Maximization of Lifetime and Minimization of Delay for Performance Enhancement of WSN

Priyanka M. Lokhande* and A. P. Thakare
Sipna’s College of Engineering & Technology (COET), Amravati, MS, India

Abstract
This paper, mainly concentrates on two performance metrics parameters which are important to make the WSN as effective as possible, that is maximization of lifetime parameter and minimization of delay for the for performance of wireless sensor network. Wireless Sensing Network is usually deployed in remote or hard to reach areas. In such systems energy is mostly consumed when communication radios are on. So sleep-wake scheduling is an effective mechanism to increase network lifetime. Sleep-wake scheduling is efficient in increasing network lifetime but it could result in substantial delays because a transmitting node needs to wait for its next-hop relay node to wake up. Attempts were made to reduce these delays by developing “anycast”-based packet forwarding schemes, where each node opportunistically forwards a packet to the first neighboring node that wakes up among multiple candidate nodes such a set of nodes is called forwarding node set. Anycast forwarding schemes were used to forward the data packet to next hop node which minimizes the expected packet-delivery delays from the sensor nodes to the sink node. Based on this result, a solution was provided to the problem of how to optimally control the system parameters of the sleep-wake scheduling protocol and the anycast packet forwarding protocol to maximize the network lifetime and minimize the delay with constraint on the expected end-to-end packet-delivery delay.

Keywords: Wireless Sensor Network, anycast, sleep–wake scheduling, network lifetime, delay, packet forwarding.

1. Introduction
For any network performance is the most important and main focusing aspect, it is called the performance metrics of the network including packet delivery ratio, end to end delay, network lifetime, packet loss ratio, throughput and so on. In this paper we are focusing on parameters of network life and delay and their enhancement towards the performance improvement for the WSN by using anycast and sleep wake scheduling concept. The wireless sensor network consists of large number of sensing nodes equipped with various sensing devices to observe different phenomenon changes in real world. Wireless sensor networks (WSN) are used to remotely sense the environment. Wireless sensor networks consist of many sensing nodes that captures the changes in the environment enclose data in data packets and give these packets to sink node present in the network. Such networks are present in hard to reach areas so they remain unattended for long durations. Hence the key issue in such area is efficient use of node energy to extend the lifetime of these networks. The main focus here is on event driven sensor networks for which events occur rarely. This is a very important area of research and has many applications such as, environment monitoring, intrusion detection and so on. In such systems, there are four sources of Energy consumption these sources are: communication radios, data transmission and reception, sensors

* Corresponding author: priyanka.20it@gmail.com
and transmission and reception of control packets. Fraction of total energy consumption for data transmission and reception is small for such systems because events occur so rarely. To sense the event, constant energy is required and it cannot be controlled. Hence energy required to keep communication system on means for listening the medium and control packets is a dominant component of energy consumption which can be controlled to extend network lifetime. So sleep-wake scheduling is used to increase the lifetime of event driven sensor networks. Asynchronous sleep wake scheduling is to be used, where each node wakes up independent of its neighboring nodes in order to save energy. But due to this independence of wake-up processes, additional delays are encountered at each node along the path to sink node because each node has to wait for its next hop node to wake up before transmitting the packet. Anycast packet forwarding scheme is used to reduce this event reporting delay to sink node and thus minimization of delay is achieved.

To sense the event, constant energy is required and it cannot be controlled. Sleep wake scheduling is an important method which is used to increase the network lifetime. Section I focuses on the basic idea about the wireless communication and key issues of the network lifetime as well as anycast packet forwarding scheme. Section II focuses on the different protocols used in networking and synchronized sleep wake scheduling concept, delaying minimization problem. Sections III concentrate on the problem analysis of delay minimization and lifetime minimization for the wireless sensor network. Sleep-wake scheduling with any cast is intended to solve these two main problems. Section IV is related to proposed technology with the objective to resolve the problems related to the wireless sensor network basically performance parameter of the network. Section V deals with conclusion and future work that may improve the performance of the WSN.

Earlier work in literature review has proposed the use of anycast packet-forwarding schemes (also called opportunistic forwarding schemes) to reduce this event reporting delay (Zorzi and Rao, 2003; Jain and Das, 2008; Larsson and Johansson, 2005; Rossi and Zorzi, 2007; Rossi et al., 2008).

