DESIGN ISSUES: DUTY CYCLE CONTROL IN WIRELESS SENSOR NETWORKS

V.K. Sharma 1, Ravindra Prakash Gupta 2

1 Bhagwant Institute of Technology, Muzaffarnagar, India, E-mail: viren_krec@yahoo.com.
2 Bhagwant University, Ajmer, India, E-mail: rpg1@rediffmail.com.

ABSTRACT

Wireless sensor networks are envisioned to be deployed in the physical environment to monitor a wide variety of real-world phenomena. Wireless sensor networks (WSN’s) are becoming popular in military and civilian applications such as surveillance, monitoring, disaster recovery, home automation and many others. Almost any sensor network application faces the challenging design constraints of efficient energy consumption. The report presents concept of basic duty cycle control algorithms. Further studies will stress on the shortcoming mentioned in duty cycle control algorithms SIPF under different types of traffic conditions applied to the WSN and how different types traffic structures affect the SIPF algorithm will be examined.


1. INTRODUCTION

Advances in wireless communication and micro-electro-mechanical systems (MEMS) have led to the development of implementation of low-cost, low power, multifunctional sensor nodes and networks. A wireless sensor network (WSN) is a wireless network consisting of large number of spatially distributed tiny immobile autonomous devices or sensors with sensing, computation, and wireless communications capabilities to cooperatively monitor physical characteristics or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

The sensor node is small microelectronic device that communicate un-tethered in short distances. The information from sensors is aggregated on a “data centric basis”. Since a node is a microelectronic device, it can only be equipped with a limited power source (<0.5 Ah, 1.2 V) and it is not realistic to recharge or replace the batteries, therefore power consumption should be minimized so that overall network lifetime will be increased.

2. PRESENT WORK

This report, gives a brief outline of WSN and its applications and design issues in duty cycle control algorithms for WSNs.

3. DESIGN ISSUES: DUTY CYCLE CONTROL IN WIRELESS SENSOR NETWORK

The key design considerations for duty cycle control protocol design are scheduling and routing.

3.1 Scheduling

In order to maintain a connected network topology, to guarantee the delivery of the packets by scheduling the sleep schedules of the nodes between source and destination, the MAC layer protocols have to be carefully designed.

The SMAC [6] protocol is proposed as a MAC algorithm in order to coordinate and synchronize the sleep/wakeup duty cycles. SMAC is basically a CSMA/CA protocol based on 802.11. To maintain the synchronization, each node broadcasts its schedule in a SYNC message periodically, so that the neighbors can update that information in their schedule tables. The problem of neighbors can never see each other, which can be caused by SYNC message corruption, interference, or medium kept busy and SYNC packets can not be sent in time, is overcome by periodically followed neighbor discoveries. The SMAC does not require all nodes to be synchronized, only the nodes belonging to the same virtually constructed cluster have to be synchronized, however the border nodes have to maintain more than one schedule. The scheme works well with stationary network topologies in which frequent changes are not common.

Most of the MAC protocols have been proposed for stationary networks. The objective of the following MAC protocol is its ability to work energy efficiently in both stationary scenarios and mobile nodes. MSMAC [7] would work similarly to SMAC with stationary nodes.
In order to avoid the excess waiting time of mobile nodes in order to join a new cluster, each node discovers the presence of mobility within its neighborhood based on the received signal levels of periodical SYNC messages from its neighbors. If there is a change in a signal received from a neighbor, it presumes that the neighbor or itself are moving, and predicts the level of the mobile’s speed. The SYNC message in MSMAC also includes information on the estimated speed of its mobile neighbor or mobility information. If there is more than one mobile neighbor, then the SYNC message only includes the maximum estimated speed among all neighbors. This mobility information is used by neighbors to create an active zone around a mobile node when it moves from one cluster to another cluster, so that the mobile node can expedite connection setup with new neighbors before it loses all its neighbors.

