TREEFP: A TDMA-based Reliable and Energy Efficient Flooding Protocol for WSNs

Chilukuri Shanti and Anirudha Sahoo,
Indian Institute of Technology, Bombay, Powai, Mumbai, India
Email: {shanti, sahoo}@cse.iitb.ac.in

***Abstract*—Flooding a network with a message from the sink is required for many purposes like synchronization, code dissemination etc. While several flooding schemes exist, only a few are designed to achieve the energy efficiency required by Wireless Sensor Networks (WSNs). In this paper, we present a TDMA-based Reliable and Energy Efficient Flooding Protocol (TREEFP) for WSNs. Slot assignment in TREEFP is done such that the time taken to flood the network is bounded to a single TDMA frame. TREEFP has a tunable system parameter which brings in tradeoff between reliability, flooding delay and energy consumption because when this parameter changes, the topology of the logical flooding tree also changes. We provide details of simulation experiments to compare TREEFP with other flooding protocols in the literature like FTSP, TDFS and MST. Simulation results show that TREEFP is better than FTSP and TDFS in terms of energy and flooding delay and comparable to MST in terms of those metrics. In terms of reliability, TREEFP is better than MST and comparable to FTSP.**

I. Introduction

Wireless Sensor Networks (WSNs) are increasingly being deployed for a wide range of potential applications such as environment monitoring, earthquake detection, patient monitoring systems etc. Since nodes have limited energy and they continue to operate until their energy is exhausted, applications and protocols for WSNs should be carefully designed in an energy-efficient manner so that the lifetime of nodes can be longer. The sensing element of a node probes the surrounding environment. If an *interesting* event is detected, after performing signal processing of the observed data, nodes communicate this data to the sink using a radio link. Sometimes the sink may need to send control messages to every node in the network, for the purpose of code distribution, time synchronization etc. This *flooding* of the network may be occasional or periodic. In either case, it should be done with minimum energy expenditure and maximum reliability. Flooding protocols that use the CSMA-based approach consume a lot of energy and cannot achieve flooding within a bounded period. A better approach to flooding is to use TDMA for transmission of control messages. To the best of our knowledge, only [1] proposes a TDMA-based flooding scheme, called TDFS, for synchronization. However, the scheme proposed in [1] is a probabilistic one, i.e., it cannot guarantee that a network will be flooded within a certain time period and the time taken to flood (or synchronize) the entire network is high. In this paper, we propose a TDMA-based Reliable and Energy Efficient Flooding Protocol (TREEFP) for WSNs. TREEFP uses a novel method of forming a spanning tree (rooted at the sink) to enable flooding. The topology of the spanning tree can be designed by varying the *level width coefficient* (described in Section IV) based on the *flooding delay*, reliability and energy consumption requirement. Flooding delay is the time required to flood the entire network so that all the nodes in the network receive the control message, from the time the sink originates a control message. Unlike TDFS, TREEFP employs a deterministic TDMA protocol and aims to minimize the flooding delay for a given flooding spanning tree by reusing slots whenever possible. It also carefully computes the sleep and wake up cycles of nodes to achieve reliability and energy efficiency.

Acknowledgements are widely used as means of reliable message delivery when the medium is lossy. Acknowledgements consume time and energy and do not solve the problem of downstream nodes not getting the message when a forwarding node dies. TREEFP aims to make the network as reliable as possible without using acknowledgements. Though TREEFP can be used to flood the network for any purpose, we focus on one particular application of flooding: clock synchronization. We choose this application as it is necessary for TDMA-based MAC protocols and also for applications that require signal-processing, which in-turn requires chronology of events. Our major contributions in this paper are:

- We propose TREEFP, a reliable and energy efficient manner in which flooding can be done. In TREEFP, the sleep and wake-up cycles of nodes are computed carefully. This ensures that the control message reaches the entire network by the end of the control phase with minimum energy consumption.
- We propose a novel method of forming the flooding tree as per the requirement of the network in terms of flooding delay, reliability and energy consumption. This is achieved by setting an appropriate value of *level width coefficient* for the network.
- We define a *reliability metric* which can be used to quantify reliability of a given network.
- We propose a heuristic for minimizing the TDMA frame size so that flooding finishes in one frame time, thus providing minimum and bounded flooding delay.

