbers CNS-0751121 and CNS-0628093.
* Corresponding author. Tel.: +1 631 632 8402.
E-mail address: xwang@ece.sunysb.edu (X. Wang).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.08.002

correspondent source nodes and inquirer mobility leads to different correspondent sink nodes, which bring more challenges to the design of data dissemination schemes for large-scale wireless sensor networks.

Many issues need to be addressed in order for a large-scale sensor network to function properly. First, to detect an event quickly and accurately, there should be enough sensor nodes to cover the area to be monitored. Second, the data distribution path should be established quickly and reliably with minimal overhead due to the monitoring requirement and the limited capability of sensor nodes, and the distribution scheme should be able to handle target and inquirer mobility efficiently. Third, due to the energy constraint of sensor nodes and the difficulty of recharging them, it is very critical for sensor nodes to conserve energy and maximize the lifetime of the entire sensor network. Many research efforts have been made in recent years trying to achieve one or some of the objectives mentioned above. However, a comprehensive solution is needed to consider the sensor system as a whole and optimize the overall system performance.

There are many interactions and tradeoffs between different design objectives. To conserve energy, a common method is to turn off sensor nodes periodically or randomly [3–5]. However, a sleep node could not participate in data collection and forwarding, which may introduce considerable sensing delay and data transmission delay. The data distribution schemes normally do not take into account the energy conservation requirement [6–8], and flooding data and control messages will consume on-board energy excessively. The routing algorithms for ad hoc networks are normally sophisticated, and extending them directly to sensor networks [9,6,10] may introduce considerable overhead in resource-limited sensor nodes. To handle inquirer mobility, the continuous update of inquirer's location throughout the sensor field [6,8] will increase energy consumption and the chance of transmission collision. TTDD [7] provides a solution to inquirer mobility, but cannot handle target mobility efficiently. On the other hand, tree reconfiguration is proposed in [11] to track a mobile target at the cost of frequent tree-structure maintenance.

The aim of this work is to create a holistic system design as seen in Fig. 1 to optimize the performance of a large-scale sensor network by taking into account different performance factors as well as their interactions and tradeoffs. Specifically, our design utilizes the knowledge in different layers for cross-layer control schemes. We introduce a sub-layer between the media access control (MAC) layer and the network layer, in which a virtual grid is formulated to more efficiently manage the sensor network and a two-level scheduling scheme is designed over the grid structure to better control the duty cycle of sensor nodes to maximize energy conservation as well as to meet the sensing coverage and delay requirement. At network layer, based on the virtual grid structure, we develop a simple stateless data dissemination scheme that can quickly and reliably distribute query and event data, and efficiently handle target and inquirer mobility. The data distribution scheme also takes advantage of the application-layer information, such as the region within which the monitoring objects will be located and the intelligence level of sensor nodes, to further improve the data forwarding efficiency. The efficient data distributions help to avoid significant energy consumption for path establishment. The contributions of this paper are highlighted as follows.

- A virtual grid-based structure to simplify the system design is developed. The system architecture supports efficient data distribution, minimizes transmission delay, and conserves node energy.

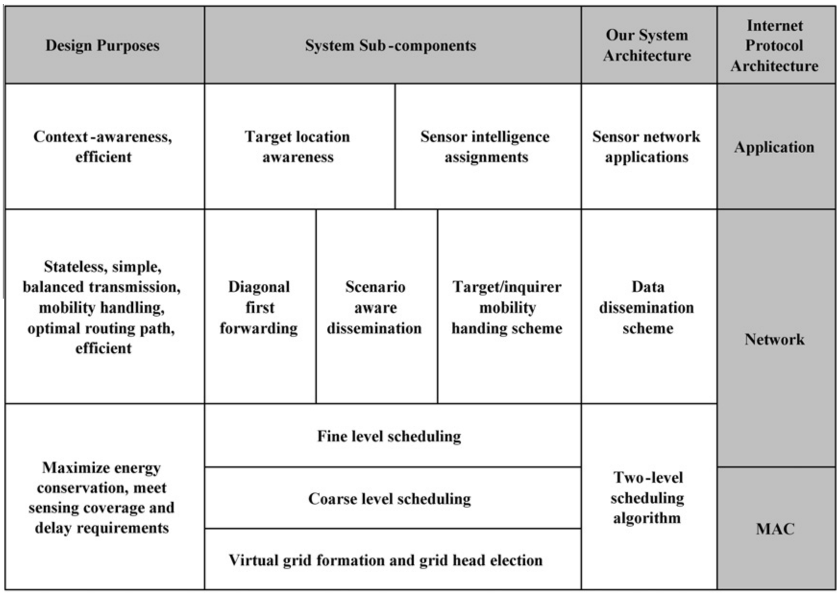


Fig. 1. An overview of the system architecture.

Please cite this article in press as: Z. Zhou et al., A holistic sensor network design for energy conservation and efficient data dissemination, Comput. Netw. (2010), doi:10.1016/j.comnet.2010.08.002

- A scenario-aware data dissemination scheme is designed, which can forward packets quickly without the overhead for route discovery and maintenance. The dissemination scheme can also efficiently handle inquirer and target mobility.
- A two-level node activity scheduling scheme is developed to conserve node energy and extend network lifetime, while still meeting the sensing coverage and delay requirement.
- The impact of our system design on the transmission and sensing delay is analyzed, and an adaptive scheduling scheme is proposed to reduce the *initial* sensing delay introduced by our scheduling scheme.

The rest of this paper is organized as follows. Section 2 compares our system design with those in related work. We give an overview of the system in Section 3. We then introduce our virtual grid formulation and maintenance scheme in Section 4, present the scheduling scheme in Section 5, and describe the data dissemination scheme in Section 6. We analyze the delivery delay and propose an improvement in Section 7, and evaluate the performance of our system in Section 8. Further discussions are presented in Section 9. Section 10 summarizes this paper.

2. Related work

There have been active research efforts in sensor networks in recent years. The query and data distribution schemes in [6,8] generally assume that inquirers are stationary or have low mobility; hence these schemes cannot be applied directly to the case when inquirers have high mobility. Logical objects are introduced in EnviroTrack [12] to be attached to moving entities. The emphasis of the work is however on programming abstraction, and there is no specific consideration on data transmissions or energy conservation. Siphon [13] proposed to use powerful nodes to form virtual sinks and reduce congestion during data dissemination. With a carefully-chosen node duty cycle and the specific forwarding design, there will be no congestion in our system, and there is no need to use special nodes to form a secondary structure for congestion relief.

Similar to our work, both TTDD [7] and Comb-Needle [14] exploit an infrastructure to facilitate transmissions and other functions over the sensor network. TTDD [7] attempts to solve the inquirer mobility problem. When a source node detects a target, it creates a grid structure throughout the sensor field, based on which the query from an inquirer traverses two tiers to reach the source node. Different source nodes do *not* share a common grid structure, which makes TTDD difficult to handle multiple mobile objects. Comb-Needle scheme [14] attempts to combine push and pull strategies for on-demand information dissemination and gathering. The event data are duplicated in a linear neighborhood of each node detecting the object and an inquirer will dynamically form an on-demand routing infrastructure resembling a comb to retrieve the data. One of our design goals is to handle both the target and inquirer mobility. However, the mobility of inquirer was only briefly mentioned and the mobility of target was not discussed in the Comb-Needle paper. With this scheme, a mobile agent may generate a series of queries at different time and locations, and thus a series of comb structures along its trajectory, which would create a high overhead. The movement of targets could potentially trigger to generate a series of needles, and thus consume a large amount of storage space. Also, Comb-Needle implicitly assumes the events would happen before the query. It is not clear how the queries could be sent if the event data are not in storage. Our work takes advantage of the application layer information to distribute queries only toward the region to be monitored, and allows query search based on the node intelligence level which would facilitate the handling of target mobility. The design of more efficient routing schemes to distribute both queries and event data were not specifically considered in Comb-Needle. Instead of following the paths created by combs to distribute queries and data [14], our *stateless diagonal-first forwarding* strategy transmits both data and query along the most efficient paths. An option is proposed in [14] to exploit general geographic routing schemes to transmit data back to the inquirers. However, as pointed out in [15], general geographic routing schemes would be very inefficient to work in the sensor network environment where the event data are not transmitted continuously. The number of infrastructures required in both TTDD (i.e., the source-based grid) and Comb-Needle (i.e., the source-based needle and the sink-based Comb) increases proportionally with the number and mobility of targets and inquirers, which would limit the scalability of these two schemes. In contrast, the virtual grid infrastructure proposed in our work is shared by all nodes in the networks, and is independent of the number and mobility of the targets and inquirers. TTDD and Comb-Needle focus mainly on the efficient match of queries and events. The goal of our work, however, is to design a holistic sensor network infrastructure that could efficiently disseminate both event data and queries with the consideration of mobility of both targets and inquirers, and our two-level node activity scheduling schemes will significantly reduce energy consumption while ensuring the sensing coverage as well as timely and reliable query and event data transmissions by taking advantage of the virtual grid infrastructure.