### 2. Literature Review

Various sleep-wake scheduling protocol have been proposed in literature (Elson et al., 2002; Lu et al., 2004; Tseng et al., 2003; Van Dam and Langendoen, 2003; Ye et al., 2004). In synchronized sleep–wake scheduling protocols sensor nodes periodically or a periodically exchange synchronization information with neighboring nodes. However, such synchronization procedures could incur additional communication overheads and consume a considerable amount of energy. On-demand sleep–wake scheduling protocols (Nosovic and Todd, 2000; Shih et al., 2001) is one scheduling where nodes turn off most of their circuitry and always turn on a secondary low-powered receiver to listen to “wake-up” calls from neighboring nodes when to relay the packet. But this on-demand sleep–wake scheduling can significantly increases sensor nodes cost due to the additional receiver. This paper takes into account asynchronous sleep–wake scheduling protocols (Polastre et al., 2004; Polastre et al., 2005; Schurgers et al., 2002) in which each node wakes up independently of neighboring nodes for energy saving. However, this independence of the wake-up processes causes additional delays at each node along the path to the sink because each node needs to wait for its next-hop node to wake up before it can transmit the packet. This delay could be unacceptable for delay-sensitive applications which require the event reporting delay to be very small. So for minimizing this event reporting delay, anycast packet forwarding technique is used.

In traditional packet-forwarding schemes, every node has one designated next-hop relaying node in the neighbourhood, and it has to wait for the next-hop node to wake up when forwarding of a packet has to be done. In contrast, under anycast packet-
forwarding schemes, each node has multiple next-hop relaying nodes in a candidate set (we call it as forwarding set) and forwards the packet to the first node that wakes up in the forwarding set. It is easy to see that, compared to the basic scheme; anycast clearly reduces the expected one-hop delay. But anycast does not necessarily lead to minimum end to end delay because the packet has to relay through a time consuming path. Therefore to reduce this expected end to end delay, one challenge is how each node chooses its anycast forwarding policy, like, forwarding set this is to be solved. Another important concept here is that implementing anycast in isolation does not give good performance it has to be jointly controlled with parameters of sleep scheduling like wake up rate. Anycast addresses these challenges.

This technique shows how to implement the solution to the delay-minimization problem to construct an optimal solution to the lifetime-maximization problem for a specific definition of network lifetime. However, anycast does not necessarily lead to the minimum expected end-to-end delay because a packet can still be relayed through a time-consuming routing path. Therefore, the first challenge for minimizing the expected end-to-end delay is to determine how each node should choose its anycast forwarding policy (e.g., the forwarding set) carefully. Study of Jain and Das (2008), Larsson and Johansson (2005) and Rossi and Zorzi (2007) proposes heuristic anycast protocols that exploit the geographical distance to the sink node. The work of Rossi et al. (2008) and Liu et al. (2009) considers MAC-layer anycast protocols that work with the separate routing protocols in the network layer. However, these solutions are heuristic in nature and do not directly minimize the expected end-to-end delay. The algorithms in Choudhury and Vaidya (2004) use the hop-count information (i.e., the number of hops for each node to reach the sink) to minimize some state-dependent cost (delay) metric along the possible routing paths. However, these algorithms do not directly apply to asynchronous sleep–wake scheduling, where each node does not know the wake-up schedule of neighboring nodes when it has a packet to forward.

The second challenge stems from the fact that good performance cannot be obtained by studying the anycast forwarding policy in isolation. Rather, it should be jointly controlled with the parameters of sleep–wake scheduling (e.g., the wake-up rate of each node). It will directly impact both network lifetime and the packet-delivery delay. Hence to optimally trade off network lifetime and delay, both the wake-up rates and the anycast packet-forwarding policy should be jointly controlled. However, related work is studied in the literature (i.e., Biswas and Morris, 2005; Choudhury and Vaidya, 2004; Jain and Das, 2008; Larsson and Johansson, 2005; Liu et al., 2009; Rossi and Zorzi, 2007; Rossi et al., 2008; Zorzi and Rao, 2003) but it is not jointly controlled.

3. Analysis of Problem

Here two important problems are analysed, which affect the performance parameter of the WSN. Anycast packet forwarding should be implemented along with sleep scheduling here in order to get good performance. Two main challenges that have to be addressed in enhancing the performance of event driven wireless sensor network are:

A. Delay Minimization Problem

With the wakeup rates of the sensor nodes, optimally choosing the anycast forwarding policy to reduce the expected delay

B. Lifetime Maximization Problem

With constraint on the expected end-to-end delay, finding ways to maximize the network lifetime by controlling both the wake-up rates and the anycast packet-forwarding policy.

Sleep-wake scheduling with anycast intend to solve these two main problems. The lifetime of event driven sensor network consists of two phases: configuration phase and operation phase. In configuration phase, node optimizes control parameters of anycast forwarding policy and their wakeup rate. After configuration phase, operation phase begins in which each node alternates between two sub phases, sleeping sub phase and event reporting sub phase.