II. Related Work

In a classical flooding protocol, the source node sends a message that is received by all its neighbors. Each recipient node rebroadcasts the packet after caching the source id and sequence number of the message. This process repeats till all the nodes in the network receive the message. However,

978-1-4577-0351-5/11/$26.00

when CSMA is used, a straightforward broadcast of packets is very costly in terms of time and energy. [2] studies the seriousness of this broadcast storm problem, which results in redundancy, contention and collisions. [2] and [3] propose ways of reducing the broadcast problem. However, all these are based on CSMA and hence waste a lot of energy and time in contention and collisions. INFUSE [4] is a TDMA-based MAC protocol for data dissemination, hence greatly reduces the energy and time taken for flooding. However, [4] deals with lossy channels by implicit acknowledgements, which consume energy. Packet retransmissions are made in the next TDMA slot of the node, resulting in non-deterministic flooding delay.

Time synchronization (which is typically done through flooding) for TDMA systems has been studied extensively. In [5] transmitter-side non-determinism is eliminated. [6] proposes an hierarchical level structure where the root node initiates the control phase and the control messages are relayed from higher level nodes to lower level nodes. This scheme reduces the synchronization error as compared to [5], but both [6] and [5] have some synchronization error which may lead to loss of data, and in both the schemes, the number of messages exchanged is large. Solutions such as [7] use the inherent broadcast property of the wireless medium and use a single message to synchronize several receivers. However, these schemes use CSMA MAC for transmitting control messages and hence are unreliable and energy-consuming, with considerable idle-listening time.

[1] proposes a TDMA based synchronization scheme to minimize energy expenditure. This greatly reduces the energy spent, but the authors propose a guideline for determining the size of the synchronization frame based on the assumption of strict uniform density deployment. If the synchronization frame is too large or too small, there will be wastage of time or loss of synchronizm. Also, this scheme needs several rounds for calculation of transmission and reception slots. In TREEFP, slot assignment is distributed without any message passing, once the topology information is gathered. TREEFP is also more reliable than [1], as it determines the size of the control frame exactly while reusing slots whereever possible and allows a receiver to listen to alternate senders till it receives a control message. TREEFP requires each node to know its position relative to the sink. This can be achieved by programming the nodes with their location at the time of deployment or by using any good localization algorithm.

III. The TREE Flooding Protocol

When the network has to be flooded with a control message, the sink originates a control message which may contain synchronization information, code to be distributed etc. The control message has to be received by all the nodes in the network. This phase is called the *flooding phase*. The actual data transfer takes place using any MAC. The nodes are said to be in *data phase* when they are communicating data to the sink. Nodes alternate between the flooding and data phases. In this section, we propose a TDMA-based flooding scheme called TREEFP, designed with the assumptions that the nodes and the sink are stationary and each node knows the network topology. The network topology can be collected during a short beacon exchange phase after deployment as in [8], or can be programmed into the nodes at the time of deployment.

TDMA-based network flooding is done in two steps:

- A spanning tree of the network is formed with the sink as the root, to facilitate multihop communication from the sink to other nodes.
- Each non-leaf node is assigned a slot in which it can relay the control message sent by the sink, without interfering with other transmissions in the network. These slots are assigned in such a way that the length of control frame is minimized and all nodes of the network receive the control messages by the end of the control frame. This, in turn, guarantees a bounded flooding delay (delay is equal to the control frame length).

A. Formation of the Spanning Tree

In TREEFP, the nodes are logically organised so that flooding is achieved with maximum possible reliability and minimum energy and flooding delay. TREEFP uses a novel method of building the spanning tree. The levels of the spanning tree are based on the radial distance of the nodes from the sink. A network shown in Figure 1 is organised as a tree with the sink at the root. The level closest to the sink is the first level and is denoted as C_1. Based on the position of the node and the sink, the radial distance of the node from the sink is calculated by each node. Based on this distance and α, each node calculates the level C_i to which it belongs. We refer to α as *level width coefficient*, such that $0 < \alpha < 1.0$. The maximum distance between two successive levels is αR. If the radial distance of a node from the sink is between $(i-1)\alpha R$ and $i\alpha R$ $(i > 0)$, the node determines itself to be in level C_i. At the beginning of every control frame, a control message originates at the sink and flows radially outward. The message traverses from inner level to outer level one hop at a time and reaches all the nodes in the network. [8] studies how the parameter α determines the minimum node density in the area for the network to remain connected. As per this study, when the network is deployed with uniform node density, α has to be less than a threshold value for the network to remain connected. There are $H+1$ number of levels in the sensing area (including the sink, which is at level 0), where for a network of radius $RADIUS$, H is given by

$$H = \lceil RADIUS/\alpha R \rceil$$

The level farthest from the sink is identified as C_H. The outermost level C_H may not have the full level width of αR, as the network radius may not be an exact multiple of αR. Let the width of the i^{th} level be w_i. Hence,

$$w_i = \begin{cases} \alpha R & \text{if } 0 < i < H \\ RADIUS - (H-1)\alpha R & \text{if } i = H \end{cases} \quad (1)$$

Once the level to which a node belongs is decided, parent assignment is done starting from the outer-most level. Each node (say A) at level C_i is assigned a node from level C_{i-1} which is within the reception radius of A as A's parent.

