Reducing the Power Consumption of Wireless Sensor Networks with Anycast

SRIDHAR MANDA.

Abstract— In this paper, we are interested in reducing the energy consumption of event driven wireless sensor networks for which events occur infrequently. It also maximizes the lifetime and minimizes the delay of wireless sensor networks. In present systems most of energy consumed [1] when radios are on, waiting for a packet to arrive. To reduce energy consumption of sensor nodes sleep-wake scheduling algorithm is an effective mechanism .This algorithm also prolong the lifetime of these energy constrained wireless sensor networks. However, sleep-wake scheduling could result in substantial delays because a transmitting node needs to wait for its next-hop relay node to wake up. It reduces these delays (energy consumption) by implementing "anycast"-based packet forwarding schemes, where every node forwards a packet to its first neighboring node that wakes up among multiple nodes.

In this paper, we study how to optimize the anycast forwarding schemes for minimizing the expected packet –delivery delays from the sensor nodes to the sink. Based on this result, we then provide a solution to the joint control problem of how to optimally control the system parameters of sleep-wake scheduling protocol and the anycast packet forwarding protocol to minimize the energy consumption and maximize the network lifetime, subject to a constraint on the expected end-to-end packet – delivery delay. Our solution can outperform prior heuristic solutions in the literature, especially under practical scenarios where there are obstructions.

Index Terms — Anycast, Delay, end –to- packet, Energy consumption, Network lifetime , Sensor Networks, Sleep-wake scheduling,.

1 .INTRODUCTION

Recent wireless sensor networks have a unique capability to remotely sense the environment. These systems are often in remote areas. So it is hard to operate such networks for long duration. Therefore maximizing network life time by reducing the energy consumption [6] has been a key issue in the development of wireless sensor networks. In this we focus on event driven asynchronous sensor networks with low data rates, where events occur infrequently. Different sleep-wake scheduling protocols have been proposed in the literature. Among this one protocol is Synchronized sleep-wake scheduling protocols [2] . In these protocols, sensor nodes exchange the synchronization information with neighboring nodes periodically or aperiodically. However this procedure also incurs additional overhead and consumes a considerable amount of energy.

Next protocol is On-Demand sleep-wake scheduling protocols have been proposed, where nodes are turn off most of the time and always turn on the low powered receiver to listen to "wake-up" calls from neighboring nodes when there is a need for relaying packets [7]. However this protocol also increases the cost of sensor nodes because of additional receiver. So the best protocol is Asynchronous sleep-wake scheduling protocols[2] which saves energy due to independence wake-up processes, before transmitting the packet each node must wait for the next-hop node to wake up this incurs some additional delay at each node along the path. This delay could be unacceptable for unacceptable for delay-sensitive applications, such as fire detection or a tsunami alarm, which require the event reporting delay to be small. Prior work in the literature has proposed the use of anycast packet-forwarding schemes (also called opportunistic forwarding schemes) to reduce event reporting delay. I present packet-forwarding schemes, every node has one designated next-hop relaying node in the neighborhood, and it has to wait for the next-hop node to wake up it needs to forward a packet. Under anycast packet-forwarding schemes, each node has multiple next-hop relaying nodes in a candidate set (we call this set the Forwarding set) and forwards packet to the first node that wakes up in the forward set. So it's clear that anycast

---

- Sridhar manda working as an assistant professor in christu jyoti institute of technology and science, jangaon, Warangal.

reduces the expected one-hop delay. For example , assume that there are **k** nodes in the forwarding set, and each node wakes up independently according to Poisson process with the same rate, then anycast can result in a k-fold reduction in the expected one-hop delay.

However a packet can still be relayed through a time-consuming routing path in anycast so it does not necessarily lead to the minimum expected end-to-delay. So the first challenge for minimizing the expected end –to-end delay[3] is to determine how each node choose its anycast forwarding policy (e.g. the forwarding set)carefully. The heuristic any cast protocols that exploit the geographical distance to the sink node. The MAC-layer [3] anycast protocols that work with the separate routing protocols in the network layer. However, these solutions do not directly minimize the expected end-to-end delay. The algorithm which is given in this 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 is 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[2] (e.g., the wake-up rate of each node).In the later work both the wake-up rates and the anycast packet-forwarding policy should be jointly controlled.

