**IT-478 INTERNET OF THINGS**

**PROTOCOL DESIGN ASSIGNMENT**

Submitted by:

Krishanu Konar (201401127)

Kunal Suthar (201401131)

Virendra Pratap (201401404)

**ABSTRACT:**

In the following assignment, we extend and augment upon the Leach Protocol and devise a new approach for communication in wireless sensor networks (WSN). This Wireless sensor network consists of various heterogeneous sensors, all transmitting packets of different sizes, and sending packets at different time intervals. These sensors monitor some remote environmental conditions. We would attempt to refine the LEACH protocol in the setup phase and the steady state phase, so as to better the energy utilization of the nodes, equal distribution of the nodes in the clusters, account for the heterogeneous nature of the various sensors’ data and try to accommodate it in a single protocol.

**INTRODUCTION**:

The primary motive of the protocol would be to transmit the data from the various clusters, to the base station. For employing it, we would first refine the cluster formation algorithm (the Set-Up Phase), prioritizing equal number of nodes in each cluster, and assigning a cluster head judiciously. Then, we would attempt to refine the data-transfer(the Steady State phase), which would efficiently model a frame and would minimize the waiting-time of transmission, between data of various nodes.

**ASSUMPTIONS**:

* Since we are modifying LEACH protocol, we would be taking into account, most of the assumptions of the LEACH protocol.
* Every node has enough power for the transmission of the data to the base station(BS). This is assumed because in case of an outlier node, there would be a cluster comprising of that sole node only.
* The nodes can use power control to vary the amount of transmit power.
* The sensor nodes and the base station are static.
* Sensor nodes are randomly distributed in the monitoring area and their distribution is neither too dense nor too sporadic.
* All Nodes have a GPS system, which would tell them their GPS coordinates.
* All nodes have different energy level, initially.
* Number of clusters to be formed and the number of nodes in each is predetermined.

**MODIFIED LEACH PROTOCOL :**

In the leach protocol, CH is distributed randomly and thus might result in the random load node distribution (A cluster may have large number of nodes as compared to other clusters). We also want to keep the energy of all the nodes within a cluster approximately same and hence try to achieve greater lifetime of the nodes. Apart from the energy considerations, we also focus on frame formation such that each node sends data in a periodic fashion and hence can be put to sleep during its inactive time to save energy. To further explain the working of the network, we assume that the WSN has the following phases:

1) Set-up Phase (CH Formation and reformation)

2) Steady State Phase

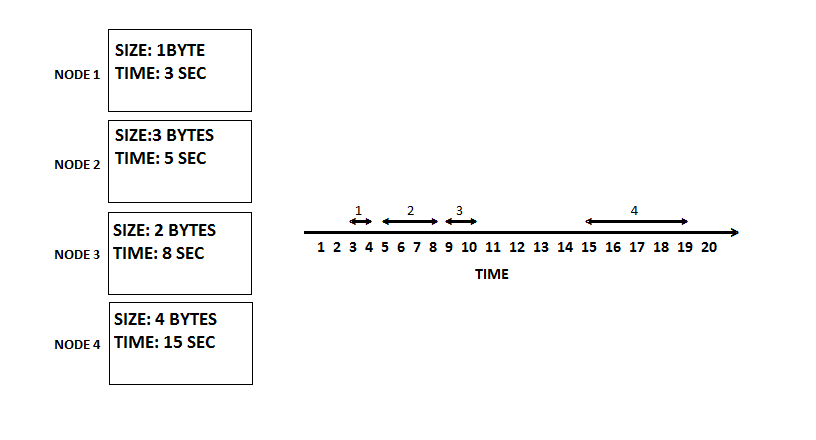
**1) SET-UP PHASE(CH Formation and Reformation):**