An alternate way of forming the spanning tree is to use a known algorithm like the Kruskal's algorithm to give the minimum spanning tree. For this purpose, each edge of the network connectivity graph is given a weight of 1. The spanning tree thus formed has the minimum number of edges and the control frame has the minimum possible length. In TREEFP, α has a value less than 1 and nodes between $(i-1)\alpha R$ and $i\alpha R$ belong to level i. The level width coefficient α increases the number of levels and the number of non-leaf nodes in the routing tree, thus increasing the number of nodes that transmit the control message as compared to the minimum spanning tree. This results in a larger control frame than that of the minimum spanning tree, but also gives increased reliability of TREEFP over the minimum spanning tree.

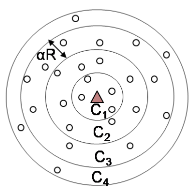

Figure 1. A Network of Sensor Nodes

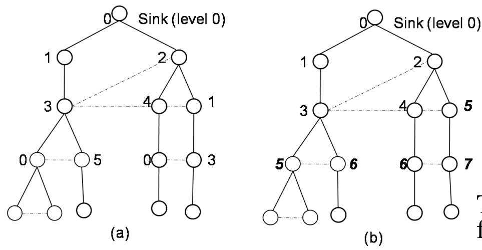

Figure 2. Time Slot Assignment in TREEFP

B. Time Slot Assignment

After the construction of the spanning tree, time slots have to be assigned to non-leaf nodes. The generation of a schedule with minimum slots can be viewed as the distance-2 graph coloring problem, which is NP-complete. Several heuristics provide schedules that are close to the ideal schedule. A heuristic with minimum colors results in maximum bandwidth utilisation, since this facilitates maximum concurrent transmissions because of maximum slot reuse. In this case, the control frame has as many slots as the number of colors. A node can only transmit a control message after it receives it from its parent. Thus, it is possible that node has to let its transmission slot go idle because it has not received the message from its parent. It would transmit the message in its slot in the next control frame, which may result in high flooding delay.

[9] proposes that a large time frame with no slot reuse is better than a short frame with maximum slot reuse, since the message can be flooded in the network in the duration of one control frame. The authors give a heuristic that orders the slots in such a way that the slots are ordered along the message path and the worst case end-to-end delay is minimised to a single frame length. The formation of the spanning tree is done using a standard algorithm like the Dijkstra's algorithm. In TREEFP, after the spanning tree is formed, it is scanned in breadth-first order in two passes for slot allocation. In the first pass, slots are allocated such that flooding is completed in one control frame. Then in the second pass, we try to reuse slots to reduce the control frame size. Slot allocation happens as per the steps given below.

- We start with an initially empty set of colors and then assign color 0 to the sink.
- Starting from the root, a breadth-first search is performed. As each non-leaf node is visited in the breadth-first search, it is assigned the lowest value in the color set that is unique from its 1 or 2 hop neighbors.
- When an existing color cannot be assigned to a node, a new color is added to the set. Figure 2(a) illustrates an example color assignment. The dotted lines together with the solid lines give the network topology graph.
- In the next pass, the tree is again scanned in the breadth first order. If the color of a child is less than that of its parent, the child is assigned the lowest color (from the current color set) that is more than its parent's color and is not used by any of the child's one hop and two hop neighbors. If no such color exists in the current set of colors, a new color is added to the set and this is assigned to the child. This results in colors being repeated wherever possible, while ensuring that a child always transmits after its parent. This enables flooding to finish in a single TDMA frame. The final color assignment is shown in Figure 2(b).