There are many sensor clustering protocols, such as LEACH [16] and PACT [17], proposed for energy conservation in wireless sensor networks. LEACH uses cluster heads to aggregate data information. However, it assumes that all cluster heads can always communicate with base stations directly, no matter where they are actually located. PACT uses passive clustering, which allows nodes to become backbone communication nodes in turn. Cluster heads and gateway nodes rotate their roles according to their energy levels. Different from these clustering protocols, our virtual grid structure is simpler, and there is no need to exchange role information among neighbors or involve control procedures with high overhead to build and maintain the grid. Our scheme also does not require any sophisticated routing algorithms to form the forwarding path and there is no need to maintain any routing tables. This

Please cite this article in press as: Z. Zhou et al., A holistic sensor network design for energy conservation and efficient data dissemination, Comput. Netw. (2010), doi:10.1016/j.comnet.2010.08.002

further reduces the energy consumption as well as storage and processing requirements in sensor nodes. The virtual grid infrastructure was also used in [18–21] to facilitate multicast group membership management and efficient multicast transmissions over dynamic mobile ad hoc networks. The sensor networks have more constrained resources and a higher requirement for energy conservation, so the data and event transmission scheme in this work is much simpler by taking advantage of the sensor network features. In addition, the virtual grid infrastructure is exploited to facilitate more efficient matching of queries and event data, handling mobility, and performing the fine-level scheduling to conserve more energy.

GAF [3] proposes to turn off unnecessary nodes within the transmission range of other nodes for energy conservation. However, in sensor networks, the detection range of most sensor nodes is smaller than their transmission range. To detect events in a timely fashion, it is necessary to maintain a necessary sensing coverage. Sensor coverage problems have been studied extensively in the literature and a survey can be found in [22]. The coverage studies in the literature, however, often do not consider transmission at the same time. Also, coverage control based on local topology information would require nodes to wake up periodically at the same time for passing and listening to handshake messages [23], which would be difficult and consume more energy, especially when there is no strict time synchronization between nodes. PEAS [5] considers the detection range when forming a connected network. S-MAC [4] tries to reduce the additional delay caused by sleep nodes, but there is still a considerable time delay at each hop, which makes the end-to-end delay significant when there are many hops between the source and sink node. The power saving (PS) mode in IEEE 802.11 DCF is designed for single-hop networks and requires time synchronization. Tseng et al. [24] proposes three sleep schemes to improve the PS mode in IEEE 802.11 multi-hop networks. But the overhead and delay can be significant, since a sender has to wait for its neighbors along the data forward path to wake up before sending out the actual packet. In [25], energy conservation was considered in conjunction with data distribution. In contrast to existing work, we exploit the virtual grid infrastructure in this work to better control node activity to conserve more energy while ensuring higher sensor network performance. Specifically, we provide two-level node activity scheduling scheme. Our coarse level scheduling only keeps awake a necessary number of nodes to maintain coverage, while turning other nodes off for a longer period to conserve energy. Further, with the facilitation of the virtual grid, we introduce a fine-grained scheduling scheme in the time domain to further conserve energy while limiting the initial sensing delay. In addition, with each grid as a management and transmission unit, no additional delay is introduced at intermediate nodes while turning off redundant nodes for energy conservation. Different from existing work in the literature, which normally considers a specific problem, this paper proposes a cross-layer system architecture that efficiently integrates energy conservation, sensing coverage, data dissemination, and mobility handling of both target and inquirer in a unified framework.

3. System design

3.1. System overview

To facilitate sensor network management, the literature work often proposes to group nodes in a sensor network into different clusters. However, it often involves a significant overhead to form and maintain the clusters in an infrastructure-less network with low-cost sensor nodes that are less powerful and prone to failure. Similarly, for surveillance purpose, there is a need to timely capture the events or targets and disseminate the information quickly without incurring a big overhead to establish transmission paths as often proposed in general ad hoc networks. Finally, it is critical to conserve energy while not incurring high detection delay and data transmission delay. To address all these issues, we propose a system architecture with components and objectives summarized in Fig. 1. We first introduce our system components as follows.

- **Virtual grids**. Location information is often needed in sensor networks for event localization [26–28]. We take advantage of the location information to greatly simplify sensor network management. A sensor field is divided into virtual grids, and each node can determine its grid coordinates based on a virtual reference point and its own location, which can be obtained from various localization systems [26–28]. A leader is elected in each grid for local management (information processing, query storage, etc.) and packet forwarding.
- **Two-level scheduling scheme**. To conserve more energy while better controlling the sensing delay, we introduce a two-level node activity scheduling scheme between the MAC and network layer to control node activity both at the topology level and in the time domain.
  - *Coarse-grained scheduling*: only a necessary set of working nodes is kept active to satisfy the sensing coverage requirement, and other nodes are turned into long-term sleep mode.
  - *Fine-grained scheduling*: each grid is divided into several sub-grids, and working nodes in each sub-grid can sleep alternately according to a schedule in a smaller time interval.

  This scheduling scheme can save energy significantly while effectively reducing the maximal sensing delay.
- **Scenario-aware data dissemination scheme**. At the network layer, we introduce a simple scenario-aware, grid-based packet forwarding scheme to efficiently distribute data and query messages, and to handle inquirer and target mobility. The distribution scheme is stateless, and does not need any extra control processes for topology discovery and path maintenance. The knowledge about the location of the event source (or target) and the sensor node intelligence in the application layer are used to help further reduce query distribution overhead and delay, as well as speeding up event notification. Since only the grid leaders participate in active data forwarding, collision and energy consumption are significantly reduced. By keeping grid leaders active all the time, the data can be sent out immediately without being delayed.

3.2. Terminologies and notations

Before describing our system design, we first introduce some terminologies and notations used in this paper.

Source: A sensor node that generates data reports for a target or an event of interest.

Sink: A sensor agent for an inquirer that collects data reports from the sensor network.

Inquirer and target mobility: Both inquirer and target can move quickly. In Fig. 2, a soldier (inquirer) selects a nearby sensor node *B* as the sink node. A tank (target) is detected by sensor node *A*. Node *A* becomes the source node and the data report is sent to the sink. The inquirer and target mobility will change the correspondent sink and source nodes, so that inquirer and target mobility is also known as sink and source mobility in some contexts.

R_{trans}:Transmission range of a sensor node.

R_{detect}:Detection range of a sensor node.

4. Virtual grid formulation and maintenance

We introduce a virtual grid infrastructure to facilitate delay-constrained fine-grained scheduling and quick establishment of query and data dissemination paths without incurring a high overhead. How to build the virtual grid structure and how to elect and maintain the leader for a grid efficiently? We introduce our schemes in this section.

4.1. Virtual grid formation

To simplify the system design, the whole sensor field is divided into virtual grids as shown in Fig. 3. To ensure that all nodes in adjacent grids can communicate with each other directly, the grid size (the side length of a grid) is set to be less than $\frac{1}{2\sqrt{2}} * R_{\text{trans}}$. Each grid is identified by grid coordinates. A sensor node can calculate its grid coordinates (a,b) from its location (x,y) obtained from GPS or localization techniques as:

$$\begin{cases} a = \lfloor \frac{x - x_0}{\text{grid\_size}} \rfloor, \\ b = \lfloor \frac{y - y_0}{\text{grid\_size}} \rfloor, \end{cases} \tag{1}$$

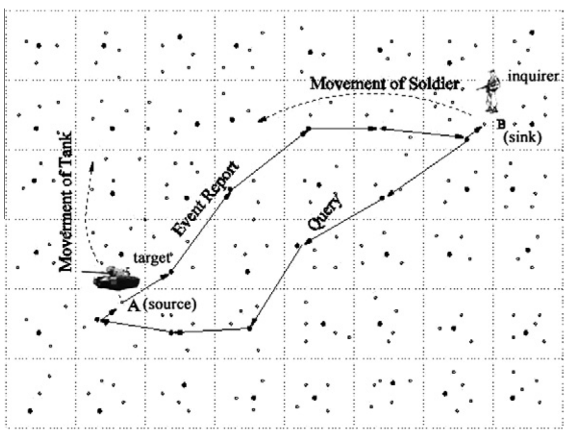


Fig. 2. An enemy tank is detected and an event notification is sent back to the soldier.

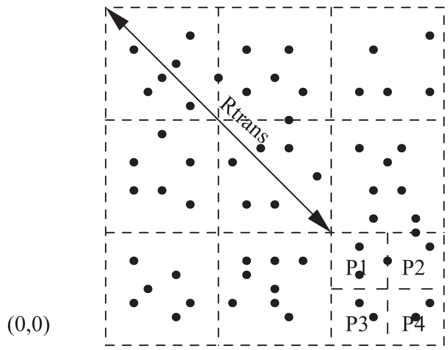


Fig. 3. Sensor nodes in a virtual grid.

where (x_0,y_0) is the location of the virtual origin, which is set at the network initialization stage as one of the global parameters. For simplicity, we assume all the grid coordinates are non-negative.

4.2. Grid leader election

In each grid, a leader will be elected to take care of important management and forwarding functions. To save energy, we consider a two-level scheduling scheme, and only the active nodes after our coarse-grained scheduling (to be described in Section 5.1) will participate in the grid leader election.

Initially, each active sensor node *s* will compete for becoming a grid leader by broadcasting an announcement (carrying its location and address) after waiting for a random time period *f*(*s*), which is determined as follows:

$$f(s) = \text{Random}(T_{\text{grid}})/\text{energy}(s). \tag{2}$$

Here, energy(s) is the remaining energy of node *s* and Random(T_{grid}) returns a random number between 0 and a constant T_{grid}, which determines the average time used for grid leader election. A node with more remaining energy has a higher chance to become grid leader. Once a node receives a grid leader announcement from its own grid, it will give up its own attempt. When there are multiple announcements, the one with a higher address wins the competition. A grid leader will re-announce its role when it detects the existence of another leader with a lower address (e.g., by overhearing a message sent to or from another leader). After this stage, there is one grid leader in each grid. A grid leader can communicate directly with other grid leaders in neighboring grids.