This Technique assumes that the sensor network employs asynchronous sleep–wake scheduling to improve energy efficiency, and nodes choose the next-hop node and forward the packet to the chosen node using the following basic sleep–wake scheduling protocol. This basic protocol
generalizes typical asynchronous sleep–wake scheduling protocols in order to account for anycast. In this basic protocol, we assume that there is a single source that sends out event-reporting packets to the sink. This is the most likely operating mode because when nodes wake up asynchronously and with low duty-cycles, the chance of multiple sources generating event-reporting packets simultaneously is small.

The sensor nodes sleep for most of the time and occasionally wake up for a short period of time $T_{active}$. When a node $i$ has a packet for node $j$ to relay, it will send a beacon signal and an ID signal (carrying the sender information) for time periods $t_B$ and $t_C$, respectively, and then hear the medium for time period $t_A$. If the node does not hear any acknowledgment signal from neighboring nodes, it repeats this signalling procedure. When neighboring node $j$ wakes up and senses the beacon signal, it keeps awake, waiting for the following ID signal to recognize the sender. When node $j$ wakes up in the middle of an ID signal, it keeps awake, waiting for the next ID signal. If node $j$ successfully recognizes the sender, and it is a next-hop node of node $i$, it then communicates with node to receive the packet. Node $j$ can then use a similar procedure to wake up its own next-hop node. If a node wakes up and does not sense a beacon signal or ID signal, it will then go back to sleep. Assumption is made that the time instants that node $j$ wakes up follow a Poisson random process with rate $\lambda_s$. It is also assumed that the wake-up processes of different nodes are independent. The independence assumption is suitable for the scenario in which the nodes do not synchronize their wake-up times, which is easier to implement than the schemes that require global synchronization (Elson et al., 2002; Lu et al. 2004; Van Dam and Langendoen, 2003).

Advantage of Poisson sleep–wake scheduling is that, due to its memory less property, sensor nodes are able to use a time-invariant optimal policy to maximize the network lifetime.

A well-known problem of using sleep–wake scheduling in sensor networks is the additional delay incurred in transmitting a packet from source to sink because each node along the transmission path has to wait for its next-hop node to wake up. To reduce this delay, we use an anycast forwarding scheme as described in Figure 2 (Joohwan et al., 2010) Let $\bar{C}_i$ denote the set of nodes in the transmission range of node $i$. Suppose that node $i$ has a packet, and it needs to pick up a node in its transmission range $\bar{C}_i$ to relay the packet. Each node $i$ maintains a list of nodes that node $i$ intends to use as a forwarder. We call the set of such nodes the forwarding set, which is denoted by $Fi$ for node $i$. In addition, each node $j$ is also assumed to maintain a list of nodes $i$ that use node $j$ as a forwarder (i.e $j \in Fi$). As shown in Figure 3, node $i$ starts sending a beacon signal and an ID signal successively.

![Figure 2: System Model](image-url)
All nodes in C\(\tilde{i}\) can hear these signals, regardless of whom these signals are intended for. A node j, that wakes up during the beacon signal, or the ID signal will check if it is in the forwarding set of node i. If it is, node j sends one acknowledgment after the ID signal ends. After each ID signal, node i checks whether there is any acknowledgment from the nodes in \(F_i\). If no acknowledgment is detected, node i repeats the beacon-ID-signalling and acknowledgment-detection processes until it hears one. On the other hand, if there is an acknowledgment, it may take additional time for node i to identify which node acknowledged the beacon-ID signals, especially when there are multiple nodes that wake up at the same time. Let \(T_r\) denote the resolution period, during which time node i identifies which nodes have sent acknowledgments. If there are multiple awoke nodes, node i chooses one node among them that will forward the packet. After the resolution period, the chosen node receives the packet from node I during the packet transmission period \(T_p\), and then starts the beacon-ID-signalling and acknowledgment-detection processes to find the next forwarder. Since nodes consume energy when awake, Tactive should be as small as possible. However, if is too small, a node that wakes up right after an ID signal could return to sleep before the following beacon signal. In order to avoid this case, set

\[ T_{active} = T_A + \varepsilon_{detect}, \]

where \(\varepsilon_{detect}\) is a small amount of time required for a node to detect signal in the wireless medium.

4. Proposed Work and Objectives

A technique to resolve the problem of network and maximization of lifetime and minimization of delay is going to be developed.

