

PROTOCOLS FOR LOCAL DATA DELIVERY IN WIRELESS MICROSENSOR NETWORKS

*Zhao Cheng, Mark Perillo, Bulent Tavli
and Wendi Heinzelman*

Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627–0126

Sameer Tilak and Nael Abu-Ghazaleh

Department of Computer Science
Binghamton University
Binghamton, NY 13902–6000

ABSTRACT

Sensor networks are becoming increasingly important as tools for monitoring remote environments. As sensors are typically battery-operated, it is important to efficiently use the limited energy of the nodes to extend the lifetime of the sensor network. Two factors can greatly influence the performance of protocols for these networks—the data delivery model, which describes how the end user wants to access the data, and the network dynamics, which include sensor mobility as well as changes in sensor data rates throughout the lifetime of the network. In this paper, we look at several media access control protocols for sending data from sensors to a local data collector. Comparing these protocols shows that there is an inherent tradeoff in energy efficiency with adaptability of the protocol.

1. INTRODUCTION

The goal of sensor networks is to monitor an environment for the presence of physical phenomena. Attention has recently turned to wireless microsensor networks, consisting of dozens to hundreds of small, battery-operated sensor nodes. These types of sensor networks require protocol architectures that are energy efficient in order to extend the lifetime of the sensor network. Furthermore, protocols must be able to handle network dynamics, including node mobility, changes in the sensors' data rates, and movement of phenomena of interest through the network. As we will see in this paper, these tend to be conflicting goals.

There has been much research recently related to application-specific protocol architectures for wireless microsensor networks [1, 3, 7, 8]. This research has focused on how to obtain highly energy efficient protocols for data transfer in large-scale sensor networks. At the same time, several wireless communication standards have emerged, such as the IEEE 802.11 wireless LAN (WLAN) standard [5] and the Bluetooth personal area network (PAN) standard [2]. If sensor networks can be built on top of standardized protocols, these networks can interact with other types of networks and benefit from the reduced cost of radios. While there is an increased flexibility in using standardized approaches compared with application-specific approaches, the tradeoff is generally a loss in energy efficiency.

In this paper, we examine several different protocols we have developed in the context of sensors sending data to a central data collector in a dynamic environment. Consider an event-driven data delivery model [10], where nodes adjust their sending rates based on events they detect locally. If no events of interest are detected, a sensor will stop sending data, whereas if one or more events of interest are detected, the sensor will send data, possibly varying its

data rate depending on what is sensed. Similarly, when sensors are mobile, a sensor may be out of range of the central data collector due to its mobility or the central node's mobility. All of these factors will impact the performance of different protocols.

To make efficient use of the limited sensor energy while handling network dynamics, the goals of local data delivery protocols are to:

1. introduce low overhead,
2. minimize the time sensors are awake,
3. reduce collisions,
4. support nodes joining/leaving the local network, and
5. efficiently reallocate bandwidth as sensors' data rates change.

While it is well-known that TDMA scheduling is energy-efficient and achieves many of the goals above, there are different ways to employ TDMA scheduling that tradeoff overhead to determine and disseminate the schedule with reaction to dynamic networks. The following section describes four protocols we have developed that achieve TDMA scheduling via a-priori scheduling, reservation-based scheduling, distributed scheduling, and controller-learned scheduling. We will present the protocols and then compare them in terms of their ability to react to changes in network connectivity and sensor data rates as well as their energy efficiency.

2. THE PROTOCOLS

In this section, we discuss four different protocols for sending data from sensor nodes directly to a central data collector/cluster-head node. A comparison of these protocols will show that there is an inherent tradeoff in adapting to changes within the network with overhead in the protocol.

2.1. LEACH-EM: LEACH for Event-Driven Data and Mobile Nodes

One clustering architecture that is used for large-scale microsensor networks is called LEACH [7]. LEACH is a cluster-based protocol that uses time-division multiple access (TDMA) for intra-cluster communication between the sensors and the cluster-head. When clusters are formed in LEACH, the cluster-head node creates a schedule that gives each sensor in the cluster a time slot in which to transmit its data, allowing the sensors to sleep for all other time slots. LEACH includes mechanisms to allow the high-energy position of cluster-head to be rotated among the nodes to

evenly distribute the energy load. Clusters are adapted based on the cluster-head nodes for each round, and new TDMA schedules are created based on the new clusters.