4.3. Grid leader maintenance

Since the grid leader will play an important role as to be discussed, it is important to maintain an active grid leader in each grid. When the energy of a grid leader is below a threshold $E_{\text{gridleader}}$, it will broadcast a request to reelect a new grid leader. All the active sensor nodes in the grid with remaining energy above $E_{\text{gridleader}}$ will participate in the competition using the election procedure described above. The old grid leader will transfer the leading role to the new grid leader. If no grid leader is elected, either due to the loss of the control messages (reelection request or leader announcement) or because the energy of all the active sensor nodes in the grid is below $E_{gridleader}$, the old grid leader will keep its role and broadcast the reelection request periodically. If there is no new grid leader elected after a few rounds, the grid leader will assume that the energy of all active nodes is below the threshold. For reliability purpose, the grid leader will broadcast the grid information it keeps, so that each active node can keep a copy. The grid leader will send reelection request again when detecting that more nodes become active after sleep.

If a grid leader dies suddenly, there is no grace period for grid leader reelection. When a sensor node detects that the grid leader is dead, i.e., it fails to send packets to the leader node for several times, it will send a leader reelection request to the grid. The nodes with energy higher than $E_{gridleader}$ will compete for the role, and the new leader needs to recollect the information about the grid. If a reelection message is sent from a node other than the grid leader (e.g., the grid leader dies) and all the active nodes have energy lower than $E_{gridleader}$, and if no leader is announced within a certain time period, all the active nodes will compete for becoming a grid leader, but have a flag set in their message indicating low energy. If the grid leader has sent each (low energy) node a copy of the grid information before it died, the reelected leader will already have the information and there is no need for the leader to collect the information again.

5. Scheduling scheme

As mentioned in Section 1, nodes in a sensor network are normally densely deployed to improve reliability. Simply leaving all the nodes in active working mode is not necessary for network monitoring, and will not increase the reliability or network lifetime in a long run. On the contrary, the redundant nodes not only consume extra energy, but also lead to more contention when forwarding packets and more transmission failures. A technique commonly used for conserving energy in sensor nodes is to turn the redundant ones into sleep mode. Although the scheme is simple, there is a challenge to schedule the node activity properly to maximize the energy conservation while still meeting the sensing and transmission requirements.

In this paper, we consider a two-level scheduling scheme to further reduce energy consumption, while at the same time meeting the sensing requirement and limiting the transmission delay. At the coarse level, a probing method will be followed to determine the nodes that need to be kept active to meet the sensing coverage requirement. All the remaining nodes will be turned to long-term sleep mode to save energy. At the fine level, the working nodes will follow a periodic schedule, so that only a subset of the working nodes needs to be active at anytime. That is, working nodes alternate between active and short-term sleep mode to save more energy, while still limiting the initial sensing delay. Fig. 4 depicts the state transition diagram of our scheduling scheme, and we will describe in details the coarse-grained scheduling in Section 5.1 and fine-grained scheduling in Section 5.2.

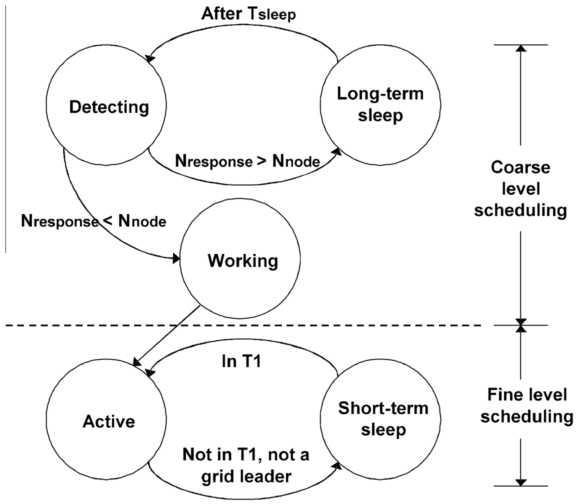


Fig. 4. State transition diagram of a sensor node. To depict the state transition process due to fine-grained scheduling, a node in P1 of Fig. 3 is used here as an example.

5.1. Coarse-grained scheduling

At the coarse level, a node needs to determine whether it should stay active, so that there are enough *working* nodes around to meet the sensing coverage requirement. The process is designed as follows. When a sensor node is powered on, to determine whether a node should stay active, the node needs to go through a detection process by changing its mode to *detecting* after waiting for a random period. The node then broadcasts a detecting message to its neighbors to detect the number of active nodes within the detection range (detection range is generally smaller than transmission range). If a working node in the detection range of the detecting message sender (based on the sender's location) has energy higher than $E_{gridleader}$, it sends back a response message after a small random delay. If the number of response messages received exceeds N_{node}, the detecting node will be considered redundant and go to long-term sleep. We call this kind of sleep nodes *redundant sleep nodes*. If the number of response messages received during a certain period is smaller than N_{node}, the node will enter the working mode. The value of N_{node} depends on the node redundancy and sensing coverage requirements. After this initial stage, a topologically-connected network is established, and the working node will not go back to long-term sleep any more.

After a time period T_{sleep}, a redundant sleep node will wake up and enter the detecting mode right away. It determines whether it should stay active by checking the number of working neighbors using the procedure described above. Before changing the mode from detecting to working, the node needs to announce the change to the grid leader to facilitate the leader election as described in Section 4.2.

5.2. Fine-grained scheduling

If turning off more working nodes, the monitoring domain will not be fully covered and there will be a sensing delay if a target appears in the area not covered. To capture a temporary event, the monitoring domain needs to be fully covered all the time. For continuous events, we introduce a fine-grained scheduling scheme to conserve more energy. To limit the target detection delay, our fine-grained scheduling scheme takes advantage of the virtual grid structure to control the activity of sensors. More specifically, each grid is further divided into four sub-grids, $P1$, $P2$, $P3$ and $P4$ (shown in Fig. 3). Also, the time interval T_{interval} is divided into four time slots, $T1$, $T2$, $T3$ and $T4$. Each sub-grid will be associated with one time slot, $T1$ for $P1$, $T2$ for $P2$, $T3$ for $P3$, and $T4$ for $P4$. During any time slot, only the working nodes in the corresponding sub-grid need to stay active, so a working node only needs to stay active for 1/4 of T_{interval}. For example, nodes in sub-grid $P1$ can be active in $T1$ and turned to sleep during the other time slots. There is a tradeoff in selecting T_{interval}. A large T_{interval} will increase the initial sensing delay as described in Section 7.2. If T_{interval} is too small, there is not enough time for sensor nodes to get the chance to send out a packet. Reducing the number of working nodes at any given time will not only reduce energy consumption but also reduce the chance of transmission collision.

With the coordination of the grid leader, the time synchronization within a grid is not difficult. The global synchronization is not required. Different grids can have different time schedules. The grid leaders will stay active to reduce the transmission delay. When an event occurs and the notification needs to be sent to the sink, the packet is forwarded by the detecting node to the grid leader in the neighboring grid that is closer to the sink node. This process is repeated by grid leaders until the packet reaches the sink.

6. Data dissemination

It is critical to send the surveillance information timely and reliably. To lower the cost, sensor nodes often have limited capabilities. Therefore, there is a need to form data dissemination paths quickly with minimum overhead, in presence of target and inquirer mobility and the unreliability of sensor nodes. In this section, we introduce our *scenario-aware* data dissemination scheme, efficient and stateless data forwarding strategy, and target and inquirer mobility handling approach.

6.1. Scenario-aware data dissemination

The information about the target to be monitored can help us design more efficient query and data dissemination schemes. With the application-layer knowledge, we develop three data dissemination strategies at the network layer based on the type of inquirers as follows:

- Target location aware: the inquirer knows the current location of the target.
- Target area aware: the inquirer knows the area within which the target is currently located, but does not know the exact location of the target.
- Target location unaware: the inquirer does not have any location information about the target.

6.1.1. Target location aware data dissemination

For the first type, the inquirer wants to monitor some targets at a specified location. The sink node, representing the inquirer, first registers the query with the leader in its grid. The grid leader forwards the query to the leader in the neighbor grid that is closer to the source node, which is monitoring the target. The query will be progressively propagated by using the forwarding strategy described in Section 6.2 until it reaches the leader in the source's grid, which will then notify all the working nodes in its grid about the query.

6.1.2. Target area aware data dissemination

For the second type, the inquirer wants to detect some events in a sub-area. Again, the representing sink node first sends its query to its local grid leader, and the query is forwarded toward the target area. When the request reaches the target area, it will be flooded to all grid leaders in this sub-area. For example, in Fig. 5, the soldier in (2,3) wants to know the information about enemy tanks in area [6,6,8,8]. The query will be forwarded to the target area. When the query arrives at area [6,6,8,8], it will be flooded in this sub-area. Instead of flooding all nodes in the field, here only grid leaders participate in the forwarding process, which can greatly reduce energy consumption and the chance of transmission collisions. A grid leader in the sub-area broadcasts the query when receiving the query for the first time and drops it otherwise. The working nodes can overhear the query when it is forwarded by the local grid leader. A sensor node that is in the sleep mode can later solicit the ongoing queries from the grid leader. Hence, all the sensor nodes in this sub-area will be aware of what kind of events they need to detect.