A. Proposed Work

The system of wireless sensor network with N nodes is considered. Each sensor node is having charge of both detecting the events and relaying the packets. If a node detects an event, the node packs the event information into a packet and delivers the packet to a sink via multihop relaying. We assume that every node must have at least one such multihop path to the sink destination. The sensor nodes employs asynchronous sleep wake scheduling for maximizing the network lifetime. And system uses any cast packet forwarding scheme to forwards the data packet to sink node through multihop relaying path. Sensor node sleeps for most of time and remains awake for some specific period of time to sense events. When node has packet to relay to next node, it sends signalling information to next node. Main focus is to decrease the incurred transmission delay and increase the network lifetime by saving the energy of individual node.

B. Objectives

Objectives of this proposed work are summarized as follows

- Analysis of delay incurred during packet transmission in wireless sensor network.
- Analysis of network lifetime for wireless sensor network.
- Proposed technique for increasing energy efficiency to maximize lifetime of network and minimization of incurred packet transmission delay to sink node
- Improve the performance of the event driven wireless sensor network.

C. Any Cast Forwarding and Sleep-Wake Scheduling Parameters

There are three control variables that affect network lifetime and end to end delay experienced by the packet

1. Wake-Up Rate:

The sleep–wake schedule is determined by the wake-up rate \(\lambda_j\) of the Poisson process with which each node j wakes up. If \(\lambda_j\) increases, the expected one-hop delay will decrease, and so will the end-to-end delay of any routing paths that pass through node j. However, a larger wake-up rate leads to higher energy consumption and reduced network lifetime. Awake probability is calculated which the function of \(\lambda_j\) is. Suppose that node i sends the first beacon signal at time 0, as in Figure 2 If no nodes in Fi have heard the first m-1 beacon and ID signals, then node i transmits the mth beacon and ID signal in the time-interval \([t_B + t_C + t_A](m-1), (t_B + t_C + t_A)(m-1)+t_B+t_C\). Therefore, provided that node i is sending the mth signal, the
probability that node \( j \in \mathcal{C}_i \) wakes up and hears this signal is

\[
p_j = 1 - e^{-\lambda_j (t_B + t_C + t_A)}.
\]

where \( p_j \) is the awake probability of node.

2. Forwarding sets:

The forwarding set \( F_i \) is the set of candidate nodes chosen to forward a packet at node. In principle, the forwarding set should contain nodes that can quickly deliver the packet to the sink. However, since the end-to-end delay depends on the forwarding set of all nodes along the possible routing paths, the optimal choices of forwarding sets of these nodes are correlated. We use a matrix \( A \) to represent the forwarding set of all nodes collectively, as follows

\[
A \triangleq [a_{ij}, i = 1, \ldots, N, j = 1, \ldots, N]
\]

where \( a_{ij} = 1 \) if \( j \) is in node \( i \)'s forwarding set, and \( a_{ij} = 0 \) otherwise. We call this matrix a forwarding matrix. Reciprocally, we define \( F_i(A) \) as the forwarding set of node \( i \) under forwarding matrix \( A \), i.e. \( F_i(A) = \{ j \in \mathcal{C}_i | a_{ij} = 1 \} \). We let \( A \) denote the set of all possible forwarding matrices.

3. Priority:

Let \( b_{ij} \) denote the priority of node \( j \) from the viewpoint of node \( i \). Then, we define the priority assignment of node \( i \) as

\[
b_i = (b_{i1}, b_{i2}, \ldots, b_{iN}),
\]

where each node \( j \in \mathcal{C}_i \) is assigned a unique number \( b_{ij} \) from \( 1, \ldots, |\mathcal{C}_i| \), and \( b_{ij} = 0 \) for nodes \( j \notin \mathcal{C}_i \). When multiple nodes send an acknowledgment after the same ID signal, the source node \( I \) will pick the highest-priority node among them as a next-hop node. Although only the nodes in forwarding set need priorities, we assign priorities to all nodes to make the priority assignment an independent control variable from forwarding matrix \( A \). Clearly, the priority assignments of nodes will also affect the expected delay. In order to represent the global priority decision, we next define a priority matrix \( B \) as follows:

\[
B \triangleq [b_{ij}, i = 1, \ldots, N, j = 1, \ldots, N].
\]

Let \( B \) denote the set of all possible priority matrices.