The above method gives rise to some desirable properties for flooding the network. Firstly, the flooding delay is low and deterministic (equal to the duration of one TDMA frame length). Secondly, the energy consumption during the flooding interval is also deterministic. In addition, when node density is uniform, reliability of the network can be computed. The flooding delay and flooding energy consumption are given by

$$\begin{aligned} D_{flood} &= T \cdot \rho \\ E_{flood} &= \{[T \cdot P_{tx} + (T+L-1) \cdot P_{rx}] \cdot (\rho - 2 \cdot t_{tr}) + \\ &\quad [T \cdot P_{tr} + (T+L-1) \cdot P_{tr}] \cdot 2 \cdot t_{tr}\}/(T+L-1) \end{aligned}$$

E_{flood} is the average energy spent per node in a flooding phase. The duration of each time slot is ρ seconds and in this time a node's radio has to transition twice (to switch on and off) and also transmit or receive data. The first term in the first square bracket captures the fact that only non-leaf nodes transmit the control message and the number of non-leaf nodes in the spanning tree is T. All the non-leaf nodes (except the sink) and all the leaf nodes receive the message leading to the second term. The terms in the second square bracket account for radio transitions to transmit and receive.

The difference between TREEFP and the heuristic of [9] is in the formation of the spanning tree. TREEFP forms the spanning tree as discussed in Section III-A. The tunable parameter α can be varied to form different spanning trees. As α changes, the TDMA frame length and network reliability also change. As TDMA frame length changes, flooding delay and flooding energy change. Thus, an operating value of α can be chosen to suit the need of the network in terms of reliability, flooding delay and flooding energy.

If the spanning tree is a minimum spanning tree (MST) instead of the spanning tree formed as in Section III-A, the control frame has the minimum possible length. However, loss of a control message (because of the inherent lossy nature of the wireless medium) results in all downstream nodes in the subtree of the node (which did not receive the message) not receiving the control message. TREEFP is more reliable than the MST based protocol. This is possible because TREEFP can have more non-leaf nodes (depending on the value of α) which leads to transmission of more control messages across the network. Reliability of TREEFP is discussed in Section IV.

In TREEFP, each node is assigned a slot equal to its color. A node in TREEFP can be in one of the two states: *active* or *sleeping*. A node is in active state in its allotted transmission slot to relay (transmit) a control message. A node also goes into active state at the beginning of its receive slot to receive a control message. At all other times, it is in sleeping state. Each non-leaf node is allotted a slot to transmit the control message as explained in Sections III-A and III-B. A node in level C_i can receive a control message from any non-leaf node within its reception radius, i.e., from any of its one hop neighbors. To avoid idle listening, a node A first builds an array which contains node ids of all its one-hop neighbors and then sorts this array in the non-decreasing order of their transmission slot number. We denote this array as $\Gamma(A)$ for node A. Every node in the network knows the network topology and every node runs the same spanning tree formation and time slot assignment algorithm. Hence every node knows the transmission slots of every other node in the network. Hence A can build the above said sorted array and it uses this array to receive the flooding message. If node A receives the flooding message after its transmission slot, it cannot transmit the message in the current TDMA frame. This may have an impact on the children of node A. This is why the potential senders of A are sorted in the order of their transmission slots, so that node A can receive the flooding message as early in the TDMA frame as possible.

Once the transmission and set of reception slots are calculated, non-leaf nodes change state as shown in Figure 3. Each node maintains a flag called $recvd$ which is set to 0 at the beginning of the control frame. This flag is set to 1 upon reception of a control message. Node A goes into active state only for the first slot listed in $\Gamma(A)$ to receive a control message. If it receives a control message in this slot, it can sleep till its transmission slot. Otherwise, A has to be active during its other probable reception slots, till it receives a control message. Hence, as long as there is at least one node in $\Gamma(A)$ that successfully transmits a control message, A receives a control message. Leaf nodes change state similarly for reception of control messages, but do not change state for transmission. Hence, the state transitions for leaf nodes are only those shown as dashed arcs in Figure 3.

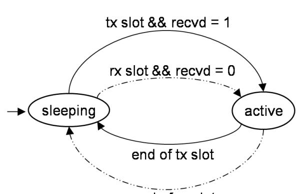

Figure 3. State Transition Diagram for a Normal Control Phase

IV. RELIABILITY IN TREEFP

In TREEFP, every non-leaf node transmits the control message once. A node that is connected to K non-leaf nodes can receive a control message even if it could not receive (K-1) messages (due to wireless error or due to faulty nodes). For a network of uniform node density (denoted by $\lambda\ nodes/m^2$), the number of non-leaf nodes within the transmission radius of a node depends on the location of the node in the network. In the discussion below, we quantify the number of failures a node can tolerate and still receive a control message in a control frame.