6.1.3. Target location unaware data dissemination

For the third type, the inquirer wants to detect some kind of events in the whole area. Since the sink node does not know the location of the target, the query will be flooded throughout the field and reach all grid leaders in the field. The working nodes will overhear the bypassing messages, and later all sensor nodes will know the events to detect. Even in this case, since queries are forwarded only by the grid leaders, it can still save a lot of energy and lessen the broadcast storm problem.

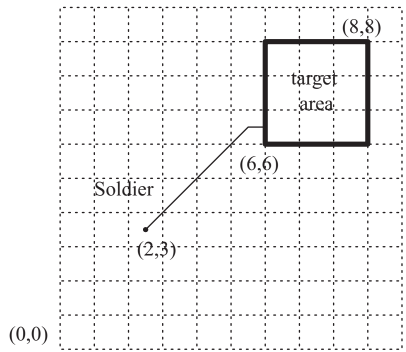


Fig. 5. Routing path for the target area aware data dissemination.

6.2. Stateless diagonal-first forwarding strategy

When forwarding a control or data packet to a grid or a sub-area, the packet will be forwarded to the leader of the neighboring grid that is closer to the destination grid or sub-area. Normally, there are three candidate grid leaders to send a packet to. For example, in Fig. 6(a), if a sink node in grid $A(1,2)$ forwards a query to a source node in grid $E(5,6)$, there are three candidate grid leaders: $B(1,3)$, $C(2,2)$ and $D(2,3)$. In order to minimize the number of grids traversed along the path to the source, the diagonal candidate grid leader is preferred first, until the query reaches the grid leader whose grid coordinates have the same vertical or horizon coordinate value as that of the source grid. The grid leader will then forward the query horizontally or vertically to the source grid. For example, in Fig. 6(b), the queries are first forwarded through a diagonal path, and then through a vertical path to Source 1 or a horizontal path to Source 2 or 3. Fig. 5 demonstrates an example when the destination is a sub-area.

As the leader in a grid may change over the time, it would incur more control overhead for all nodes to track and maintain the grid leader information of their neighbor grids. Instead, our scheme is stateless. When a sensor node wants to send a data packet, it first calculates the destination's grid coordinates (x_2,y_2) and its own grid coordinates (x_1,y_1) using Eq. (1). When the destination is a sub-area, the packet is forwarded towards the nearest grid in the sub-area. The next-hop's grid coordinates $(x_{\text{nexthop}},y_{\text{nexthop}})$ is calculated as:

$$\begin{cases} x_{\text{nexthop}} = x_1 + \frac{x_2 - x_1}{|x_2 - x_1|}, & \text{when } x_2 \neq x_1, \\ x_{\text{nexthop}} = x_2, & \text{otherwise}, \end{cases} \tag{3}$$

$$\begin{cases} y_{\text{nexthop}} = y_1 + \frac{y_2 - y_1}{|y_2 - y_1|}, & \text{when } y_2 \neq y_1, \\ y_{\text{nexthop}} = y_2, & \text{otherwise}. \end{cases} \tag{4}$$

The forwarding node then broadcasts the data packet with the next-hop grid coordinates inserted in the packet header. When a neighbor grid leader receives this packet, if its grid is the designated next-hop, it forwards the packet following the same procedure; otherwise, it drops the packet. To improve the data delivery success ratio, the receiving grid leader will send an acknowledgment to the sender. If the sender cannot get the acknowledgment for a certain time period T_{resend}, it will resend the packet. There may be some void areas because of deployment problems or node failure. When a grid along the path is void, after trying several times, the forwarding leader will try to send the packet to the other two candidate grid leaders (Fig. 6). If this alternative path forwarding fails, a sending node can backtrack to the previous node, which will try to send through its horizontal or vertical neighbors. The number of steps in backtracking can be limited to prevent the overhead and delay. When the maximum backtracking limit is reached, if the forwarding still fails, the packet will be dropped. Besides simplifying the forwarding process, broadcast is also used for transmission efficiency when there are multiple next-hops. When the packet header contains multiple next-hops, a receiving grid leader waits for a random time before sending back the acknowledgment to avoid collision.

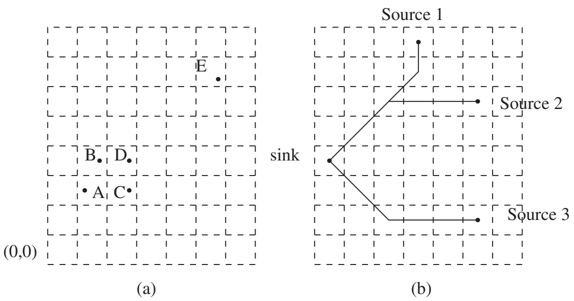


Fig. 6. Three routing candidates and the diagonal-first routing path.

When a data packet reaches the destination grid, the grid leader will check its forwarding entry to see whether there is any forwarding request from a move-away inquirer. If there is such a forwarding request, the grid leader will further forward the data packet to the new grid where the inquirer is currently located; otherwise, the grid leader will broadcast the data packet in its grid so that the sink node can receive it. We will discuss the rule for creating forwarding entry in detail in the next section.

Our *diagonal-first* routing path traverses the same number of grids as the straight-line path between the source and the sink node, which is the shortest path. Based on the virtual grid structure built at the link layer, our routing scheme is simple and reliable. As it does not need active path discovery and maintenance, it is more applicable in the resource-constrained sensor networks. If multiple sources exist and share some portion of the path (e.g., in Fig. 6(b), the path segment from the sink to grid (2,5) is shared by Source 1 and Source 2), only one copy of the query needs to be sent along the shared path, and then the query message will be duplicated and sent to different source nodes at the branching grid (i.e., grid (2,5)). The data forwarding from the source to sink node also follows the diagonal-first routing strategy, but the path from the source to sink node is normally different from the one from the sink to source node. The use of different transmission paths not only reduces collisions in two directions, but also leads to more balanced energy consumption in the network.

6.3. Handling target/inquirer mobility

Although sensor nodes are normally stationary, inquirers and targets can move quickly. We will discuss how to handle inquirer and target mobility in this subsection.

6.3.1. Inquirer mobility

When moving within the same grid, an inquirer does not need any location update. The query results will be forwarded by the grid leader to its desired sink node. When the inquirer moves from one grid to another, it will pick another nearby sensor node as its sink node, and register with the leader of the new grid by providing both the query information and the original grid coordinates. Sometimes, an inquirer moves out of a grid and then moves back again. To avoid a forwarding loop, the leader in the new grid will first check whether it already has a forwarding entry for the inquirer. If it has, the leader will delete the old forwarding entry; otherwise, the grid leader will send a forwarding request to the original grid leader, which will add an entry into its forwarding table and forward the data packets for the inquirer to the new grid later.

The forwarding path may get longer and longer if the inquirer keeps moving from one grid to another. To avoid this case, before the query expires, the inquirer can send a location update that is similar to a new query, while the old path will expire.

6.3.2. Target mobility

To meet different requirements and to track mobile targets more efficiently, we introduce the concept of *node intelligence* in this work. Whether the data packets will be generated for an event or forwarded to the sink node depends on the intelligence level of the source node. A sensor node is assigned to an intelligence type, NORMAL or SMART. For a SMART node, a number between 0 and MaxSmartness is used to represent the SMART level of the node. The intelligence assignment is controlled by the inquirer and flooded at the network initialization stage. A NORMAL sensor node will only generate data packets if the detected event has been inquired, while a SMART sensor node will search for the relevant query with the effort corresponding to its intelligence level if it has not been inquired for the event. Table 1 lists the behavior of a NORMAL/SMART sensor node when detecting an event.

In target location aware data dissemination, the sink node will send the query to the source node whose original location is known. But sometimes the event may be detected by a node other than the queried source node. If the detecting node is SMART, and if it knows the query (e.g., by overhearing), it will just send the data directly towards the inquirer; otherwise, it will send the data with its SMART level to the local grid leader. Upon receiving the event, if the SMART level is 0, the grid leader will only check its own query table; otherwise, the grid leader will search for the queries within the maximal *ring* distance of surrounding grids, where the maximal ring distance is equal to the SMART level of the event source. If a query for this event is found, the data packets will be sent to the sink, and the sensor nodes overhearing the search result will cache the query; otherwise, the data packets are dropped. When the target moves out of their detection range, both NORMAL and SMART nodes become silent.

Similarly, for the case that a sub-area has been queried, if the target is detected by a node outside the queried area, the node will keep silent if it is NORMAL, and in the case it is a SMART node, it will generate data packets and search through its leader for queries within the grid range according to its SMART level. When the target location is unknown, the sink inquires events in the whole area. Since every node is queried, when a node detects the target, it will generate and send out data packets no matter whether it is NORMAL or SMART. In all three cases, the query may be unavailable due to loss or other reasons, so a SMART node will always search for the query through its grid leader according to the node's SMART level.

Table 1
The behavior of a NORMAL/SMART sensor node on detecting an event.

Intelligence type		NORMAL	SMART
Location aware	Queried	Report	Report
	Not queried	Not report	Search (level$_{smart}$)
Area aware	In area	Report	Report
	Out of area	Not report	Search (level$_{smart}$)
Location unaware		Report	Report

7. Delay analysis and improvement

In this section, we examine the average delivery delay through quantitative analysis and simulation, and discuss the method to reduce the initial sensing delay.