We we address these challenges. We first investigate the *delay-minimization problem [4]* : given the wake-up rates of the sensor nodes, how to optimally choose the anycast forwarding policy to minimize the expected end-to-end delay from all sensor nodes to the sink. low-complexity and distributed solution to this problem is developed. We then formulate the *lifetime maximization problem [6]* : given a constraint on the expected end-to-end delay, how to maximize the network lifetime by jointly controlling the wake-up rates and the anycast packet-forwarding policy. We show how to use the solution to the delay-minimization problem to construct an optimal solution to the lifetime-maximization problem for a specific definition of network lifetime.

II. SYSTEM MODEL

Let consider **N** nodes in a wireless sensor network . Each sensor node will detect events and relay packets. If a node detects an event it packs that event information into a packet and delivers to sink **S** via multihop relaying. Let assume that every node has at least one multihop path to the sink and there is a single sink. We assume 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[2] . In This protocol we assume that there is a single source that sends out event-reporting packets to the sink. When nodes wake up asynchronously and with low duty-cycles, the chance of multiple sources generating event-reporting packets simultaneously is small. Furthermore, this basic protocol ignores the detailed effects of collision. The sensor nodes sleep for most of the time and occasionally wake up for a short period of time $\mathbf{t}_{\mathbf{active}}$ When a node has a packet for node to relay, it will send a beacon signal and an ID signal[7] (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 signaling procedure. When a 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, it then communicates with node *i* to receive the packet. Node 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. In this assume that the time instants that a node wakes up follow a Poisson random process with rate also assume 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.

The 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. Here it focused on when the wake-up times follow a Poisson process, also extended with non-Poisson wake-up processes, with more technically involved analysis.

The additional delay incurred in transmitting a packet from source to sink when we use sleep –wake scheduling in sensor networks 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. Let $\mathbf{c}_i$ denote the set of nodes in the transmission range of node . Suppose that node has a packet, and it needs to pick up a node in its transmission range $\mathbf{c}_i$ to relay the packet. Each node maintains a list of nodes that node intends to use as a forwarder. We call the set of such nodes the *forwarding set*, which is denoted by F_i for node . In addition, each node is also assumed to maintain a list of nodes that use node as a forwarder (i.e.,). As shown in Fig. 1, node starts sending a beacon signal and an ID signal successively. All nodes in can hear these signals, regardless of whom these signals are intended for. A node that wakes up during the beacon signal or the ID signal will check if it is in the forwarding set of node . If it is, node sends one acknowledge-mnet after the ID signal ends. After each ID signal, node checks whether there is any acknowledgment from the nodes in . If no acknowledgment is detected, node repeats the beacon-ID-signaling and acknowledgment-detection processes until it hears one. On the other hand, if there is an acknowledgment, it may take additional time for node to identify which node acknowledges the beacon-ID signals, especially when there are multiple nodes that wake up at the same time.

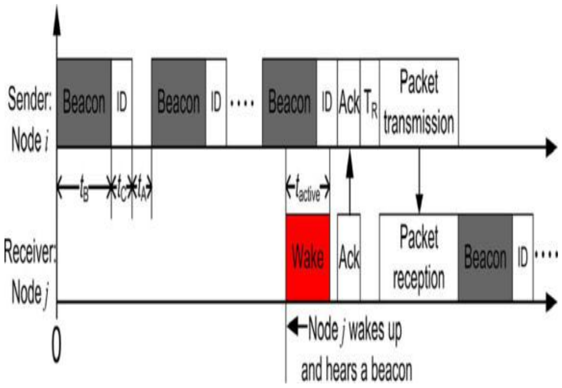


Fig. 1. System model.

