ISSN:0975-9646

Exalt Network Lifetime for Wireless Sensor Network with Purn Delay

Jangala Vineela, A.Srinivas Reddy
QIS College of Engineering and Technology, Ongole (dt)

Abstract: Wireless sensor networks are originally designed as distributed event-based systems that differ from traditional communication networks in several ways. These networks typically have nodes with severe energy constraints, variable quality links, low data-rate and many-to-one event-to-sink flows. In distributed event-based systems, generally events occur infrequently. Some sensitive applications such as volcanic monitoring, fire detection data should be transmitted within a specified delay to the base station. In such systems, most of the energy is consumed when the radios are on, waiting for an arrival to occur. So, sleep-wake scheduling is implemented which is an effective mechanism to 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. In this paper, we study the joint control problem of how to optimally control the sleep-wake schedule, the anycast candidate set of next-hop neighbors, and anycast priorities, to maximize the network lifetime subject to a constraint on the expected end-to-end delay. We provide an efficient solution to this joint control problem and analyze the end-to-end delay under anycast. We develop an optimal distributed anycast algorithm that minimizes the end-to-end delay of all nodes and solve the lifetime-maximization problem and it can be easily applied to energy-constrained event-driven wireless sensor networks.

I INTRODUCTION

Sleep-wake scheduling is an effective mechanism to prolong the lifetime of energy-constrained sensor networks. In this paper, we are interested in event-driven wireless sensor networks, where events occur occasionally. Therefore, by putting nodes to sleep when there are no events, the energy consumption of the sensor nodes can be significantly reduced.

In synchronized sleep-wake scheduling protocols, sensor nodes periodically or aperiodically exchange synchronization information with neighboring nodes. However, these synchronous protocols could incur additional communication overhead, and consume a considerable amount of energy. Here, we are interested in asynchronous sleep-wake scheduling protocols. In asynchronous sleep-wake scheduling protocols, the sleep-wake schedule at each node is independent of that of other nodes, and thus no synchronization is required. However, due to the lack of knowledge of the sleep-wake schedule of other nodes, it incurs additional delays for packet delivery when each node needs to wait for its next-hop node to wake up. But this delay could be unacceptable for some kind of applications like fire alarm etc.

Previous work has proposed the use of *anycast* to reduce event reporting delay. In anycast each sending node tries to wake up a group of neighboring nodes in a candidate set, and the sending node then picks the first node that wakes up to relay packets. Therefore, the delay to wake up the next-hop neighbors can be significantly reduced. On the other hand, the end-to-end delay not only depends on the per-hop delay, but also the end-to-end path that packet traverses. Hence, the set of candidate nodes must be carefully chosen because it will also affect the possible routing paths.

In this paper, we directly optimize the system with respect to the end-to-end delay. In particular, we formulate the joint control problem of how to optimally control the sleep-wake schedule, the anycast candidate set of neighboring nodes, and anycast priorities among neighboring nodes, to maximize the network lifetime subject to a constraint on the end-to-end delay. We provide an efficient and appropriate solution to this joint control problem and show how to optimally choose the candidate set in order to minimize the end-to-end delay for all nodes.

II BASIC MODEL

We consider a wireless sensor network with N nodes. Each sensor node is in charge of both detecting events and relaying packets. If a node detects an event, the node packs the event information into a packet, and delivers the packet to a sink s via multihop relaying.

With sleep-wake scheduling, 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 followed by an ID signal (carrying sender information). Here, mainly four conditions will a raise. They are:

- When node j wakes up and senses a 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 the next-hop node of node i, it then communicates with node i to receive the packet.
- If a node wakes up and does not sense any beacon signal or any ID signal, it will then go back to sleep.

Let Ci 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 Ci 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 a *forwarding set*, which is denoted by Fi.

A. Sleep-wake Schedule

The sleep-wake schedule is determined by the rate λj of the Poisson process with which each node j wakes up. If λj increases, the expected one-hop delay will decrease, and so will the end-to-end delay of any routing paths that pass though node j. However, it leads to higher energy consumption at node j so that the network lifetime may decrease.

B. Forwarding Set

The forwarding set Fi is the set of candidate nodes chosen to forward a packet at node i. 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, choosing the correct forwarding set is not easy.

C. Priority

When multiple nodes send an acknowledgement after the same ID signal, the source node i needs to pick one of them as a forwarder. We assume that node i assigns priorities to all nodes in Ci, and will pick the node with the highest priority among these nodes that wake up. Clearly, the priority assignment will also affect the expected delay.

III METRICS

The performance metrics that we are interested in:

End-to-End Delay: We assume that the end-to-end delay for event delivery is dominated by the cumulative sum of the delay for each hop to wake up and to relay a packet to its next-hop neighbor. We define the end-to-end delay as the delay incurred by the first packet, which is the sum of the delay for each hop to wake up and to relay the packet to its next-hop neighbor.

Network Lifetime: We assume that the *network lifetime* is determined by the shortest lifetime of all nodes. In other words, the network lifetime for a given awake probability vector p.

The objective of this paper is to choose awake probability vector p, forwarding matrix $\mathbf{A}$, and priority matrix $\mathbf{B}$ to maximize the network lifetime, subject to the constraint that the expected delay from each node to sink s.