7.1. Average delivery delay

Compared with IEEE 802.11-like protocols, an extra sleep delay is introduced by our system due to sensor nodes going to sleep periodically.

If a node is in sleep mode, when an event happens, the node cannot detect the event until it wakes up. This extra delay is common for any protocol that adopts energy conservation schemes by turning off redundant sensor nodes. In our scheme, this extra delay is between 0 and 3/4 of T_{interval}, i.e., the node goes to sleep as a result of fine-grained scheduling just before the event happens and detects the event after 3/4 of T_{interval} when it wakes up. There is no extra delay if sensor nodes are awake. The average delay for the first hop is

$$D_1 = 3/8 * T_{\text{interval}} + T_{\text{other}},$$

where T_{other} includes all other delays except for the sleep delay. T_{other} can include *channel access delay*, *transmission delay*, *propagation delay*, *processing delay*, etc. These delays are inherent to any multi-hop networks using contention-based MAC protocols, and have the same impact on both our scheme and any 802.11-like protocols.

Different from other periodic wake up and go sleep schemes, our system does not have extra delay along the forwarding path, since grid leaders are always awake. Suppose there are N hops from the source to the sink. The total average delivery delay is

$$D(N) = 3/8 * T_{\text{interval}} + N * T_{other}.$$

The overhead ratio that the sleep delay introduces is

$$P_{\text{sleep}} = \frac{3/8 * T_{\text{interval}}}{3/8 * T_{\text{interval}} + N * T_{\text{other}}}.$$

Apparently, the longer the forwarding path is, the smaller the overhead ratio due to the initial sleep delay.

All grid leaders are awake in our basic scheme. To save more energy, some grid leaders can go sleep periodically as well. This strategy trades off more energy saving with a longer delivery delay. Only in this case, extra delay will be introduced at intermediate nodes along the forwarding path, since they have to wait for the neighbor grid leaders to wake up. The average delay for each hop is

$$D'_i = 3/8 * T_{\text{interval}} + T_{\text{other}}.$$

The total average delivery delay is

$$D'(N) = 3/8 * N * T_{\text{interval}} + N * T_{\text{other}}.$$

In this case, the overhead ratio introduced by the sleep delay is

$$P'_{\text{sleep}} = \frac{3/8 * T_{\text{interval}}}{3/8 * T_{\text{interval}} + T_{\text{other}}}.$$

7.2. Adaptive node scheduling

Although only the first hop introduces the extra delay due to the sensor nodes in sleep mode, the total average delay can still be large when T_{interval} is large. To further reduce the total average delay, adaptive scheduling can be adopted. When a source sends out the first a few data packets, nearby awake working nodes can overhear these packets. If the distance between an awake working node and the source is less than a certain threshold, for example, the grid size, the working node can decide to stay awake to be ready for detecting the target. When the distance is greater than the threshold, or if the working node has not heard packets from the source for a while, it will follow the normal activity schedule. By doing so, we only have the extra sleep delay for the first a few packets at the first hop. Compared with the entire field, the area around the source is small. Even though the working nodes closed to the sources are kept active for fast event detection, overall the system can still benefit from our two-level node activity scheduling scheme for more energy conservation. This scheme is efficient when the event is continuous, and most sensor networks are deployed to track continuous events.

7.3. Simulation study

We conducted simulations to study how T_{interval} impacts the initial delay for the first few packets at the first hop. The simulation settings and protocol parameters are described in Section 8. As shown in Fig. 7, the initial delay decreases almost linearly when T_{interval} decreases, which validates the analysis above that the initial delay is close to $3/8{*}T_{\text{interval}}$. The users can select an appropriate T_{interval} according to the requirement of the initial delay. A smaller T_{interval} can be adopted if a small initial delay is required.

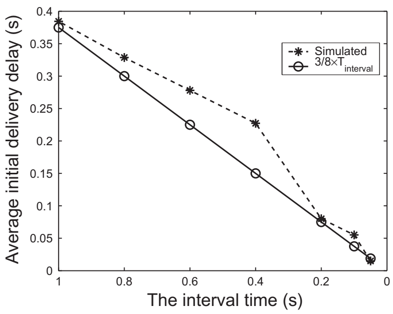


Fig. 7. Average initial delivery delay vs. T_{interval}.

8. Performance evaluation

In this section, we evaluate the performance of our system by using simulation. For the convenience of presentation, in this section, we call our scheme EEDD (Energy-Efficient Data Distribution). We first describe our performance metrics and simulation scenarios. We then evaluate the system performance in these scenarios. Finally, we show the comparisons between our scheme and TTDD. The results confirm that EEDD can deliver data efficiently and handle both target and inquirer mobility well.

8.1. Scenarios and metrics

We implemented our system architecture (Fig. 1) in Network Simulator version 2 (NS2) [29]. We use 802.11 DCF as the underlying MAC. The transmission, reception, idle, and sleep power levels of sensor nodes are 24.75 mW, 13.5 mW, 13.5 mW and 0.015 mW, respectively [4]. The transmission range R_{trans} is 250 m, and the detection range is 20 m. To ensure that all nodes in adjacent grids can communicate with each other directly, the grid size is set to 80 m, which is less than $\frac{1}{2\sqrt{2}} * R_{\text{trans}}$. N_{node} is set to 1, the minimum number of alive nodes to ensure full sensor coverage. T_{interval} depends on application requirements. We set T_{interval} to 0.6 s, and it only impacts the event detection delay for the first a few packets at the first hop. We have studied the impact of T_{interval} on detection delay in Section 7.1. T_{grid} for grid leader election is set to $2{*}T_{\text{interval}}$. T_{sleep} is set to Random (0, 10) s. With a larger T_{sleep} and a correspondingly longer T_{interval}, more energy can be saved; however, if T_{sleep} is too large, it would take a longer time to recover the sensing coverage when the number of alive sensor nodes in the neighborhood is smaller than N_{node}.

We use three metrics to evaluate the performance of EEDD. The *average energy consumption ratio* is defined as the ratio of the average energy consumption to the initial energy in the network. This metric also indicates the overall lifetime of sensor nodes. The *delivery success ratio* is the ratio of the number of successfully delivered data packets to the number of data packets generated by the source. This metric shows the data delivery efficiency. The *average delivery delay* is defined as the average time delay between the moment a source transmits a packet and the moment a sink receives the packet. This metric indicates how quickly the sink can get the report from the source. The extra sleep delay is not included here since it largely depends on T_{interval} and only exists for the first a few packets at the first hop.

In most scenarios, we use a field of $400{*}800\ \text{m}^2$, where 3200 nodes are randomly distributed. The average distance between nodes is 10 m. By default, one source, one sink and no target or inquirer mobility are assumed, except in the performance studies on the impacts of mobility and sink/source population, and the source location is known by the sink. Except in the study on the impact of target mobility, the intelligence level of sensor nodes is set to NORMAL. The source generates one report packet per second. Each simulation lasts for 800 s. A simulation result was gained by averaging over ten runs with different seeds.

8.2. Simulation results

8.2.1. Impact of node density and reliability

A higher node density can help extend network lifetime. To evaluate how node density impacts the performance of EEDD, the average distance between sensor nodes is varied by changing the number of deployed nodes. To study how node failure affects EEDD, we allow randomly-chosen nodes to fail simultaneously at time 30 s.

Fig. 8(a) shows the energy consumption ratio when node density and failure ratio increase. For a fixed failure ratio, when the average distance between nodes is greater than 20 m, most energy saving is due to periodically turning off working nodes according to our fine-grained scheduling scheme. When the average distance between two nodes is 40 m, there are four sensor nodes in each grid and one in each sub-grid on average, since the grid size is 80 m. The grid leader is always awake. The working nodes in a sub-grid are awake periodically. If the grid leader is located in the active sub-grid, it is the only awake node in the whole grid; otherwise, one sensor node in the active sub-grid and the grid leader are both awake. So nearly two of four sensor nodes are awake in this scenario. When the node density becomes lower, the energy saving reduces. On the other hand, at a higher node density, more sensor nodes can go to long-term sleep at the coarse scheduling level leading to further energy saving. Due to the memory limitation of the simulation tool, we cannot simulate the scenario with even more nodes. But the trend shows that more energy will be saved with a higher node density. For a given node density, the energy consumption increases when node failure ratio increases, as the effective node density reduces.

Fig. 8(b) shows the average delivery delay with the variation of node density and node failure ratio. For a given failure ratio, the average delay is seen to increase with the node density. When the node density increases, more control messages are generated, e.g., to detect the number of active nodes for scheduling purpose, which leads to more collisions and hence a longer transmission delay. When the average node distance is below the detection range, i.e., 20 m, the increase of delay is much smaller. This is expected, as the redundant nodes within the same detection range are turned off without impacting the transmission. For a fixed node density, the average delay increases when node failure ratio increases. When a grid leader fails to forward packets to its preferred neighbor grid leader, it has to try alternate paths as described in Section 6.2, which leads to a longer transmission delay.

Fig. 8(c) shows that the success ratio drops slightly when node density increases due to more collisions. With a fixed node density, the success ratio drops when node failure ratio increases. But the delivery success ratio is still above 95% even when the failure ratio is 20%. This confirms that EEDD is resilient to node failure.