Du et al. [8] proposed the algorithm in order to reduce endtoend latency with duty cycle MAC protocol. The nodes that are forwarding data has to be awake only when they are receiving or transmitting a packet. The protocol sends a small control frame along the data forwarding path in order to inform every node when to be awake in order to receive the packet. There are three stages of an operational cycle; SYNC, DATA, and SLEEP. Fig. 3.1 shows an overview of the RMAC algorithm.

In the SYNC stage, RMAC synchronizes the clocks on the sensor nodes. In the DATA stage, firstly a control frame is sent in order to initiate the traffic. PIONs namely a series of Pioner frames are used as control frames like RTS and CTS. A PION is for requesting communication from downstream, like an RTS frame and also used for confirming the communication to upstream like CTS. Using a PION in dual purpose increases the efficiency. During the SLEEP period, nodes go to sleep if they do not have a communication task, that is set by a PION. If they are stimulated with a PION, they must stay awake for a specific time in order to be able to receive and forward the packet. Completing its task, each node goes back to sleep state.

The cross layer scheduling algorithm for power efficiency [5] is proposed in order to conserve energy by turning off some sensor nodes. The idea is sensor nodes dynamically create on off schedules such that the nodes will be awake only when they are needed. The scheduling and routing schemes work separately. There are two phases of the algorithm: The Setup and Reconfiguration Phase and the Steady State Phase.

**The Setup and Reconfiguration Phase:** It is initialization of the network to update the network routes and queries. This phase is relatively short; its goal is to set up the schedules that will be used during the steady state phase. The setup and reconfiguration algorithm is independent of the underlying routing algorithm. Therefore, many of the algorithms available for routing in ad hoc and sensor networks can be used. Power aware routing algorithms may be preferable, as they have been shown to provide substantial increases in network lifetime.

**The Steady State Phase:** It is similar to forwarding phase. It utilizes the Schedule established in the setup and reconfiguration phase to forward the data to the base station. Each node stores a schedule table. The scheduling for sleep and active states are calculated according to the packets that the nodes will transfer. There are three different actions considered in this paper are: Sample, Transmit, and Receive.

3.2 Routing

Putting nodes to sleep affects network layer, because the sleeping nodes are no longer the part of the network, so they can not participate in the routing. Moreover there will be topology changes caused by sleep schedules. A link between two nodes will be active if and only if both nodes are active. The path selection has to be carefully engineered, because the algorithm affects the latency and power consumption.

“A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management” [9] algorithm is proposed, in order to construct the approximate topology of the network, using neighborhood information and putting the redundant nodes to sleep. These nodes logically organize the network in the form of clusters comprised of nodes in their neighborhood. TopDisc forms a Tree of Clusters (TreC) rooted at the monitoring node, which initiates the topology discovery process.

The “topology discovery request” message floods through the network; every active network receives the message. The node receiving the “topology discovery request” may respond this message in two different ways: Direct Response (i.e. every active node receives the request, forwards it to one of its neighbors, and immediately sends back a response with its neighbor list along the reverse path) or Aggregated Response (i.e.
before sending a response, it waits its child nodes’ responses in order to aggregate the responses then sends back to its parent).

The TopDisc algorithm is a hierarchical tree based clustering scheme gathers neighborhood information from all sensor nodes. However, this protocol provides only partial link information. The algorithm assigns the nodes their duties and gets the topology of the network. Then, the idle nodes are selected and they put to sleep, and duty cycle assignment is done. The main idea is to employ minimum required number of awake nodes, and the rest of the nodes are sleeping.

“Adaptive Sleep Discipline for Energy Conservation and Robustness in Dense Sensor Networks” [10], propose a randomized algorithm to provide robustness to the variations in network connectivity. The algorithm does not require nodes to keep any state information about their individual neighbors. Each node independently decides when to sleep and wakeup, based on local observations.

The main constraints considered are latency, capacity (i.e. ability to carry a certain load), employing no global timeslots or coordination with neighbor. If the estimated activity is too low to satisfy the delay constraint, the node decides to wake up more often. Conversely, if the activity is higher than necessary, the node decides to sleep longer.

REFERENCES