Dual Tree Data Routing Scheme for Wireless Sensor Networks

Kayiram Kavitha1, Polsani Rajashee Rao2, Sreeja Tummala3, Gajarla Vasavi4 and Dr.R.Gururaj5
1 Dept. of CS & IS, BITS-Pilani, Hyderabad Campus, Hyderabad, India.
{kavitha@bits-hyderabad.ac.in, rajashrepolsani@gmail.com, 10812sreeja@gmail.com, vasavi.gajarla@gmail.com, gururaj@bits-hyderabad.ac.in}

Abstract—Wireless Sensor Networks (WSNs) play a vital role in applications like disaster management and human relief, habitat monitoring, studying the weather, eco systems etc. The source of power in WSNs is restricted to battery, as their location of deployment is usually remote. Hence the power conservation at each sensor node is desirable to increase the lifetime of the network. A significant amount of work has been done by researchers in the past to achieve energy efficiency in WSNs. In this paper, we propose a dual tree data routing scheme to optimize the power utilization in WSNs. Many of the WSN applications form a tree topology for communication. The intermediate nodes relatively closer to the root node tend to expend more energy when compared to the nodes closer to the leaves of the tree. In our proposed dual tree data routing scheme, we construct two data routing tree structures for a WSN, where each of which is built by distributing the workload uniformly among the nodes in the tree. The network switches between these two trees at specified intervals for its operations. According to this the role played by a node in one spell is reversed in the succeeding spell with respect to the level/depth of the node in the tree. This facilitates uniform distribution of load across the nodes, leading to longer network life. The proposed dual tree data routing scheme not only increases the lifetime of the network, but also utilizes the battery power optimally. Simulation results show a considerable increase in the lifetime, and effectiveness of the WSN.

Index Terms—Wireless Sensor Network, energy efficiency, tree construction.

I. INTRODUCTION

As the world is advancing towards remote sensing and reporting systems, the Wireless Sensor Networks (WSNs) [1] are in great demand. A wide range of applications of WSN are found in the fields like- military operations [2], medical (body sensing equipment), monitoring physical systems etc. In general, the deployment of such systems is done at remote places. As the nodes are battery operated, and are deployed in hard-to-reach areas, battery replacement is extremely difficult or impossible. Hence, at each node, the battery power available is limited, and it is desirable to utilize the power in an optimal way. Therefore, building an energy efficient network is a major objective in the design of WSN. A sensor node is a tiny electronic device with a small microcontroller, sensors to sense one or more physical properties, energy source (usually battery) and a radio transceiver for wireless communication. Depending upon the application, each sensor device is deployed in the physical environment, for monitoring certain conditions such as temperature, motion, sound, vibration, pollutants etc. A wireless sensor device can
communicate with other sensor devices within its transmission range. In real-time, as an application demands, many such wireless sensor devices collaborate to form a WSN.

In a WSN, the Base-Station (BS) is the node that connects the WSN with the outside world. Sensor nodes sense data continuously and then store in their built-in memory, and/or transmit to the BS. Hence, all queries from outside, first arrive at the BS, and are then routed to the concerned nodes in the network. Similarly, the results of a query will be sent back to the querying entity through the BS. We assume that the amount of computational, storage and energy resources available at a BS are very high, compared to other nodes of the network. The query/result transmission between the BS and the nodes in a WSN can be continuous or periodical. Many applications form tree topology [3] for communication in WSN, and as a result, each communication from the BS to a leaf node and vice-versa will follow a specified path. We call this communication tree structure as data routing tree. For the routing tree of a WSN, the BS is always considered as the root node.

In Fig. 1, a sample communication tree structure in a WSN is shown. The intermediate nodes are always taxed with extra burden of query dissemination [4], and results forwarding as well. The extra burden taken by an intermediate node is directly proportional to the size of its sub-tree. These nodes tend to expend more energy and die early. This may become even worse when the workload distribution among the nodes at same level is not uniform.

![Fig. 1. Tree topology for WSN](image)

In this paper, we propose a novel data routing scheme for WSN where two trees will be used during the lifetime of the network. But at a given point of time, only one tree is operational. Since we have two routing tree structures, we call this scheme as Dual Tree Data Routing (DTDR) scheme. Each of the above mentioned routing trees will have its own base-station, and these BSs are at the opposite ends of the topology. The network keeps on switching between these two routing trees for communication and data routing.