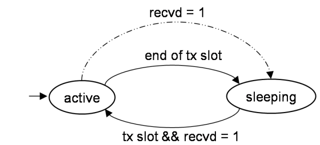

Figure 4. State Transition Diagram for the Initial Synchronization Phase

In Figure 5, we represent the levels of the routing tree with tiers which are of width αR. All the nodes within a distance of αR from the sink belong to level 1 of the tree. Similarly, all nodes belonging to level i of the tree fall in the i^{th} tier from the sink. Consider node A receiving a control message. This node can listen from any node that is in its reception range. Hence, the number of nodes from which A can receive a control message is given by K_{max}, where

$$K_{max} = \lambda \pi R^2 \tag{2}$$

Now, consider node B which is closer to the boundary of the deployment region. Since leaf nodes do not transmit control messages, the only nodes that can send a control message to this node are those in the shaded region. The shaded area (denoted by A_{conn}) is the area of overlap of two circles: a circle centered at the node B with radius R, and a tier circle centered at the sink with radius $\alpha(H-1)R$.

In Figure 5, the depicted region of overlap is the minimum area possible among all nodes in level H, since the receiver node B is at the edge of the H^{th} level. By cosine rule of triangles, the area of the shaded region is

$$A_{conn} = R^2(\psi + (H-1)^2\alpha^2\Omega - \frac{sin2\psi}{2} - \frac{((H-1)\alpha)^2 sin2\Omega}{2}) \tag{3}$$

For node B to be able to receive at least one control message, there has to be at least one node in this area of connectivity. The number of potential senders for node B is

$$K_{min} = \lfloor \lambda A_{conn} \rfloor \tag{4}$$

A_{conn} is the area of connectivity as given by Equation 3. If the value of K_{min} is greater than one, any node in C_i has more than one potential senders. Unless all the K_{min} potential senders die or their messages are lost, the receiver always receives a control message. Thus, the parameter K_{min}, which depends on α for a given network, density of deployment and transmission radius, is referred to as *reliability* of TREEFP.

Figure 8 shows how the reliability varies with α for a network with a uniform node density of 0.0025 $nodes/m^2$. The transmission radius is taken to be 100m. We considered two networks: one of 150m radius and the other of 250m radius. It is seen from Figure 8 that reliability increases with increase in network radius. As the network radius increases, the outer circle bounding C_{H-1} in Figure 5 becomes larger, because of which A_{conn} increases. Increase in A_{conn} implies higher K_{min}. For a network of given radius, reliability varies

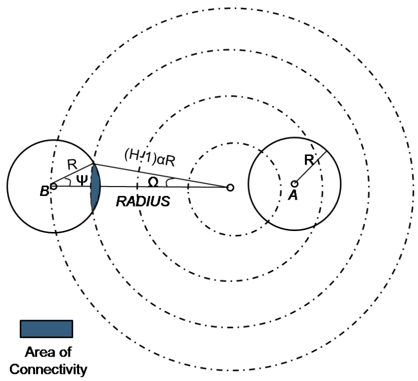

Figure 5. Reliability of TREEFP

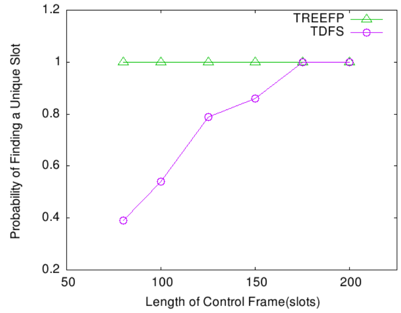

Figure 6. Probability of Every Node Being Allotted a Time Slot

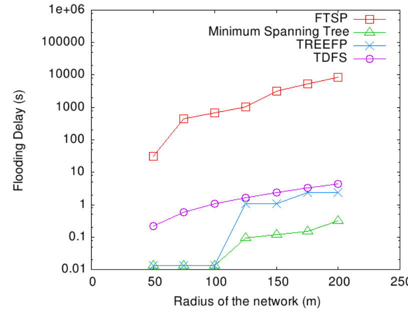

Figure 7. Time taken for Flooding Phase for Networks of Different Radii(s)