8.2.2. Impact of sink and source population

In this simulation, sources and sinks are randomly deployed in a specified sub-area. Each sink inquires the specified sub-area for events of interest. Fig. 9(a) shows the average energy consumption ratio with multiple sinks. First, with a fixed number of sources, the energy consumption ratio increases almost linearly with the number of

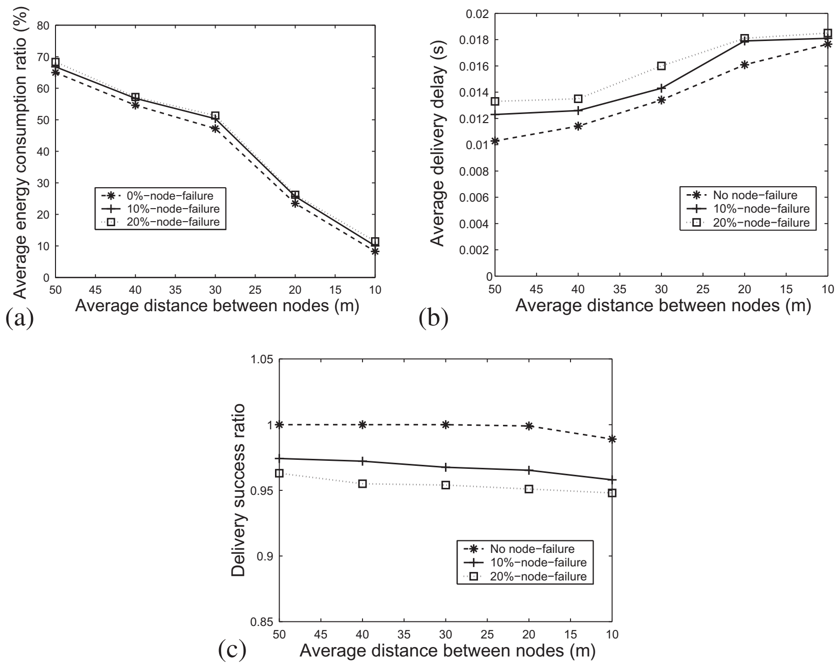


Fig. 8. Performance with different node densities and reliabilities: (a) average energy consumption ratio; (b) average delivery delay; (c) delivery success ratio.

Please cite this article in press as: Z. Zhou et al., A holistic sensor network design for energy conservation and efficient data dissemination, Comput. Netw. (2010), doi:10.1016/j.comnet.2010.08.002

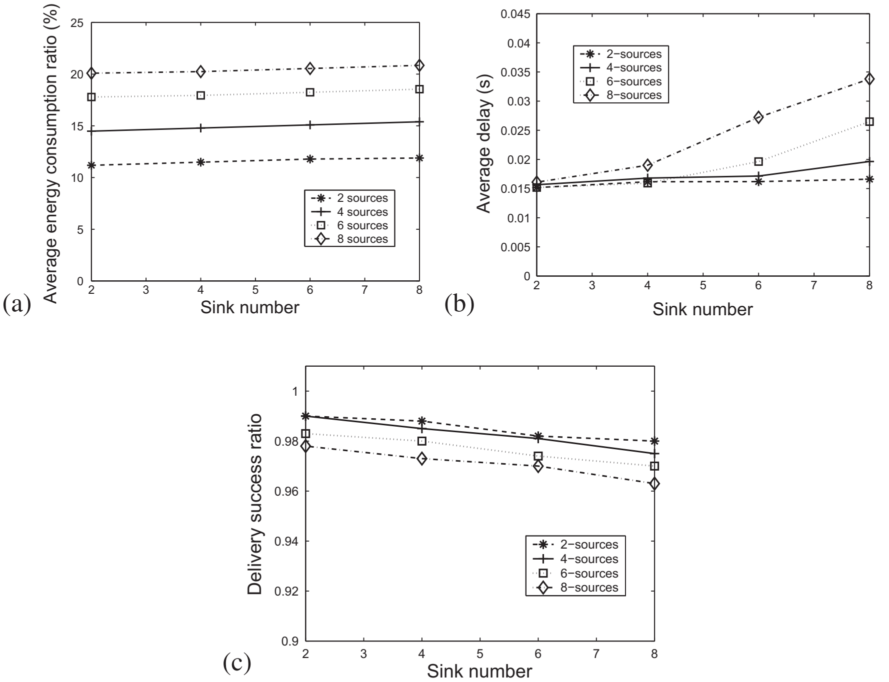


Fig. 9. Performance with different numbers of sinks/sources: (a) average energy consumption ratio; (b) average delivery delay; (c) delivery success ratio.

sinks. This is due to the fact that more queries are generated and hence more data packets are generated. Also, more grid leaders are involved in forwarding packets, since sinks are located in different grids. The increase is not significant, as a report sent for multiple sinks can share the transmission path when possible. Second, with a fixed number of sinks, the energy consumption ratio increases when the number of sources increases. The increase is also due to more data packets generated and more grid leaders involved in forwarding packets. The number of sources has

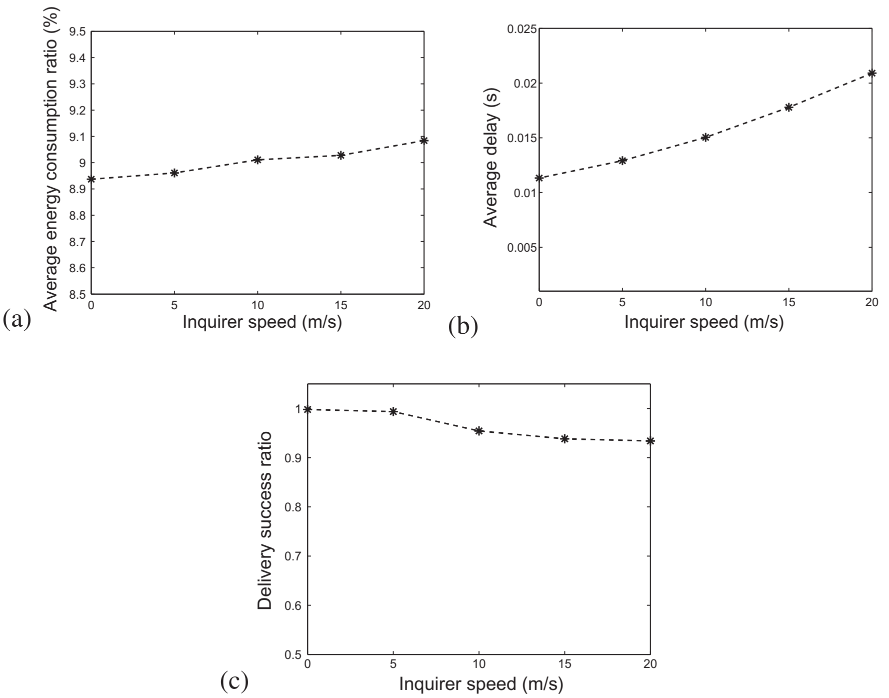


Fig. 10. Performance with different inquirer speeds: (a) average energy consumption ratio; (b) average delivery delay; (c) delivery success ratio.

a greater impact on energy consumption than the number of sinks. As the number of sources increases, more working nodes around the sources have to be awake to reduce event detection delay, and hence more energy is consumed.

Fig. 9(b) shows that the average delivery delay tends to increase with more sinks and sources. More data packets introduce more collisions and a longer transmission delay. Again, the number of sources seems to have greater impact on the delay than the number of sinks, due to more awake working nodes and hence more collisions. Similarly, Fig. 9(c) shows that the delivery success ratio decreases when the number of sources/sinks increases, due to the increased number of collisions.

8.2.3. Impact of inquirer mobility

Now we evaluate the impact of inquirer mobility in EEDD. Fig. 10(a) shows that the energy consumption ratio increases when the inquirer has a higher mobility. This is because with a higher inquirer mobility, more location update messages are generated, and more grid leaders are involved in forwarding packets with longer forwarding paths. Hence the delay increases and the success ratio decreases slightly as shown in Fig. 10(b) and (c), due to the longer forwarding paths and more collisions caused by more location update messages. However, even when the inquirer moves at a speed of 20 m/s, the success ratio is still above 90%. This indicates that our forwarding mechanism works quite well when the inquirer changes its location quickly.

8.2.4. Impact of target mobility

To study how target mobility impacts the performance of EEDD, we vary the speed of the target. In this case, an inquirer is interested in a specific sub-area [4,2,7,4]. The target will randomly move around the whole area. We let sensor nodes have a SMART level of 4, which means if the event is not inquired in a grid, the grid leader will search four layers of grids around the target region for a possible query.

Fig. 11(a) shows that the energy consumption increases when the target mobility increases. With the increase of moving speed, a target is more likely to leave the inquired area, which will trigger more query search attempts. The increased number of grid leaders involved in query search will cause more energy consumption. Additionally, the query search will also delay the sending of information from the event source to the inquirer, as shown in Fig. 11(b).

Fig. 11(c) shows that the success ratio decreases slightly when target mobility increases. When a source node searches for the query using flooding, more collisions happen. This can reduce the delivery success ratio. When the target moves at a maximal speed of 20 m/s, the success ratio is still quite high.

8.2.5. Comparison with TTDD

We compare the performance of EEDD and TTDD by varying node density, and by varying the node failure ratio with an average inter-node distance around 20 m. We double the initial energy of sensor nodes for both protocols so that TTDD will not run out of energy. We cannot simulate TTDD when the average distance between nodes is below 20 m due to memory saturation in simulation (TTDD needs to maintain more per-node information).