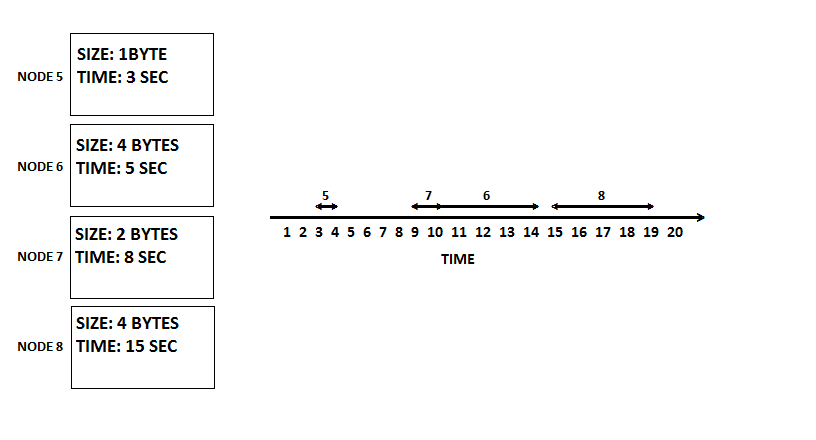
We try to achieve approximately equal number of nodes in each cluster. In this phase, every node will send their coordinates to the BS using the GPS system. BS will divide the area in circular clusters, same as the number of the predetermined number of clusters. There could be a scenario, in which some of the clusters made, would not entail the predetermined number of nodes. So, for deploying equal number of nodes in the cluster, we would employ a center based technique, which would help us in assigning a CH also. We would compute the average X coordinate and average Y coordinate in the cluster. The AVG(x,y) would be the centre of the given cluster. Now, if there is a dilemma regarding which cluster a node should go into, then the node would be a part of the cluster, whose distance from the AVG(x,y) is lesser. For assigning the CH also, we would take into account the node’s distance from the AVG(x,y) of the cluster(for good reachability) and the node’s usable power. Nodes having lesser distance from AVG(x,y) and having comparatively higher power levels would be given higher priority for becoming the CH.

Once clusters are well formed and CH is selected, TDM scheduling is done by the CH. Each node is allotted a slot depending on its generation frequency and size of data. The time slot given to a node is allotted by the following algorithm.

Each node has 2 parameters: Size of the packet and frequency of packet generation. We select packet from every node once in a frame. Considering all nodes are synchronised at the start, minimum duration of the round would be the max. time taken by a node to generate its data + the actual data transmission time. We also assume that 1 byte of data is sent in a given unit time slot. Now, for scheduling, we prioritize the nodes with the least data generation frequency. This is because it would increase latency before the data is generated again. We set the number of slots needed equal to the size of data packet. So, we first allot the slots to the node with low frequency. In case of a clash, we allot the higher frequency node the next possible slots. This approach on an average schedules the node to send the data in an efficient manner. We take following example to understand the algorithm.

Here, we consider 4 nodes in a given cluster, with the size of the packet and time after which the data is generated. The scheduling for both examples is given below:





In the first case, while allotting slots in decreasing order of the generation time does not cause any clash, so the scheduling is straightforward. But in the second case, the data generated by node 6 cannot be sent at 5th slot, as it would take 4 slots to send the complete data and hence clashes with node 7’s slots. If we allow node 6 to send data at that slot, we have to reschedule node 7’s slot. Node 7 generates data every 8 time slots, so next time, data is generated on 16th slot, which clashed with node 8’s slot, which if again rescheduled would generate data again at 30th time slot, effectively increasing latency. By using our algorithm instead, we find the next time slot where the data of the higher frequency data node(node 6) is generated again, which is at 10th slot. The following 4 slots are free, so this node is allotted that slot. This makes sure our scheduling is optimised.

After the scheduling is done, the steady state phase occurs.

**2) Steady-state phase**

The steady-state operation is pretty much similar to that of LEACH. The steady state operation is broken into frames, where nodes send their data to the cluster head at most once per frame during their transmission slot that was allocated to them in the previous phase. The duration of sending data from each node is different and is calculated in the setup phase. Also, we assume that the nodes are all time synchronized and start the set-up phase at the same time. This could be achieved, for example, by having the BS send out synchronization pulses to the nodes.

The cluster head must be awake to receive all the data from the nodes in the cluster. Once the cluster head receives all the data, it performs data aggregation. The resultant data are sent from the cluster head to the BS. Like in LEACH, to reduce inter-cluster interference, each cluster communicates using direct-sequence spread spectrum (DSSS).

**COMPARISON BETWEEN LEACH AND OUR MODIFIED LEACH PROTOCOL:**

The traditional LEACH protocol uses randomized cluster head rotation algorithm on the basis of the node’s energy to determine the CH for the next round, and it does not take into account, the heterogeneity of the nodes, i.e. different nodes transmitting different amount of data at different times, which will lead to the collision of the data frames, and would lead to data corruption. Our modified leach protocol takes the above mentioned loopholes into consideration, and also attempts to create equi-sized clusters. Hence, the LEACH protocol does not work in the given scenario.