

Connecting Wireless Sensors in Ubiquitous Computing Environments

Niwat Thepvilojanapong

Department of Information and Communication Engineering, University of Tokyo
4-6-1 Komaba, Meguro, Tokyo 153-8505, Japan
Tel & Fax: +81 3 5452 6268, E-mail: wat@mcl.iis.u-tokyo.ac.jp

ABSTRACT

A number of wireless sensors are always used to collect useful information for most of ubiquitous applications. Such information is accumulated at one, or more, sink nodes to be processed later. In this paper, we propose an energy-efficient paradigm based on hierarchical structure to autonomously connect such sensors and sink nodes in ubiquitous computing environments. As a result, sensed data can be sent to sink nodes efficiently. On the other hand sink nodes can also communicate with a group of sensors for some purposes such as querying, or issuing a command. We show superior performance of the proposed protocol by comparing to previous work in simulated networks.

Keywords

Wireless sensor networks, data collection, data dissemination, evaluation, simulation

INTRODUCTION

Most of ubiquitous applications require surrounding's data to decide their actions. Those data range over various categories based on the applications such as ambient data (temperature, humidity, light, etc.), movement or behavior of targets, abnormal information, and so on. To collect such information, a number of wireless sensors are deployed in an area of interest. Because tiny-sized sensors have limited processing capability and information from one sensor is not quite useful, we need to accumulate and process information at one, or more, sink nodes which can be considered as an *anycast* communication. On the other hand, the sink node needs to communicate with a group of sensors to achieve more efficient sensing tasks which is a kind of *multicast* communication. It may need to query information on-demand instead of waiting for periodic reporting from every sensor, which consumes considerable energy. Another form of communication is network management, such as checking the number of alive nodes. Since the scale of target area varies from a room-sized region until a vast national forest, the network of sensors must be constructed and adapted to any changes autonomously. To satisfy the above requirements, we present a scheme called *Energy-efficient Collection and Dissemination of data (ECD)*.

ECD: PROPOSED PROTOCOL

ECD can do two-way communications between sensors and sinks over multi-hop, wireless environments. The idea of ECD is based on hierarchical tree, and the details are as follows.

Bootstrap Process

Sink nodes initiate the tree construction by broadcasting a *solicit message*. A *Nonmember*, a sensor that is not attach to the tree, decides on its parent from the received solicit messages by waiting for a short period to collect a number of candidates and choosing a node whose defined metrics are the best. The candidates are kept in a *parental candidate (PC) table*, which maintains the pairs of candidate IDs and metrics. A *Member* that is an internal, or leaf, node of the tree also updates its PC table according to the incoming solicit messages. Assume T_{sol} and T_{join} denote the time that a node first received a solicit message and the time that a node joins the tree, respectively. The non-member chooses a node whose T_{sol} is the minimum, and if many nodes have the same value of this metric, it will choose a node whose T_{join} is the minimum. Less time implies a sensor is nearer the sink node, which results in the shorter path.

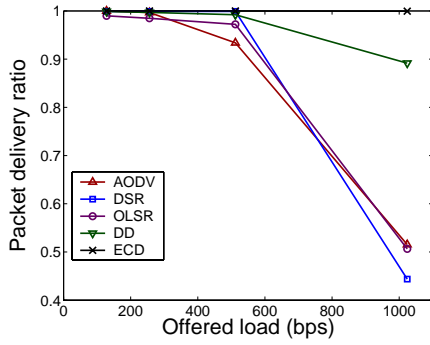
After choosing the parent, the nonmember sends a *join message* to the selected parent to inform that it is a child node. The child node then performs the same process as its parent, i.e., broadcasts a solicit message.