D. Performance Objectives of Anycast Policy and Sleep Wake Scheduling Policy

1. End-to-End Delay:

End-to-end delay is defined as the delay from the time when an event occurs to the time when the first packet due to this event is received at the sink. Performance objective is motivated as: For applications where each event only generates one packet, the above definition clearly captures the delay of reporting the event information. For those applications where each event may generate multiple packets, we argue that the event reporting delay is still dominated by the delay of the first packet. This is the case because once the first packet goes through; the sensor nodes along the path can stay awake for a while. Hence, subsequent packets do not need to incur the wake-up delay at each hop, and thus the end-to-end delay for the subsequent packets is much smaller than that of the first packet.

When there is only one source generating event-reporting packet, the end-to-end delay of the first packet can be determined as a function of the anycast policy \( (A,B) \) and the sleep–wake scheduling policy \( p \). One may argue that it may be desirable to design protocols that can potentially reduce the end-to-end delay by adjusting the anycast policy dynamically after the event occurs.

E. Network Lifetime

The second performance metric is the network lifetime, and the corresponding lifetime-maximization problem (subject to delay constraints). Let \( Q_i \) denote the energy available to node \( i \). We assume that node \( i \) consumes \( \mu_i \) units of energy each time it wakes up. So expected lifetime of node \( i \) is defined as \( Q_i / \mu_i \). Implicitly in this definition of energy consumption by data transmission is not considered. As mentioned in the introductory section, this is a reasonable approximation for event-driven sensor networks in which events occur very rarely because the energy consumption of the sensor nodes is dominated by the energy consumed during the sleep–wake scheduling.

By introducing the power consumption ratio \( e_i = \mu_i / Q_i \), The lifetime of node \( i \) can be expressed as

\[
\text{Expected Lifetime of Node } i = Q_i / (\mu_i / Q_i) = Q_i / e_i.
\]
Network lifetime is defined as the shortest lifetime of all nodes. In other words, the network lifetime for a given awake probability vector is given by

\[ T(p) = \min_{i \in N} T_i(p), \]

Based on the above performance metrics, lifetime-maximization problem (which is the second problem intended to be solved) is as follows:

\[ \max_{\bar{p}, A, B} T(p) \]

\[ D_i(\bar{p}, A, B) \leq \xi^*, \quad \forall i \in N \]

Subject to

\[ \bar{p} \in (0, 1)^N, \quad A \in A, \quad B \in B \]

Where \( \xi^* \) is the maximum allowable delay. The objective of the above problem is to choose the anycast and sleep–wake scheduling policies \( p, A, B \) that maximize the network lifetime and also guarantee that the expected delay from each node to sink \( s \) is no larger than the maximum allowable delay \( \xi^* \).

**F. End-to-End Delay Minimization**

In this section, selection of any cast policy \((A, B)\) to minimize the delay, when the awake probabilities \( p \) are given is done. The delay-minimization problem is an instance of the stochastic shortest path (SSP) problem where the sensor node that holds the packet corresponds to the “state”, and the delay corresponds to the “cost” to be minimized. The sink then corresponds to the terminal state, where the cost (delay) is not incurred anymore. Let \( i_0, i_1, i_2, \ldots \) be the sequence of nodes that successively relay the packet from the source node \( i_0 \) to sink node \( s \). Note that the sequence is random because at each hop, the first node in the forwarding set that wakes up will be chosen as a next-hop node. If the packet reaches sink \( s \) after \( K \) hops, then \( i_h = s \) for \( h \geq K \). Let \( d_i(\bar{p}, A, B) \) be the expected one-hop delay at node under the anycast policy \((A, B)\), that is, the expected delay from the time the packet reaches node \( j \) to the time it is forwarded to the next-hop node.

**Figure 3:** Selection of stopping condition

Threshold (Joohwan et al., 2010) moves from the highest-priority node \( j_9 \) to the smallest-priority node \( j_1 \) until the stopping condition is satisfied.

**5. Conclusion and Future Work**

In the event driven wireless sensor network, anycast packet-forwarding scheme is designed to reduce the event-reporting delay and to prolong the lifetime of wireless sensor networks employing asynchronous sleep–wake scheduling. Specifically, two optimization problems are focused. First, when the wake-up rates of the sensor nodes are given, we develop an efficient and distributed algorithm to minimize the expected event-reporting delay from all sensor nodes to the sink. Second, using a specific definition of the network lifetime, lifetime-maximization problem is handled to optimally control the sleep–wake scheduling policy and the anycast policy in order to maximize the network lifetime subject to an upper limit on the expected end-to-end delay. Sleep-wake scheduling with anycast substantially gives better performance than heuristic solutions in the literature under practical scenarios where there are obstructions in the coverage area of the wireless sensor network. Future work will include some additional points or strategy or method which will improve the performance of the wireless sensor network and make the network reliable and efficient.
References