LEACH has been shown to achieve good energy-efficiency and hence long network lifetime when sensors always have data to send and when sensors are static. However, if sensors enter or leave the cluster area or change their data rate due to detection of phenomena while the cluster is fixed, LEACH cannot adapt.

To make LEACH better able to adapt to network dynamics, we created the LEACH-EM protocol. In LEACH-EM, the sensors only send data to the cluster-head during their TDMA slot if they determine that they have sensed an event of interest. Furthermore, in order to track mobility within the cluster-based framework of LEACH, cluster-heads in LEACH-EM transmit “Hello” messages at predefined intervals. These Hello messages are used as feedback by the cluster members to decide whether they can reach the cluster-head or, because of mobility, they are now out of range of the cluster-head. This decision can avoid many wasteful transmissions that would otherwise occur when the node has moved out of range of the cluster-head but transmits its data anyway.

LEACH-EM assumes symmetric links such that if members cannot hear the cluster-head then the cluster-head cannot hear the members. If a node does not hear the Hello message at a given time, then it assumes it has lost connectivity with its cluster-head and it does not transmit during its TDMA slot in the following frames. However, the member still listens for the Hello message at the expected time, regardless of whether it heard the Hello message at the previous time, to ensure that it is still not in communication range of its cluster-head. There is a tradeoff in how often to transmit Hello messages—the more frequently these are transmitted, the fewer wasteful transmissions occur when the node is out of range of the cluster-head, but the higher the overhead.

The LEACH-EM protocol uses a-priori scheduling of the nodes, with beacons to inform nodes not to transmit if their transmission will not be received by the cluster-head. Thus it achieves the energy-efficiency goals defined in Section 1. However, LEACH-EM does not allow dynamic reconfiguration of the cluster, nor does it allow reallocation of the bandwidth if only a few sensors are connected or sensing the phenomenon. Thus, if new nodes enter the cluster, they cannot transmit data to the cluster-head. Similarly, if nodes leave the cluster, their TDMA slot is still reserved for them, rather than using the available bandwidth to support other sensors. The TRACE protocol defined next solves these problems at the expense of increased overhead.

2.2. TRACE: Time Reservation using Adaptive Control for Energy Efficiency

TRACE is an energy-efficient dynamic, reservation-based TDMA protocol. In TRACE, time is broken into frames consisting of two sub-frames: a control sub-frame and a data sub-frame, as shown in Figure 1. The control sub-frame consists of a beacon message, a contention slot, and a header message. The controller node transmits the beacon message to announce the beginning of a new frame. Upon hearing the beacon message, nodes that have data to send, but did not have reserved data slots in the previous frame, randomly choose sub-slots of the contention slot on which to transmit their requests. If the request is successful (*i.e.*, no collision), the controller grants a data slot to the contending node.

Following the contention slot, the controller sends the header message, which includes the data transmission schedule of the cur-

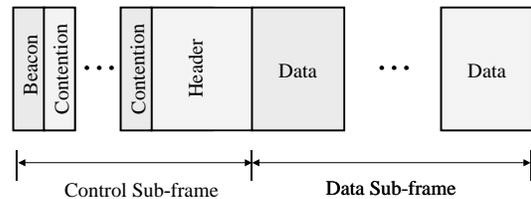


Figure 1: Frame structure for TRACE.

rent frame. The transmission schedule is a list of the nodes that have been granted data slots in the current frame along with their corresponding data slot numbers. A contending node that does not hear its ID in the schedule understands that its request was unsuccessful and contends again in the following frame.

A node keeps a data slot once it is scheduled for transmission as long as it has data to send. This feature enables a node to transmit long streams of data without interruption and is borrowed from the Packet Reservation Multiple Access (PRMA) protocol [6]. A node that does not transmit on a slot that it is scheduled for will not be granted channel access in the next frame (*i.e.*, it must contend to get a data slot again).