Join Process

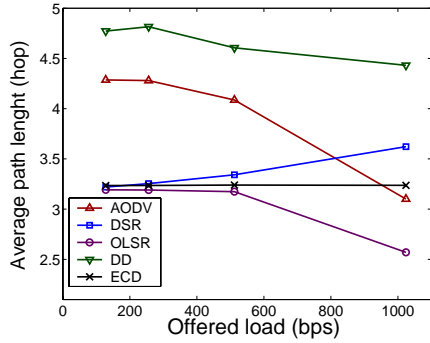
The joining sensor broadcasts a *request message*, thereby making the neighboring nodes aware of its existence. All members of the tree that hear this message reply by unicasting a solicit message to the joining sensor. Then, the process follows the bootstrap process, i.e., the joining sensor sends a join message to the selected parent.

Leave Process

A leave process is used to reconstruct the tree if an internal node has failed due to one of many possible reasons. For example, the battery of the node has discharges with time, or the node may be damaged. If the acknowledgement on MAC layer does not arrive, the node infers that a communicating party has left from the network.



(a) PDR.



(b) Average path length.

Figure 1: Results when collecting data

When an orphaned node is aware of the absence of its parent, it immediately switches to a new parent by choosing the most appropriate parent from its PC table. If there is no any candidate in the PC table, it uses the join process. However, its child and grandchild nodes will not reply to this request message to prevent routing loop. If both methods fail, the orphaned node randomly chooses one of its children as its new parent by giving a higher priority to a child that has at least one candidate. The selected child becomes a parent of the orphaned node, and finds a new parent by choosing from its PC table or using the join process. The relationships between other children of the orphaned node and the orphaned node do not change. Note that the last scenario is quite rare but may occur in very sparse networks.

Data Communication

To collect the data, a sensor sends its sensed data and all of received packets to its parent until a sink. On the other hand, disseminating data from the sink is done by broadcasting to economize energy. Data are disseminated by only parent nodes when receiving new multicast packets. The redundant propagation of a multicast packet multiple times by a single node is prevented because each node retains group ID, source ID, and the most recent sequence number of the packets received. A node discards packets with the same or lower sequence number than those it has received.

Table 1: Results when disseminating data.

	PDR		Avg. transmission	
	MAODV	ECD	MAODV	ECD
Static	0.9929	0.9879	0.43	0.38
Join	0.9946	0.9930	0.45	0.38
Leave	0.7642	0.9901	0.41	0.43

EVALUATION

To evaluate the performance of ECD, we use the *ns-2* simulation tool to run a number of simulations. We randomly placed 50 sensors in a 200m by 200m square region and randomly chose one node as a sink node. A constant bit rate (CBR) was used as our traffic sources for both anycast and multicast traffic. We compare the performance with Directed Diffusion (DD) [2], DSR, AODV, OLSR, and MAODV [1].

Fig.1 illustrates the results when the sink collects data. Packet delivery ratios (PDRs) of ECD (Fig.1(a)) are better than all of other protocols, especially at high offered load, because nodes in ECD encounter less congestion and contention. ECD also takes shorter paths (Fig.1(b)) which mean less number of transmissions, thereby it achieves more energy-efficiency.

Table 1 shows the results when the sink disseminates data. PDRs of both MAODV and ECD are comparable except very low PDR of MAODV in leaving scenario. ECD is energy-efficient scheme because the average numbers of transmissions are lower than those of MAODV.

CONCLUSION

This paper demonstrates that ECD efficiently collects and disseminates data across multi-hop, wireless sensor networks. It works and adapts to any changes of the network autonomously, and also satisfies the requirements of ubiquitous computing environments.

REFERENCES

1. Abolhasan, M., Wysocki, T., and Dutkiewicz, E. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks* 2, 1 (Jan. 2004), 1-22.
2. Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed diffusion: a scalable and robust communication paradigm for sensor networks, in *Proc. of MobiCom 2000*, (Aug. 2000), 56-67.

BIOGRAPHY

Niwat Thepvilojanapong is currently a Ph.D. student at the University of Tokyo working under the supervisions of Prof. Yoshito Tobe (Tokyo Denki University) and Assoc. Prof. Kaoru Sezaki (University of Tokyo). His research interests include ubiquitous computing, wireless sensor networks, mobile ad hoc networks, and network measurement.