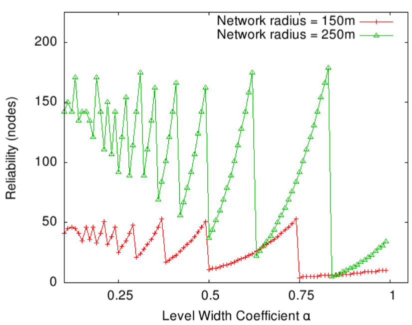

Figure 8. Variation of Reliability with α

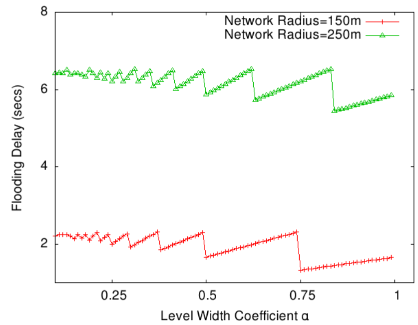

Figure 9. Variation of Flooding Time with α

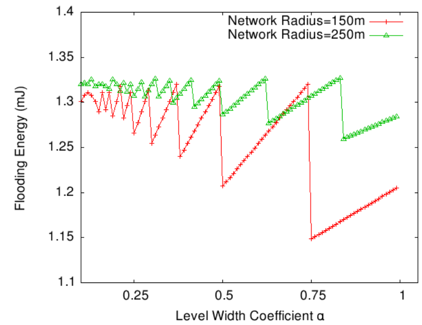

Figure 10. Variation of Flooding Energy with α

like a *sawtooth* with α. The troughs of the curve occur at those values of α where αR is an exact multiple of $RADIUS$. In these cases, the width of the last tier is αR. As a result, A_{conn} is low and so is K_{min}. As α increases, A_{conn} becomes lesser and hence, the troughs are lower. Thus, a network has least reliability when α is large and αR is an exact multiple of $RADIUS$. When α is increased slightly beyond a point where αR is an exact multiple of $RADIUS$, the number of levels decreases by 1. As α is increased further, the width of the last tier decreases. This results in a lesser number of leaf nodes which implies larger A_{conn}. This, in turn, implies larger K_{min}, i.e., increased reliability. Maximum reliability can be observed when the last tier is very thin and value of α is as small as possible to keep the network connected, since A_{conn} is large in this case. Also, note that α cannot be arbitrarily high, otherwise the network may get partitioned [8]. But high reliability (that is low α) means more non-leaf nodes in the spanning tree, which would imply larger flooding delay and energy consumption. So there is a trade off between reliability and flooding delay. Figures 9 and 10 depict the variation of flooding time and energy respectively, when α is varied. Flooding energy and flooding delay show the same trend as K_{min}. The networks considered are the same as those used for Figure 8.

V. Time Synchronization using TREEFP

A. Calculation of Clock Offset

In this section, we explain how TREEFP can be used for time synchronization. Since TREEFP is based on TDMA, it needs reliable periodic resynchronization to compensate for clock drifts at the nodes. When TREEFP is used for time synchronization, the control message originated by the base station has two values that help in synchronization. These are *globalclock* and *slot*. The *globalclock* carries the clock value of the sink at the instant the message is sent out by the sink. The *slot* field contains the slot number in which the control message is being transmitted by the transmitting node. When a node receives a synchronization message, it synchronizes itself to the global clock by setting its local clock as

$$localclock = globalclock + slot * \rho \tag{5}$$

It then copies the *globalclock* field into its outgoing message, but places the slot in which it is transmitting in the *slot* field. This message is relayed out in its transmission slot. Note that each node relays only one copy of the control message, avoiding an avalanche of messages which is typical of flooding protocols. At the beginning of the next control frame, all nodes once again switch into the sync mode (from data mode) and the synchronization is repeated.

B. Initial Synchronization Phase

Using TDMA-based flooding scheme for clock synchronization gives rise to a *chicken-and-egg* problem. The nodes need to be synchronized to be able to correctly flood the synchronization message. But until they get the synchronization message, they are not synchronized. When the network has just come up, a bootstrapping mechanism is required to synchronize the nodes for the very first time. Thus, nodes cannot follow state transition proposed in Figure 3, instead the nodes change state as per Figure 4. The state transitions of leaf nodes are shown as dashed arcs in this figure.

During this bootstrapping phase, a node stays active till it receives a control message. If the channel is not lossy, nodes which are one hop away from the sink receive the control message broadcast by the sink. On reception of the control message, these nodes set their local clocks as per Equation 5 and go to sleep. Then they wake up in their respective transmission slots to relay the control message. Nodes which are more than one hop away from the sink are still active and hence receive the control message transmitted by the first hop nodes. The same process then continues until the control message reaches all the leaf nodes. By this time, all the nodes are synchronized to the global clock. From then on, nodes synchronise periodically, changing state as per Figure 3. To cater for clock-drift that occurs in between successive synchronisation phases, guardbands are provided at the beginning and end of each time slot.

VI. Simulation Experiments

For simulation, we used NS2 and considered parameter values given in Table I. The values of the physical parameters of the nodes are taken from [10], which contains representative values for μamps sensor nodes. We consider a data rate of 19.2 Kbps and each time slot in the control frame lasts for ρ seconds. For a packet of 32 bytes, $\rho = \frac{32\times8}{19.2}ms = 13.5ms$. All experiments were repeated to achieve a confidence interval of 97%. We considered networks of radius varying from 50m to 200m, with the sink roughly at the middle of the sensing area.