Using the reservation approach in TRACE requires overhead in every frame for contention and to transmit the schedule to nodes in the network, but it enables contentionless medium access in the data sub-frame, thus letting nodes sleep when they are not transmitting. Furthermore, the header allows TRACE to automatically adapt to network dynamics.

2.3. PBP: Predictive Backoff Protocol for IEEE 802.11

Rather than having a central node create and disseminate the TDMA schedule as in LEACH-EM and TRACE, it is possible to have the nodes themselves figure out when they should transmit, creating a TDMA schedule in a distributed manner. We have designed a protocol called Predictive Backoff Protocol (PBP) [4] that tries to achieve distributed TDMA scheduling. PBP is based on the IEEE 802.11 two-way handshaking (DATA-ACK) MAC protocol. Rather than pick a backoff time *after* transmission, as in 802.11, in PBP, nodes pick their backoff time *before* transmission and piggyback this information onto their DATA packets. In this way, the sensor informs other sensors when it may start its next transaction after the successful completion of the current transaction. The destination (central node in our scenario) repetitively piggybacks this information in the ACK frame to help some hidden nodes learn about this backoff time information. The listening sensors keep a registration entry for the sensors they overhear. Knowing when other sensors will start to transmit again, they avoid collisions by picking their backoff value from one that is not reserved by other sensors. As shown in Figure 2, PBP achieves much better throughput than 802.11 and close to optimal throughput.

When sensors join and leave the network, the channel throughput is not severely affected. PBP guarantees that no two nodes will collide twice consecutively. Thus a new node may only collide when it first joins the network. PBP allows reconfiguration of channel resources by easily supporting sensors joining or leaving the local network and efficiently allocating bandwidth only to sensors that need it. The draw-back to this approach is that it is not

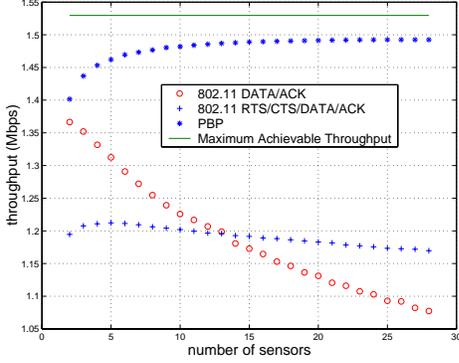


Figure 2: Example showing throughput improvement using PBP.

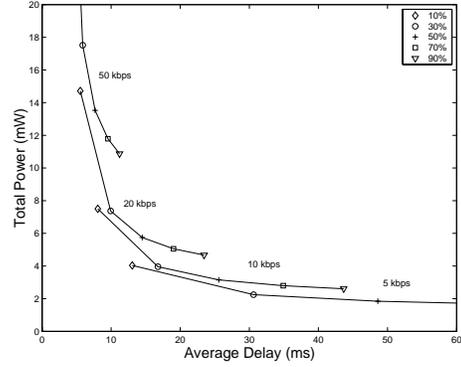


Figure 3: Power versus delay tradeoff using ASP.

very energy-efficient, requiring extra overhead to announce back-off states and send ACK messages for every DATA message, requiring sensors to always be awake and receive all messages to ensure their registration tables are accurate, and not providing any guarantee that collisions will not occur.

2.4. ASP: Adaptive Share Polling for Bluetooth

The previous approaches relied on the sensors themselves informing either the central node or other nodes about their data requirements (in the form of reservations). An alternative approach is to have the central node *learn* about changes in the data requirements of the sensors. We have created a Bluetooth scheduling protocol called ASP that does just that [9].