First, we construct the First Routing Tree (FRT) for the network, with a node as the base-station. Then we select one of the leaf nodes of FRT (normally predefined) to act as the base-station (BS) and construct the Second Routing Tree (SRT). In general, the nodes that play the role of a BS are predefined due to the reason that these base-stations will have very high computational, battery and storage resources at their disposal. According to the scheme, we assume that there are two such base-stations at opposite ends of the topology. While constructing the FRT and SRT, we adopt the workload-aware load-balancing technique proposed in our previous work [5]. The switching between FRT and SRT is based on the power levels in the network, and is explained in detail in succeeding sections. This dual tree data routing approach increases the lifetime of the WSN, by optimising the battery power utilization at each node.

The rest of the paper is organized as follows. We discuss the related work in Section 2, and Section 3 describes the proposed DTDR scheme with algorithms. Section 4 shows the simulation setup and results. Finally, we conclude the paper with Section 5.

II. RELATED WORK

In this section, we try to give the brief summary of various techniques that are existent for WSNs for constructing energy efficient routing trees.
The Energy-driven Tree Construction (ETC) algorithm [6] uses First-Heard-From (FHF) [7] approach in its first phase, to construct a query/result routing tree for a WSN, where each node after hearing from other nodes within its transmission range, selects one among them (from where it has heard first) as its parent. Each of the remaining nodes will become an Alternate Parent (AP) for that node. This process is continued till all the nodes are added to the tree. The list of such alternate parents for a given node is maintained as the Alternate Parent List (APL). Each node stores its APL and Child Node Lists (CNL) locally. In ETC algorithm, the maximum number of children for a node is indicated by Branching Factor ($\beta$), which is considered to be the threshold value to indicate the maximum number of children a node can have. The value $\beta$ is calculated at the base-station using the formula $d/\sqrt{n}$ where $d$ is depth of the tree and $n$ is number of nodes in the tree. Later, in the second phase, this $\beta$ value is disseminated to all the nodes in the tree for load-balancing. The candidate nodes for balancing are those which exceed the threshold ($\beta$). The excessive workload of the candidate nodes is distributed amongst the other nodes of the tree. The candidate node instructs some of its child nodes to look for a new parent. Then such child nodes will select the first node of their respective APLs as their new parent. This process of balancing the workload continues for all nodes of the tree.

The major drawback in ETC approach is that, while reorganizing the tree, if a node’s branching factor exceeds the threshold ($\beta$), some of the children will be attached to a new parent (picked from the APL) whose branching factor is less than the threshold. Unfortunately, this algorithm will not choose the parent with minimal branching factor; instead it just selects the first possible node that satisfies the required conditions in terms of branching factor, from the APL as its parent. As a result, in the reorganized tree, some nodes are overburdened while some are under loaded.

Our earlier work on WSN data routing, proposed an approach called Workload-Aware Tree Construction (WTC) scheme [5]. The WTC scheme uses an efficient strategy to construct a query routing tree which optimally balances the workload among the nodes at a given level. In this scheme, the workload of every node is computed based on the number of child nodes. As the nodes are arranged level-wise, the average workload associated with each level is computed to compare it with workload of the individual nodes at that level. If a node is found to possess workload more than that of the average workload, then some of its child nodes can be attached to other nodes at same level with the least workload. This way, a tree is balanced with respect to workload, having the load distributed uniformly among the nodes at a given level. In the proposed Dual Tree Data Routing scheme described in this paper, we have adopted WTC approach for constructing the routing trees.

III. PROPOSED TREE-CONSTRUCTION SCHEME

In this paper, we propose a Dual Tree Data Routing (DTDR) Scheme, whose objective is to optimise the power utilization in WSN.

A. Dual Tree construction and Load-balancing

Now, we describe the process of constructing two distinct routing trees for a WSN. As already mentioned, in our proposed DTDR approach, we have two base-stations one each for FRT and SRT. The base-station of FRT is called as First Base-station (FBS) and similarly we have Second Base-Station (SBS) for the SRT. As per our assumptions we make sure that the SBS is at the far end of the FBS.

We first construct the first data routing tree using FHF approach. In this approach, the FBS (the root of FRT and is at level 0) sends a ‘hello’ message to the nodes within its transmission range to form level 1 nodes. All the receiving nodes respond with ‘ack’ message, and become the children of FBS. This forms the level 1 of the FRT. Now, these child nodes of FBS forward the ‘hello’ message further to nodes within their transmission range which are not already in the tree. In this process, each node receiving ‘hello’ messages from multiple sources selects a node as its parent from where it has received the ‘hello’ message first and sends ‘ack’ message confirming the parent-child relationship. This process of ‘hello’ and ‘ack’ messaging is repeated till all the nodes are connected to form a tree. Once the initial tree is constructed then we apply the workload balancing technique as specified in Algorithm 3.1 and which is proposed by us in our earlier work [5]. The complete tree structure for the FRT is stored in a suitable data structure.