Let t_R denote the resolution period, during which time node identifies which nodes have sent acknowledgments. If there are multiple awake nodes, node chooses one node among them that will forward the packet. After the resolution period, the chosen node receives the packet from node during the packet transmission period t_P , and then starts the beacon-ID-signaling and acknowledgment-detection processes to find the next forwarder. Since nodes consume energy when awake, t_{active} should be as small as possible. However, if t_{active} 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, we set , $\mathbf{t_{active} = t_A + \in_{detect}}$ where $\in_{detect}$ a small amount of time required for a node to detect signal in the wireless medium. In the rest of the paper, we assume that $\in_{detect}$ is negligible compared to $\mathbf{t_A}$.

Algorithms:

Here the value iteration algorithm used to find out the delay from source node to sink node. In this the source node delay set to infinite because we don't know when an event occurs and the sink node value set to 0 because when an event occurred it receives that without any delay.

Value-Iteration Algorithm

```
1) Initially every node i sets $D_i^{(0)} = \infty$
   Where $D_i$ is the Delay of every node.And $D_s^{(0)} = 0$ where $D_s$ is sink.
2) While(k<=N)
   $D_i^{(h)} = \min_{\mathcal{F}_{i,\vec{b_i}}} f\left(\vec{\pi}_i^{(h-1)}, \mathcal{F}_i, \vec{b}_i\right)$   (10)
   Where $\vec{\pi}_i^{(h-1)} = (D_j^{(h-1)}, j \epsilon \mathcal{C}_i)$. Let $(\mathcal{F}_i^{(h)}, \vec{b}_i^{(h)})$ be the corresponding solution the notation$D_i^{(h)}$.
3) End.
```

This algorithm used to give the priorities to the neighboring nodes from source node to the sink node.

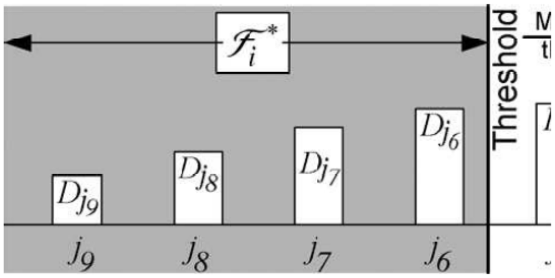


Fig2: The LOCAL-OPT algorithm which moves from highest priority node j_9 to the smallest node j_1 until the stop condition is satisfied.

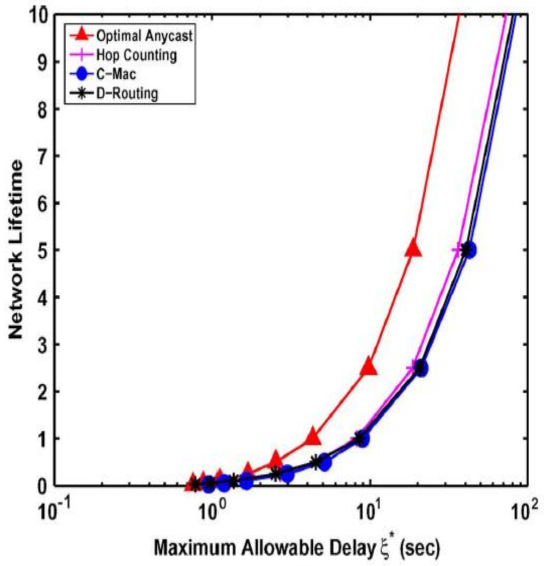
LOCAL – OPT ALGORITHM

```
1) From neighboring nodes Node $i$ receives the delay values $\vec{\pi}_i$ .
2) Assigns the optimal priorities $\vec{b}_i^*$ and let $j_k$ be the index of neighboring node with priority $k$.
3) Prod=1 and sum=0
4) For $k=|\mathcal{C}_i|$ to 1 do
5) Sum = sum+$D_{jk} \cdot p_{jk} \cdot prod$
6) Prod=prod $\cdot$ (1-$p_{jk}$)
7) Compute $f = t_D + (\frac{t_I + sum}{1-prod})$
8) If k>1 and $D_{jk-1} \geq f - t_D$ then
9) Break
10)End if
11)end for
12)$\mathcal{F}_i^* = \{j_k, j_{k+1,} \ldots, j_{|c_i|}\}$
13) return
```