Bluetooth is an emerging wireless communications standard whose low-power consumption makes it suited for sensor networks. Local traffic in a Bluetooth piconet is controlled by the piconet master, which polls its slaves during certain time slots to which all piconet members are synchronized. The scheduling algorithm the master uses to poll the slaves can have significant impact on the energy efficiency of the protocol. We have developed a new polling approach called Adaptive Share Polling (ASP), which was designed as a means to reduce power consumption when traffic patterns consist of short packets from constant bit rate applications [9]. ASP balances the energy versus latency tradeoff that is inherent in any polling type network by ensuring that the polling success rate remains within a pre-specified target range. The scheduler at the master implicitly learns the share of the bandwidth that needs to be allocated to each of the slaves in order for the success ratio to remain within the target range and polls the slaves at the necessary rate in order to ensure that they are given their respective allocated shares. ASP is capable of adaptively changing the allocated shares based on observations of polling successes and failures.

ASP over Bluetooth allows the master to automatically account for changes in sensor rates on a node-by-node basis, without requiring explicit communication between the slave and the master. As shown in Figure 3, choosing different success ranges allows the master to make different tradeoffs in energy versus latency. Using ASP as the poller for Bluetooth achieves the goals of energy-efficiency, at both the slaves and the master, no collisions, and efficient reallocation of bandwidth as sensors' data rates change. However, it does not account for nodes entering or leav-

ing the piconet. Furthermore, it will not be as energy-efficient as an application-specific approach.

3. DISCUSSION

The previous section described four very different protocols that allow sensors to send their data to a central data collector. In order to evaluate these protocols, consider an "ideal" protocol, where each sensor that has data to send wakes up only to send its data and goes back to sleep immediately after sending its data. To accomplish this requires that sensors wake up at different times, using some sort of TDMA scheduling. While a fixed TDMA schedule requires the minimum amount of overhead, and is hence the most energy-efficient approach, it does not account for network dynamics and will not allow new nodes to enter the network or reallocation of the bandwidth as sensors' data rates change. On the other hand, a continuously changing TDMA schedule can track sensors and their changing data rates but requires a large amount of overhead.

If we compare the protocols in Section 2, we see that LEACH-EM requires very little overhead, making it the most energy-efficient approach. However, its only adaption to a dynamic environment is via Hello messages which allow sensors to know they are no longer connected to the central node. TRACE can adapt better to a dynamic environment, at the cost of added overhead for reservation contention and transmission of the updated schedule. LEACH-EM and TRACE are both TDMA-based protocols, where nodes are informed of the schedule (once or at the beginning of each frame). PBP, on the other hand, sets up a TDMA schedule in a distributed manner, by having the sensors themselves learn when they should transmit rather than being told when to transmit by the central node. This is the most flexible approach, as there is no frame constraint and hence the network can support arbitrary numbers of nodes and there is no need for central control, but it requires a large amount of energy due to all sensors listening to others' transmissions. Finally, Bluetooth uses another type of scheduling, namely master polling. In Bluetooth, sensors do not know when they will be polled and hence must remain awake and waste energy. ASP adapts the Bluetooth standard to inform sensors of when they will next be polled, allowing them to sleep and save energy. While ASP can adapt to changing data rates at the nodes, the added flexibility of supporting the Bluetooth standard means that it will not be as energy-efficient as application-specific

approaches.

In order to quantify the tradeoff between energy efficiency and adaptability of the protocols, we ran simulations using a 25-node network with random node mobility, where nodes enter and leave the transmission range of the controller node, and LEACH, LEACH-EM, TRACE and PBP. We could not run similar simulations using ASP due to the Bluetooth limitation of a maximum of 8 nodes per piconet. Figure 4 shows the data received by the controller over time, the energy dissipation over time, and the data per unit energy. As expected, when there is little change in connectivity at the beginning of the simulation, LEACH is the most energy-efficient, followed by LEACH-EM, since these protocols have the least amount of overhead. However, as soon as nodes start moving outside the transmission range of the cluster-head, in LEACH, nodes still send data to the cluster-head and thus waste a large amount of energy sending data that never reaches the cluster-head. On the other hand, nodes in LEACH-EM realize they can no longer reach the cluster-head and stop transmitting their data, thereby saving energy. However, as new nodes enter the transmission range of the cluster-head, in LEACH-EM, these nodes cannot participate in data transfer, while in TRACE and PBP, these new nodes are immediately incorporated into the network. Thus, while LEACH-EM achieves the highest data per unit energy, only TRACE and PBP actually ensure all the data is received at the controller. As PBP requires all sensors listen to each others' transactions to learn of reserved slots, it requires much more energy than the other protocols (approximately a factor of 25).