IV. MINIMIZATION OF END-TO-END DELAYS

In this section, we consider how each node should choose its forwarding set and assign priorities to neighboring nodes to minimize the delay $Di(p,\mathbf{A},\mathbf{B})$. We first derive a recursive relationship for the delay, $Di(p,\mathbf{A},\mathbf{B})$ where awake probability vector p, forwarding matrix **A**, and priority matrix **B.**

At each iteration, each node uses the delay estimates from the previous iteration to update the forwarding set and the priority assignment. We will show that the algorithm converges in N iterations, and the resulting **A** and **B** minimize the expected delay $Di(p,\mathbf{A},\mathbf{B})$. Steps for OPT-DELAY algorithm are:

Step (1) At iteration 0, each node i sets: if $i = s$ *then* $D_i^{(0)} = 0$, otherwise $D_i^{(0)} = \infty$, and $F_i^{(0)} = \theta$. Each node arbitrarily assigns priorities to neighboring nodes.

Step (2) At iteration h (≥ 1), each node i sets $b_i^{(h)} = b_{i*}(\pi_i^{(h-1)})$, where $\pi_i^{(h-1)} = (D_j^{(h-1)}, j \in Ci)$.

Step (3) Each node i updates $F_i^{(h)}$ by finding the optimal forwarding set for $\pi_i^{(h-1)}$ and also updates $D_i^{(h)}$ as follows $D_i^{(h)} = f'(\pi_i^{(h-1)}, F_i^{(h)})$

Step (4) If $D_i^{(h)} = D_i^{(h-1)}$ for all nodes $i \in N$, this algorithm terminates. Otherwise, each node increases h by one and goes back to **Step (2)**.

Where, We call the function $f()$ is *the local delay function*, b_i^* *is* optimal priority assignment and πi is neighboring delay vector. Here, in order to minimize $f()$, the optimal priority assignment b_i^* can be completely determined by the neighboring delay vector πi.

V LIFETIME-MAXIMIZATION

In the previous section, we solved the delay-minimization problem. In this section, we use the result to develop a solution to the lifetime-maximization problem. We develop an efficient binary search algorithm for computing the optimal value.

Step (1) Initially, sink s sets $p^{(1)} = 0.5$ and $k = 1$.

Step (2) Sink s sets $q^{(k)} = \ln(1 - p^{(k)})^{-\max \epsilon N ei}$

Step (3) Nodes run the OPT-DELAY algorithm for given

$$p^{(k)} = (p_i^{(k)} = 1 - e^{-q^{(k)}/e_i}, i \in \mathcal{N})$$

Step (4) After N iterations, the optimal forwarding set and the optimal priority assignment under $p^{(k)}$ are found. Nodes j that are not in the other node's forwarding set, i.e., $j \notin F^* i$ ($\mathbf{A}\epsilon(p(k))$) for all nodes i, send feedback of their delays $Dj(p(k),\mathbf{A}^*(p(k)),\mathbf{B}^*(p(k)))$ to sink s.

Step (5) Let Dmax be the maximum feedback delay arrived at sink s.

- If $D\max > \xi^* + \epsilon$, then sink s sets $p(k+1) = p(k) + 0.5k+1$, increases k by one, and goes back to **Step (2)**.
- If $D\max < \xi^* - \epsilon$, then sink s sets $p(k+1) = p(k) - 0.5k+1$, increases k by one, and goes back to **Step (2)**.
- If $D\max \in [\xi^* - \epsilon, \xi^* + \epsilon]$, then the algorithm terminates, and returns $q(k)$ as the optimal solution.

The reason that we take $q(k)$ with respect to the maximum ei in **Step (2)** is because this makes all $p(k)$ i less than or equal to $p(k)$. (Note that we only search $p(k)$ over (0, 1].) In**Step (4)**, only such a node j that does not belong to any other forwarding set needs to send the feedback delay to the sink s because the node with the maximum delay does not belong to any other forwarding set. Since sink s only needs to know the maximum delay, there is no need for the other nodes to feedback their delays.

CONCLUSION

In this paper, we study how to use anycast to reduce the end-to-end delay and to prolong the lifetime of wireless sensor networks employing asynchronous sleep-wake scheduling. In particular, we study the joint control problem of how to optimally control the sleep-wake schedule, the anycast candidate set of next-hop neighbors, and the anycast priorities, in order to maximize the network lifetime subject to a upper limit on the expected end-to-end delay. We provide an efficient solution to this joint control problem, and as a part of the solution, we also show how to optimally choose the anycast candidate set to minimize the end-to-end delay from all sensor nodes. The algorithms that we have developed can be easily applied to energy-constrained event-driven wireless sensor networks.

REFERENCES

[1] G. Lu, B. Krishnamachari, and C. S. Raghavendra, .An Adaptive Energy-Ef_cient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks,. in *Proc. IPDPS*, pp. 224.231, April 2004.

[2] 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.

[3] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, “On Maximizing the Lifetime of Delay-Sensitive Wireless Sensor Networks with Anycast,” *Technical Report, Purdue University, http://web.ics.purdue.edu/˜kim309/Kim08tech.pdf*, 2007.

[4] M. Zorzi and R. R. Rao, .Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance,. *IEEE transactions on Mobile Computing*, vol. 2, pp. 349.365, October 2003.

[5] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, .Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks with Anycast,. *Technical Report, Purdue University, http://web.ics.purdue.edu/.kim309/Kim08tech2.pdf*, 2008.

[6] in wireless ad-hoc networks,. in *37th Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL), October 1999.

[7] J.-H. Chang and L. Tassiulas, .Energy conserving routing in wireless ad-hoc networks,. in *Proceedings of IEEE INFOCOM*, pp. 22.31.

[8] Y. T. Hou, Y. Shi, and H. D. Sherali, .Rate allocation in wireless sensor networks with network lifetime requirement,. in *Proceedings of IEEE/ACM MobiHoc*, pp. 67 . 77, 2004.