Fig. 12(a) shows the energy consumption ratio comparison between EEDD and TTDD with different node densities. As described in the previous subsection, two out of four sensor nodes are awake most of the time when the

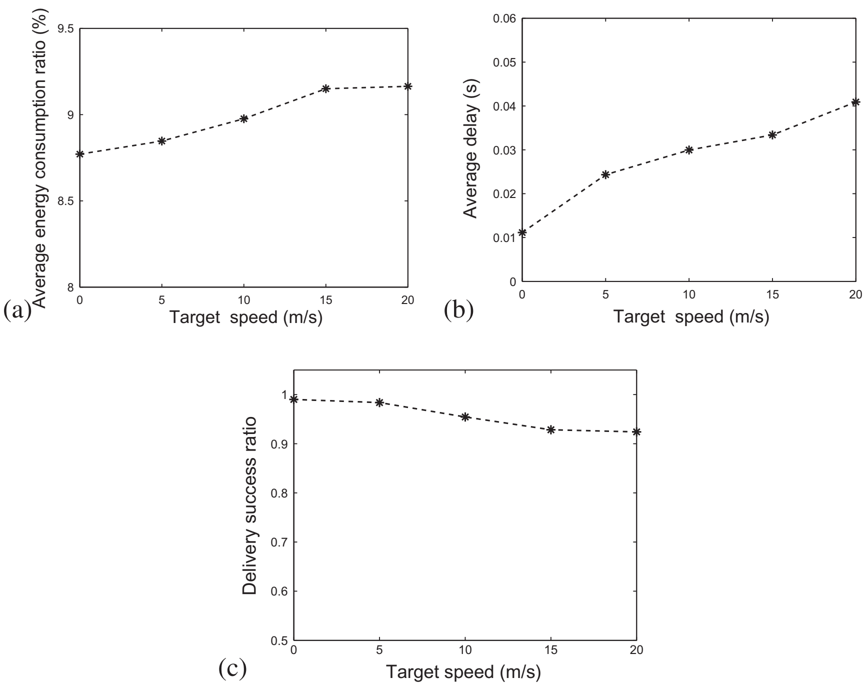


Fig. 11. Performance with different target speeds: (a) average energy consumption ratio; (b) average delivery delay; (c) delivery success ratio.

Please cite this article in press as: Z. Zhou et al., A holistic sensor network design for energy conservation and efficient data dissemination, Comput. Netw. (2010), doi:10.1016/j.comnet.2010.08.002

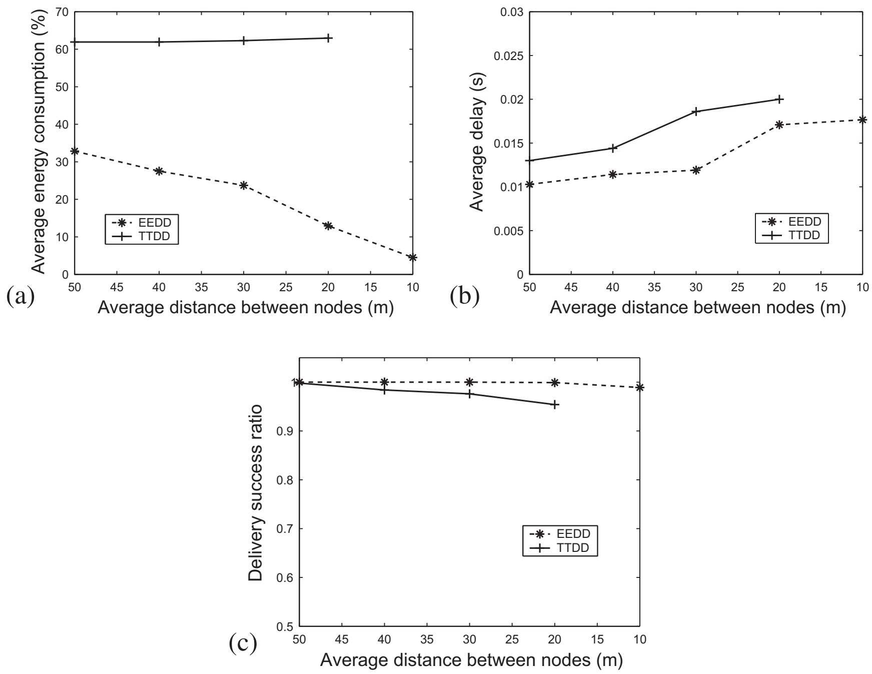


Fig. 12. Performance comparison with different node densities: (a) average energy consumption ratio; (b) average delivery delay; (c) delivery success ratio.

average distance between nodes is 40 m. In this case, we can see that EEDD only consumes less than half of the energy consumed in TTDD, which indicates that our fine-grained scheduling scheme is very effective in conserving sensor network energy. When node density increases, the energy consumption of EEDD decreases due to more working nodes going to sleep. When the average inter-node distance is 10 m, EEDD saves energy significantly, while the energy consumption is six times higher at a lower node density when the average inter-node distance is 50 m.

Fig. 12(b) shows that the average delay for both EEDD and TTDD increases when the node density increases. This

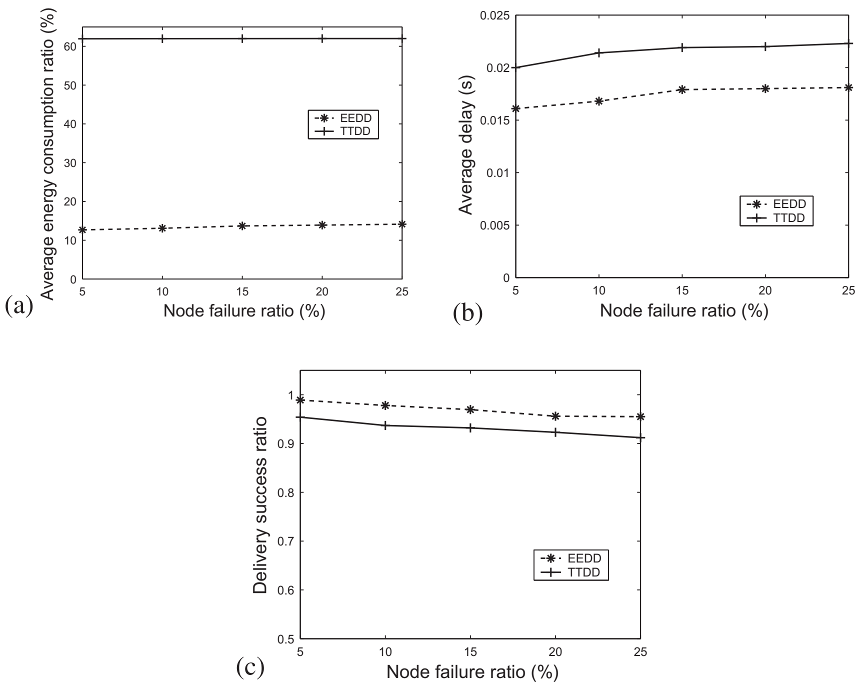


Fig. 13. Performance comparison with different node failure ratios: (a) average energy consumption ratio; (b) average delivery delay; (c) delivery success ratio.

is because there are more working nodes resulting in more collisions. The delay for EEDD is smaller than that for TTDD, as our two-level scheduling can more efficiently arrange node activity to increase energy conservation and reduce the chance of transmission collisions. Remember that the initial extra sleep delay is not counted in EEDD because it largely depends on T_{interval} and only appears when the source sends out the first few packets. The impact of the initial extra delay will be averaged out as time goes by when the source keeps tracking the target.

As demonstrated in Fig. 12, the success ratio for EEDD decreases slightly with the density, and decreases quicker for TTDD due to more working nodes and hence more collisions. Overall we can see that TTDD consumes energy six times greater than EEDD when the average inter-node distance is 20 m, and according to this trend, EEDD would save energy up to twelve times when the node distance is 10 m. More energy can be saved with a higher node density. The average transmission delay is similar since the extra sleep delay only impacts the first few packets at the first hop. With fewer working nodes awake and fewer collisions, EEDD has a shorter average delay.

Fig. 13(a) shows the impact of node failure in EEDD and TTDD. When more nodes fail, the energy consumption increases slightly for EEDD. This is because that the node density decreases when node failure ratio increases, and the energy saving due to a higher node density decreases. Fig. 13(b) shows that the average delay increases for both EEDD and TTDD when the node failure ratio increases. In TTDD, when a dissemination node on the grid fails, it takes time to detect the failure and find an alternative node to forward the data. EEDD takes time to reelect a grid leader to replace the dead one, and simultaneously tries to use alternate paths for routing packet. Either approach increases the delivery delay. Fig. 13(c) shows that the delivery success ratio for both EEDD and TTDD decreases when the node failure ratio increases. EEDD is seen to be more reliable with a higher delivery success ratio in all the failure test scenarios. This demonstrates that EEDD can achieve significant energy saving while providing more efficient and reliable data delivery.

9. Further discussions

In our fine-grained scheduling scheme, each grid is divided into four sub-grids. Actually to save more energy, the grid can be further divided into six sub-grids, nine subgrids and so on. The larger the number of sub-grids, the more energy can be saved. However, the time slot for each subgrid should be large enough to allow nodes to communicate. In addition, as the number of sub-grids increases, the impact of T_{interval} on the initial detection delay also increases.