We now, consider one of the leaf nodes of the first tree structure (usually at the far end of FBS) as Second Base-Station (SBS). From SBS we initiate the tree construction process using FHF as described above to construct SRT. We apply WTC technique to balance the workload in SRT. Further, we assume that both FBS
and SBS have equal capabilities in terms of computing power, energy and storage. Thus we have two distinct routing trees namely FRT and SRT for the WSN.

Algorithm 3.1: This Algorithm uses breadth first technique (for level-wise traversal) to balance the tree. Standard Queue operations enqueue and dequeue are used. Procedures workload(), averageWorkload() are followed in the same section.

Procedure Workload_balancing(BaseStation)

Begin
    Initialize a queue;
    enqueue(BaseStation);
    while (queue != EMPTY)
        temp=dequeue();
        for each node € child(temp)
            enqueue(temp);
        if (WorkLoad(temp) > 1.5*AverageWorkLoad(temp.level))
            balance(temp);
    End if;
    End for;
    End while;
End.

Procedure 3.1: Workload(node)

This procedure returns the workload of a node.

var WL, ChildList;

Begin
    ChildList := ChildNodeList(node);
    If (|ChildList|==NULL)
        WL := node.level;
    Else
        size of ChildList
        WL := \sum_{i=1}^{size of ChildList} Workload(ChildList[i-1])+node.level;
    End if;
    Return WL;
End.

Procedure 3.2: AverageWorkload(level)

This procedure computes the average workload of a given level. (Assuming NODES as array of nodes at that level. The method getNodes(level) gives the list of nodes at that level, getWorkload(node) returns the computed stored value of workload of a given node and AWL is the average workload);

var k;

Begin
    NODES := getNodes(level);
    k := |NODES|;
    AWL := \sum_{i=1}^{k} getWorkload(NODES[i])/k;
    Return AWL;
End.

B. Dual Tree operations

The network starts functioning with FBS as its base-station, and the FRT as the data routing scheme. The FBS collects data continuously from all its child nodes, which forward their own data and the data received from the downstream (children). This network with FRT continues to work until the total residual power of the network becomes equal or close to 50% of the total power in the network that was available when FRT started its operations. The residual power in the network at any given point of time is the remaining power (sum of the power available at all the nodes) at that time.

As soon as the total residual power of the network becomes equal to or close to 50% of the total power in the network that was available when FRT started its operations, the network switches to the SRT. Now, the SBS will start playing the role of base-station for the network. This network with SRT continues to work until the total residual power of the network becomes equal or close to 50% of the total power in the network that was available when SRT started its operations in the recent spell.

This process of switching between two data routing trees continues throughout the network lifetime. The network stops functioning as soon as both of its base-station(s) become isolated (that means BSs become
orphans). This implies that the network can continue its functioning with only one tree in case the other one becomes orphan well before the second one. A node becomes orphan node when all its child nodes terminate their connections due to power outage. This dual tree switching scheme is depicted in the Algorithm 3.3.

Algorithm 3.3: Let node, be the \(i^{th}\) node in the network of ‘n’ nodes. ‘T’, ‘k’ be integers, residual_power(node,) returns the (current/remaining) power in the \(i^{th}\) node of the network, power(node,) returns the in the \(i^{th}\) node of the network, operate_tree(Base_station) performs the tree operation with Base_station as its base-station. CNL(node) returns the Child Node List of the node. FBS is First Base-Station 1, SBS is Second Base-Station.

Procedure Dual-tree(FBS, SBS)

Begin

\[ T = \sum_{i=1}^{n} \text{power}(\text{node}_i) \]

While((CNL(FBS) == NULL) && (CNL(SBS) == NULL))

\[ k = 2; \]

if \((\sum_{i=1}^{n} \text{residual}_\text{power}(\text{node}_i) > (T/k))\)

operate_tree(FBS);

else if \((\sum_{i=1}^{n} \text{residual}_\text{power}(\text{node}_i) \leq T/k)\)

operate_tree(SBS);

end if;

end if;

k = k + 2;

End while;

End.