In addition to TREEFP, FTSP and TDFS, we simulated a time division flooding scheme with a minimum spanning tree (MST) overlay described in Section III-B. We considered the MST scheme because this scheme requires the minimum number of slots for flooding the network. Note that [6] proposes a similar hierarchical scheme for time synchronization, but does not use time division and has two way message passing and hence consumes more energy than the spanning tree algorithm simulated by us.

Parameter	Value	Parameter	Value
$InitialNodeEnergy$	54,000J	P_{tx}	30mW
ρ	13.5 ms	P_{idle}	30mW
R	100m	P_{sleep}	0.003mW
$TransitionTime$	2.45ms	$TransitionPower$	30mW
P_{rx}	63mW		

Table I
PHYSICAL PARAMETERS USED IN SIMULATION

A. Time Slot Assignment

Time division flooding has been used in TDFS [1]. Time slot assignment in this scheme is not accurate and takes several rounds, even after which some nodes may not be assigned a transmission slot. Figure 6 shows the probability that a node finds a unique slot after a certain number of rounds, for a network of radius $150m$ with 172 nodes. If the control frame length is chosen to be less than the number of nodes, some nodes never get a slot in TDFS. When the control frame length is larger than the number of nodes in the network, all nodes are allotted a transmission slot after some rounds. Since TREEFP exactly estimates the number of slots required, every node is assigned a slot with 100% probability.

The time taken for one flooding phase for networks of different radii is plotted in Figure 7. Node deployment is taken to be of uniform density ($0.0025 nodes/m^2$). However, TREEFP can be used even for random deployments, as long as connectivity requirement is met, as discussed in [8]. α is chosen to be the value that corresponds to maximum flooding time. For example, when the network radius is $150m$, α is taken to be 0.74, as this results in maximum flooding time (Figure 9). Hence, we compare the worst case flooding time of TREEFP with the flooding time of the other protocols. Figure 7 is drawn on the logarithmic scale for clarity, as the time consumed by FTSP is much greater than that consumed by the MST scheme or TREEFP. It can be seen that FTSP takes maximum time, because of collisions. TDFS allots a unique slot for every node in the network, whereas TREEFP allows reuse of slots whereever possible and allots slots only to non-leaf nodes. It can be seen that the flooding delay is much lower for TREEFP, compared to that of TDFS. The MST takes the least time for flooding, as the slot allocation is minimal in this case. TREEFP takes considerably less time compared to FTSP, but slightly more time compared to the MST method.

B. Energy Consumption

Figure 11 shows the average energy consumed by a node during the flooding phase. FTSP is most energy consuming because of collisions and idle listening. A uniform node density of $0.0025 nodes/m^2$ is considered. For each network radius, α is chosen so that maximum flooding energy is spent for TREEFP, as described in Section VI-A. This graph is plotted on the logarithmic scale for clarity, as the energy spent by TREEFP is order of magnitude lesser than that spent by FTSP. Each node wakes up for one slot to transmit and once to receive in a control frame in the MST, TDFS and TREEFP schemes. However, the number of nodes that transmit is different in each of these schemes. Every node in TDFS transmits the control message and hence, this is the most energy consuming of the three protocols. Only non-leaf nodes transmit in the MST scheme and TREEFP. Of these two, the MST scheme uses lesser energy as it has lesser non-leaf nodes. However, TREEFP is more reliable compared to this scheme as will be discussed Section VI-C.

C. Reliability

Reliability of MST scheme and TREEFP depends on the minimum number of non-leaf nodes within a node's range, as discussed in Section IV. Figure 12 (node density = $0.0004 nodes/m^2$) and Figure 13 (node density = $0.0025 nodes/m^2$) show the percentage of total number of nodes that do not receive a control message in a flooding phase, when a percentage of the total messages relayed are lost randomly. The network considered is of radius $150m$. Loss of some relayed messages results in the control message not reaching a large number of nodes in the case

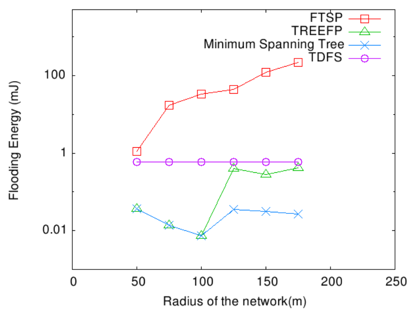

Figure 11. Flooding Energy Consumed for Networks of Different Radii(J)