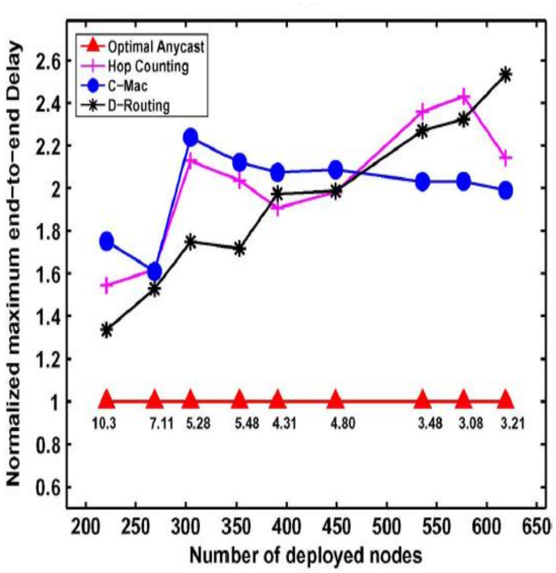
This algorithm used to increase the life time of wireless Sensor network by taking the initial time delay as 0. So depends upon the delay of each neighboring node we will calculate the lifetime of each node.

Binary Search Algorithm:

```
1: Initial Setup: The sink sets $T^{(1)}$ to half of the maximum possible lifetime
        Sets $T_{record} = 0$.
2: for m=1 to $m_{max}$ do
3: Here every node (i) computes $p_i^{(m)} = 1 - e^{-(t_I/(T^{(m)} e_i))}$ and
4: value iteration algorithm runs for $(p_j^{(m)}, j \epsilon c_i)$
5: Nodes $i$ hat satisfy $D_i^* > D_j^*$ for all neighboring nodes $j$ send feedback of their delay values $D_i^*$ to the sink.
6:Then the sink sets $D_{max} = max_i D_i^*$ and
7: if $D_{max} > \xi^*$ then
8: $T^{(m+1)} = T^{(m)} - (\frac{T^{(1)}}{2^m})$.
9:else $T^{(m+1)} = T^{(m)} - (\frac{T^{(1)}}{2^m})$ and $T_{record} = T^{(m)}$.
10. end if ; end for ;
11: return $T^* = T_{record}$.
```



(a)



(b)

VI. CONCLUSION

Anycast packet forwarding[3] scheme is used to reduce the event reporting delay and the asynchronous sleep wake scheduling algorithm reduce the energy consumption of each node and which is also prolong the lifetime of wireless sensor networks. Here it is concentrated on two aspects one is if the wake-up rates of the sensor nodes are given we developed an efficient and distributed algorithm to minimize the expected event- reporting delay from all sensor nodes to the sink. And the other one is we interested in life time maximization problem 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.

The numerical results shows that the proposed solution can outperform prior heuristic solutions .For feature work we plan to generalize our solution to take into account non-Poisson wake-up processes and other lifetime definitions.

References:

[1] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, “Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks,” *Comput. Netw.*, vol. 43, pp. 317–337, Oct. 2003.

[2] W. Ye, H. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[3] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *Proc. SenSys*, Nov. 2003, pp. 171–180.

[4] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks,” in *Proc. IPDPS*, Apr. 2004, pp. 224–231.

[5] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[6] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A.Wang, and A. Chandrakasan, “Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks,” in *Proc. MobiCom*, 2001, pp. 272–287.

[7] M. Nosovic and T. Todd, “Low power rendezvous and RFID wakeup for embedded wireless networks,” presented at the IEEE Comput. Commun. Workshop, 2000.

[8] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy-latency-density design space,” *IEEE Trans. Mobile Comput.*, vol. 1, no. 1, pp. 70–80, Jan.–Mar. 2002.

.