C. Illustration to Dual Tree operations

Here, using an example, we illustrate the working of our proposed approach. We consider a set of 11 nodes for deployment and explain how the initial trees - FRT and SRT are constructed by FHF approach as given in ETC [6]. We apply WTC to construct the trees with load-balancing required for the dual mode. These two tree structures are separately stored for future operations.

In this example, we consider node 0, as the FBS and node 11 as the SBS. Now the process of construction of FRT starts with FBS as the base-station. The FBS sends a ‘hello’ message to all its neighbours (nodes within transmission range) i.e., nodes 1, 2, 3. Each of these child nodes reply with ‘ack’ message back to the parent node (FBS). This is to confirm the parent-child relationship in the tree. Now, nodes 1, 2, 3 send ‘hello’ message to other nodes in Fig. 2, and are the FRT of the network.

Fig. 2. First Routing Tree of the network.

Now, we choose node 11(SBS) which is one of the nodes in the last level of the FRT as the BS for the SRT which we plan to construct. The same process which was carried out for constructing FRT is used to construct SRT, and the resultant tree is shown in Fig. 3. In this research, we assume a routing model where the sensor nodes of the network keep on sending data to the BS using the routing structures formed as explained above. In the routing tree structure, the data packet sent by each node reaches the BS with multi-hop communication [8]. The nodes at the lowest level of the tree are termed as leaf nodes. They sense data and send to their respective parents. Such parents are termed as intermediate nodes. Therefore, each intermediate node apart from sensing and sending its own data also has to receive and transmit the data packets of its child nodes. All the data packets in the network are destined to reach the BS. We have observed that these intermediate nodes are taxed more than the leaf nodes in the tree structure. Even though we apply
WTC [5] to distribute the workload among nodes at the same level of the tree, we are unable to balance the workload of these intermediate nodes in an effective way. As a result of this, the intermediate nodes, closer to the BS, exhaust their power early and get disconnected from the network early. Hence, the overall network comes to a halt soon. In order to address the above issue, we propose a routing scheme in which we use two different tree structures as mentioned earlier, which become operational. Due to this, the role of the each intermediate node is changed after a fixed power interval. According to this, in one spell, node at level l will become node at level n-l (approximately) in the succeeding spell, where n is the maximum number of levels in the tree. This results in uniform power consumption at intermediate nodes in the network. Due to this, the intermediate nodes are retained in the network for a longer time.

![Second Routing Tree of the network.](image)

This helps in longer connectivity of the nodes to the network. Thus, it improves the overall lifetime of the network. The leaf nodes, which were having larger residual power previously, are now found to utilize the power in the tree operations heavily due to the change in the role in the succeeding spell, and vice-versa.

According to the example under illustration, FRT shown in Fig. 2, starts its operations of sensing and sending sensed data to the FBS till the network switches to SRT due to drop in the residual power levels. In the succeeding spell the network uses SRT for data routing as shown in Fig. 3. This tree carries on its operations with node 11 as SBS till it switches back to FRT. This way the network switches between FRT and SRT till both base-stations become orphans.

IV. PERFORMANCE EVALUATION

In this section, we present a brief report on the series of experiments we have conducted on our custom-built simulator to assess the effectiveness of the proposed DTDR scheme, and compare the same against the performance of the WTC which we consider as the most efficient single tree data routing scheme for WSNs.

A. Simulator

We have developed a custom-based simulator which is implemented using Java technology. Our simulator is meant for Windows platform and is console-based. This simulator allows us to define the geographical range of the network along with the number of nodes. We can also define the transmission range of each node. Since we plan to conduct experiments to assess the power consumption of our technique, we have also designed our simulator wherein the power allotted to each node can also be defined. The deployment of nodes using our simulator can either be random or predefined. Our simulator also facilitates the generation and assignment of workload (number of queries) to the network, which also can be either random or predefined.

B. Experimental set-up

The simulator was run for networks of sizes 20, 100, 200, 300, 400, 500, 600 and 700 nodes. The input to the simulator is the location of nodes specified by their (x, y) co-ordinates. The radio communication range of each node is set as 3m. Each sensor node is initialized with 1J of energy. The network is given some query workload so the energy at a node gets depleted gradually after each query. The queries are randomly generated at the BS. Each query is targeted to randomly selected nodes of the network. Here, the term querying means sending a dummy packet to a specified node. Further, a node responds to its query by sending a dummy packet back to the BS. Thus, we would like to make it clear that this experimentation is not
intended to query for the real data. Rather, we just focus on simulating the communication-load due to transmission of query/result.