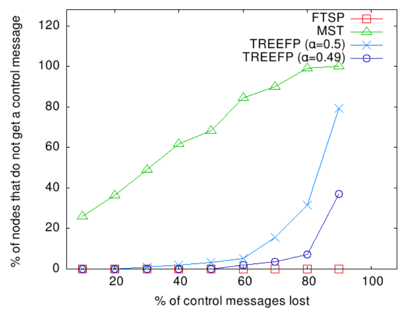

Figure 12. Effect of Loss of Control Messages on Flooding($\lambda = 0.0004 nodes/m^2$)

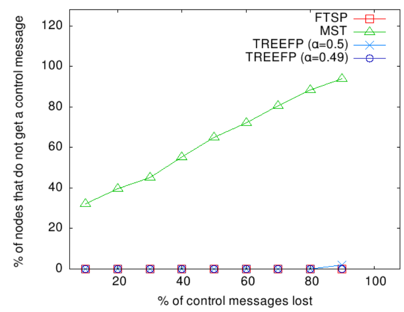

Figure 13. Effect of Loss of Control Messages on Flooding($\lambda = 0.0025 nodes/m^2$)

of the MST scheme. On the other hand, a relaying node in TREEFP has alternate senders to listen to, depending on the node density, as given by Equation 4. For each of these graphs, we chose α to be 0.5 and 0.74. These two values correspond to very low and very high reliability respectively for a network radius of $150m$, as can be seen from Figure 8. It can be seen from Figure 12 that TREEFP can flood the network reliably when up to 20% of the messages are lost when α is chosen to be 0.5. When α is chosen to be 0.49 (for higher reliability), TREEFP can result in no nodes missing the control message for upto 50% message loss.

For higher node densities (Figure 13), TREEFP results in no nodes missing the control message, even for very high message loss percentages. FTSP also has a high reliability because many messages are transmitted by each node. In TDFS, each node listens to one node within its range. Since the choice of the control frame length has a large impact on the reliability of TDFS (Section VI-A), it is not shown in Figures 12 and 13.

While TREEFP is almost as reliable as FTSP for higher node densities, it can be seen that TREEFP takes much lesser time and energy than FTSP or TDFS from Figures7 and 11. Hence, TREEFP takes lesser time for flooding and has a lower energy consumption (comparable to that of the MST scheme), while being as reliable as the more energy consuming FTSP for high node densities. This is true even for lower node densities, up to a certain percentage of message loss (Figure 12).

VII. Conclusion

We proposed a TDMA-based Reliable Energy Efficient Flooding Protocol called TREEFP for WSNs. TREEFP achieves energy efficiency by carefully determining sleep and wake-up cycles of the nodes. TREEFP has a system parameter called the *level width coefficient* which provides trade off between reliability, delay and energy consumption for flooding. This parameter can be exploited to set the operating performance of the network in terms of reliability, energy consumption and delay for flooding. TREEFP is designed such that flooding finishes in one TDMA frame duration and hence the flooding delay is bounded.

We conducted simulation experiments to compare TREEFP to other flooding protocols like FTSP, TDFS and TDMA-based MST. Results show that TREEFP is better than FTSP and TDFS and is comparable to MST, in terms of energy and flooding delay. In terms of reliability, TREEFP is always better than MST and comparable to FTSP.

References

[1] Y. Zeng, B. Hu, and H. Feng, "Time division flooding synchronization protocol for sensor networks," in *MOBIQUITOUS '07: Proceedings of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking&Services*, 2007, pp. 1–8.

[2] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99, 1999, pp. 151–162.

[3] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 15–28.

[4] S. S. Kulkarni and M. Arumugam, "Infuse: A tdma based data dissemination protocol for sensor networks." *IJDSN*, pp. 55–78, 2006.

[5] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation*, 2002, pp. 147–163.

[6] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 138–149.

[7] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 39–49.

[8] C. Shanti and A. Sahoo, "Dgram: A delay guaranteed routing and mac protocol for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1407–1423, 2010.

[9] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A global time-synchronized link protocol for sensor networks," *Ad Hoc Netw.*, vol. 6, no. 8, pp. 1201–1220, 2008.

[10] E.Shih,S.-H.Cho, N.Ickes, R.Min, A.Sinha, A.Wang and A.Chandrasekharan, "Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks." in *IEEE International conference on Mobile computing and networking (Mobicom)*, 2001, pp. 272–287.