4. CONCLUSIONS AND FUTURE WORK

We have examined several different ways of creating a TDMA schedule to allow sensors to send data to a central controller—a-priori scheduling, reservation-based scheduling, distributed scheduling, and controller-learned scheduling. We have shown that there is an inherent energy/flexibility tradeoff in the different approaches to sending data. Therefore, it is very important to match the protocol for a sensor network to the application of the sensor network. In sensor networks that are relatively static, a fixed scheduling approach like LEACH or LEACH-EM may be appropriate, but in highly dynamic approaches, a more flexible technique such as TRACE, PBP, or ASP will provide better support to the application.

5. REFERENCES

- [1] J. Agre and L. Clare. An Integrated Architecture for Cooperative Sensing Networks. *Computer*, 33(5):106–108, May 2000.
- [2] Bluetooth Project. <http://www.bluetooth.com>, 1999.
- [3] Bult, K., et al. Low Power Systems for Wireless Microsensors. In *1996 International Symposium on Low Power Electronics and Design*, pages 17–21, August 1996.
- [4] Z. Cheng and W. Heinzelman. PBP: A Predictive Backoff Protocol for IEEE 802.11 Networks. In *In submission*, 2002.
- [5] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std. 802.11-1997*, 1997.
- [6] D. Goodman, R. Valenzuela, K. Gayliard, and B. Ramamurthi. Packet Reservation Multiple Access for Local Wire-

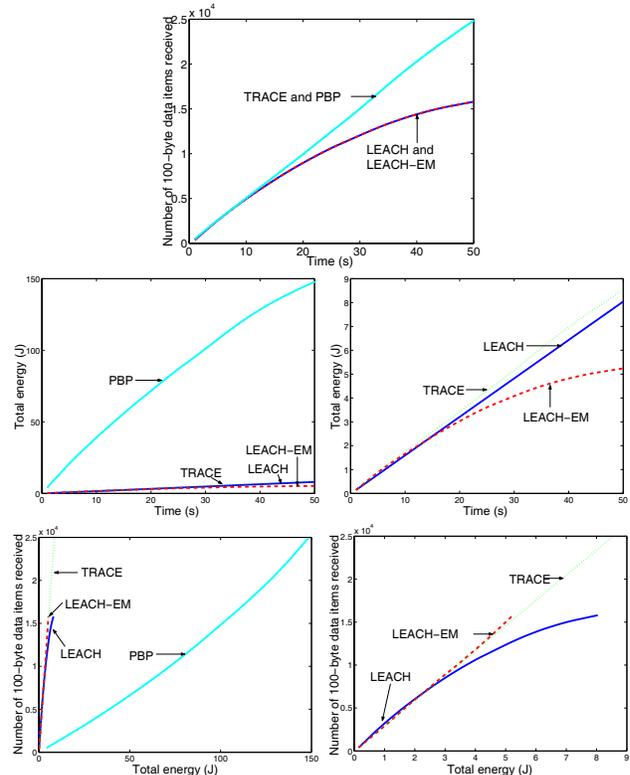


Figure 4: (a) Data transmitted to controller node over time. (b) Energy dissipated in the network over time. (c) Energy dissipated in the network over time (without PBP). (d) Data per unit energy. (e) Data per unit energy (without PBP).

less Communication. *IEEE Transactions on Communications*, 37(8):885–890, August 1989.

- [7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. In *To appear: IEEE Transactions on Wireless Communication*, 2002.
- [8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *Proceedings of ACM Mobicom '00*, Aug. 2000.
- [9] M. Perillo and W. Heinzelman. ASP: An Adaptive Energy-Efficient Polling Algorithm for Bluetooth Piconets. In *In submission*, 2002.
- [10] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. A Taxonomy of Wireless Micro-Sensor Network Models. *Mobile Computing and Communications Review (MC2R)*, June 2002.