In our system, the sensor nodes are assumed to be stationary and only the inquirer and the target are mobile. But this protocol can also be extended to the case that every sensor node is mobile. In this case, the mobility factor may be considered for grid leader election. A sensor node that has a lower speed can have a higher possibility of being a grid leader. The handoff between grid leaders should also be introduced when a grid leader moves out of its grid. Our grid structure is stable and expected to better handle mobility than general cluster-based schemes. The forwarding paths in our system are independent of the position of an individual node, so the data distributions in our system would be more reliable than the schemes that form forwarding paths based on the sensor node distribution and network topology.

10. Conclusions

In this paper, we proposed a holistic system design to optimize the performance of large-scale sensor networks. The two-level node activity scheduling scheme is introduced between the MAC and network layer to significantly reduce energy consumption, while meeting the sensing coverage and delay requirements. Our scenario-aware data dissemination scheme at the network layer can efficiently deliver query and event data, and handle target and inquirer mobility. The information at the application layer is used to further improve the system performance. A grid-based structure is introduced to simplify system design, facilitate scheduling, minimize transmission delay, and reduce flooding cost. We also studied the event detection delay and communication delay of the system and proposed an adaptive scheduling scheme to reduce the initial sensing delay.

The simulation results show that our system saves more energy as node density increases. Up to twelve times energy can be saved when the average node distance is half of the detection range, compared with the protocols without using the energy conservation scheme. At the same time, as our two-level node activity scheduling mechanism reduces the number of working nodes, our system can maintain a lower average delivery delay due to the fewer number of collisions. Reduced delivery delay and increased delivery ratio are also achieved with our efficient and reliable data dissemination scheme. The results also show that our system can efficiently handle target/inquirer mobility and node failure. Compared with TTDD, our system can handle target mobility and is more resilient to node failure. Overall, our system design can reduce the average energy consumption significantly, though our system may experience a higher event detection delay for the first few packets at the first hop.

References

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, in: International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), 2000.

[2] G. Pottie, W. Kaiser, Wireless integrated network sensors, in: Communications of the ACM, 2000.

[3] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc networks, in: Proceedings of the Seventh ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 2001.

[4] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: 21st Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings, vol. 3, 2002, pp. 23–27.

[5] F. Ye, H. Zhang, S. Lu, L. Zhang, A randomized energy conservation protocol for resilient sensor networks, ACM Wireless Networks (WINET) Journal 12 (2006) 637–652.

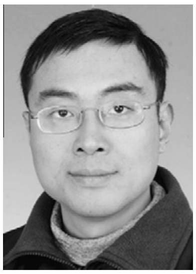
Please cite this article in press as: Z. Zhou et al., A holistic sensor network design for energy conservation and efficient data dissemination, Comput. Netw. (2010), doi:10.1016/j.comnet.2010.08.002

[6] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, Directed diffusion for wireless sensor networking, IEEE/ACM Transactions on Networking 11 (1) (2003) 2–16.
[7] H. Luo, F. Ye, J. Cheng, S. Lu, L. Zhang, TTDD: two-tier data dissemination in large-scale wireless sensor networks, ACM Wireless Networks 11 (1–2) (2005) 161–175.
[8] F. Ye, G. Zhong, S. Lu, L. Zhang, A robust data delivery protocol for large scale sensor networks, in: Information Processing in Sensor Networks (IPSN'03), California, 2003.
[9] T. He, B.M. Blum, Q. Cao, J.A. Stankovic, S.H. Son, T.F. Abdelzaher, Robust and timely communication over highly dynamic sensor networks, Real-Time System 37 (2007) 261–289.
[10] M. Zorzi, R.R. Rao, Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance, IEEE Transactions in Mobile Computing 2 (4) (2003) 337–348.
[11] W. Zhang, G. Cao, Optimizing tree reconfiguration for mobile target tracking in sensor networks, in: IEEE INFOCOM, 2004.
[12] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, L. Luo, S. Son, J. Stankovic, R. Stoleru, A. Wood, Envirotrack: towards an environmental computing paradigm for distributed sensor networks, in: ICDCS'04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), IEEE Computer Society, Washington, DC, USA, 2004.
[13] C.-Y. Wan, S.B. Eisenman, A.T. Campbell, J. Crowcroft, Siphon: overload traffic management using multi-radio virtual sinks in sensor networks, in: SenSys'05: Proceedings of the Third International Conference on Embedded networked sensor systems, ACM Press, New York, NY, USA, 2005, pp. 116–129. <http://doi.acm.org/10.1145/1098918.1098931> .
[14] X. Liu, Q. Huang, Y. Zhang, Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks, IEEE Transactions on Mobile Computing 6 (3) (2007) 241–251.
[15] X. Xiang, Z. Zhou, X. Wang, Self-adaptive on demand geographic routing protocols for mobile ad hoc networks, in: Proceedings of IEEE INFOCOM-2007, Alaska, 2007.
[16] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy efficient communication protocol for wireless micro-sensor networks, in: Proceedings of the IEEE Hawaii International Conference on System Sciences, 2000, pp. 4–7.
[17] G. Pei, C. Chien, Low power TDMA in large wireless sensor networks, Military Communications Conference 1 (2001) 28–31.
[18] X. Xiang, Z. Zhou, X. Wang, Robust and scalable geographic multicast protocol for mobile ad hoc networks, in: Proceedings of IEEE INFOCOM, Alaska, 2007.
[19] X. Xiang, X. Wang, Z. Zhou, An efficient geographic multicast protocol for mobile ad hoc networks, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Buffalo, NY, 2006.
[20] X. Xiang, X. Wang, Y. Yang, Stateless multicasting in mobile ad hoc networks, IEEE Transactions on Computers 59 (8) (2010) 1076–1090.
[21] X. Xiang, X. Wang, Y. Yang, Supporting efficient and scalable multicasting over mobile ad hoc networks, IEEE Transactions on Mobile Computing, in press.
[22] B. Wang, A survey on coverage problems in wireless sensor networks, Technical report, National University of Singapore.
[23] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration for energy conservation in sensor networks, ACM Transactions on Sensor Networks 1 (2005) 36–72.
[24] Y.-C. Tseng, C.-S. Hsu, T.-Y. Hsieh, Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks, in: Proceedings of the IEEE INFOCOM, New York, NY, 2002, pp. 200–209.
[25] Z. Zhou, X. Xiang, X. Wang, J. Pan, An energy-efficient data dissemination protocol in wireless sensor networks, in: IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM-2006), Niagara-Falls, Buffalo, NY, 2006.
[26] J. Bruck, J. Gao, A.A. Jiang, Localization and routing in sensor networks by local angle information, in: Proceedings of the ACM MOBIHOC 2005, Urbana-Champaign, IL, 2005.
[27] T. He, R. Stoleru, J.A. Stankovic, Spotlight: Low-cost asymmetric localization system for networked sensor nodes, in: Proceedings of the Fourth International Symposium on Information Processing In Sensor Networks (IPSN'05), California, 2005.
[28] D. Moore, J. Leonard, D. Rus, S. Teller, Robust distributed network localization with noisy range measurements, in: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04), Baltimore, MD, 2004.
[29] NS2 network simulator. <http://www.isi.edu/nsnam/ns/>.

Zehua Zhou received his BS degree in computer science from University of Science and Technology of China, Hefei, China, and his MS degree in computer science and engineering from the State University of New York at Buffalo, Buffalo, New York. He is currently working at Visible Technology. His research interests include wireless communications, wireless sensor networks as well as mobile ad hoc networks.

Xiaojing Xiang received her BS and MS degrees in computer science from Nanjing University, Nanjing, China, and her PhD degree in computer science and engineering from the State University of New York at Buffalo, Buffalo, New York. She is currently with Microsoft Corporation, Redmond, Washington. Her research interests include protocol design and analysis in mobile ad hoc networks, architecture design for service provisioning, routing and cross-layer protocol design in computer networks, pervasive computing and communications, as well as next generation Internet technologies.

Xin Wang received her BS and MS degrees in telecommunications engineering and wireless communications engineering from Beijing University of Posts and Telecommunications, Beijing, China, and her PhD degree in electrical and computer engineering from Columbia University, New York, NY. She is currently an assistant professor in the department of Electrical and Computer Engineering of the State University of New York at Stony Brook, Stony Brook, New York. Before joining Stony Brook University, she was a Member of Technical Staff in the area of mobile and wireless networking at Bell Labs Research, Lucent Technologies, New Jersey and an assistant professor in the department of Computer Science and Engineering of the State University of New York at Buffalo, Buffalo, New York. Her research interests include analysis and architecture design in wireless networks and communications, mobile and distributed computing, infrastructure design and performance enhancement across network layers, applications and heterogeneous networks, network and mobility management, QoS, signaling and control, as well as support for advanced network services and applications.



Jianping Pan is currently an assistant professor of computer science at the University of Victoria, British Columbia, Canada. He received his Bachelor's and PhD degrees in computer science from Southeast University, Nanjing, China, and he did his postdoctoral research at the University of Waterloo, Ontario, Canada. He also worked at Fujitsu Labs and NTT Labs. His area of specialization is computer networks and distributed systems, and his current research interests include protocols for advanced networking, performance analysis of networked systems, and applied network security. He is a senior member of the ACM and a senior member of the IEEE.