C. Performance analysis

A set of experiments were conducted to compare the performance of our proposed DTDR scheme against the existing WTC [5]. The comparison is made based on the following metrics.

1. The Number of packets transmitted during the lifetime of the network.
2. The lifetime of the network.
3. The residual power available in the network after the network comes to halt.
4. The discarded packets in the network.

We have done our experimentation with networks of sizes 20, 100, 200, 300, 400, 500, 600 and 700 nodes. In each case, we simulated five different topologies. In this section, for each of the performance metrics is the average of the five simulation runs for the above mentioned networks with tree structures for data routing. We have computed: (i) number of packets transmitted in the network during its lifetime, (ii) network lifetime, (iii) number of discarded packets and (iv) residual power in the network (the amount of power left unutilized at the end of its lifetime), for each simulation. The average values of our results for each experiment are depicted in the graphs presented in this section. From Fig. 4, we observe that our DTDR scheme has shown a good increase in the number of packets transmitted, when compared to WTC. From the results it is evident that as the size of the network increases the total number of packets transmitted also increases.

The graphs in Fig. 5 show the network lifetime which we consider as a significant parameter to quantify the effectiveness of the network. The network lifetime is the time elapsed between starting of the network and the moment it halts. A network is considered to have reached a halt state when not even a single node is connected to the BS. Otherwise, we say that the network is alive if at least one node in the network is active and connected to the BS. The graph shows excellent results associated with our approach when applied on network with sizes 20, 100, 200, 300, 400, 500, 600 and 700 nodes. A significant increase in the network lifetime can be seen in networks with 500 and 700 nodes. This shows that our proposed approach provides better lifetime for the networks when compared to the earlier technique. We know that longer the lifetime, higher will be the energy efficiency of the network.

As the network spends maximum amount of energy for packet transmission, battery power is diminishing at each of these nodes after each transmission. Optimizing the battery power utilization is one of the major issues in WSN. We measure the power utilization of WSN by considering the residual power in the network after the network dies. The residual power is the sum of the residual powers at all the nodes, which are alive and not connected to the BS. The graph shown in Fig. 6, depicts the total residual power in the network. In the DTDR scheme, the nodes undergo role reversal (due to shift in the tree levels) after a power interval, enabling them to live longer when compared to WTC scheme and thus have an efficient way of utilization of power in the network. The graph in Fig. 6 confirms this speculation.
Now in Fig. 7, the WTC is compared to our DTDR scheme in terms of number of nodes alive after executing 500 queries in networks with varying number of nodes. We observe that our approach outperforms the WTC approach as the number of nodes alive is very high. This is due to shift in the roles played by nodes.

From the above experimental results, it is proved that DTDR scheme has optimal power utilization in comparison to WTC. It is also shown that the proposed scheme keeps more number of nodes alive when compared to the WTC. Hence, we conclude that DTDR scheme is successful in optimizing the energy
consumption, and increasing effectiveness of the network through the strategy of changing the roles of intermediate nodes after a power interval. According to this, an intermediate node handling high workload in one spell, will change its role with respect to its location in the structure of the tree and is bound to handle lesser loads in the succeeding spell. But, our experimentation with respect to the number of packets discarded in the network reveals the fact that the dual tree data routing scheme results in more number of discarded packets. The same is shown in Fig. 8. We guess that this behavior is due to random nature of workload assignment to the nodes in the tree. Nevertheless we plan to study this behavior in-detail.

Fig.8. Graph showing the number of packets discarded in the network

V. CONCLUSIONS

In this paper, we have proposed a Dual Tree Data Routing (DTDR) scheme for wireless sensor networks. According to this, we have two different routing trees for the network, wherein, each tree will have a distinct base-station. Further, these base-stations are located on opposite ends of the topology of the network. At a given point of time, only one tree is operational and the network alternates between these two routing trees depending on the power levels in the network. The result is that the nodes keep changing their roles with respect to the levels in the trees in which they participate. For instance, a node which is at level 1 in tree 1 may become a node at leaf level or close to the leaf level in the other tree, during the next spell. Due to this shift in roles, nodes live longer contributing to the prolonged lifetime of the network. This is because the load handled by a node in spell s is inversely proportional to the load handled in the preceding/succeeding spell. Two consecutive spells use two different/distinct routing trees. Our approach outperformed the earlier data routing scheme WTC in terms of number of packets transmitted and the network lifetime. This greatly minimizes the residual power in the network (on halt) and helps in optimal utilization of power in the data transmission operations.

REFERENCES


