A LIFETIME-AWARE SINGLE-SESSION FLOW ROUTING ALGORITHM FOR ENERGY-CONSTRAINED WIRELESS SENSOR NETWORKS

Y. Thomas Hou Yi Shi
Virginia Tech, The Bradley Dept. of Electrical and Computer Engineering
Blacksburg, VA, USA

Jianping Pan
University of Waterloo
Waterloo, Ontario, Canada

***Abstract*— Wireless sensor networks are becoming increasingly important in recent years due to their ability to detect and convey real-time in-situ scenes for many civil and military applications. A major technical challenge for a wireless sensor network lies in the energy constraint at battery-powered nodes, which poses a fundamental limit on the network lifetime. We consider two-tiered wireless sensor networks and address the network lifetime problem for upper-tier *aggregation and forwarding nodes* (AFNs). Prior efforts have formulated the network lifetime problem into a linear programming (LP) problem that results in a *multi-session* flow routing solution. Under multi-session flow routing, each AFN must be equipped with multiple transmitters to reach various destinations at the same time, which poses scalability problem in practice. In this paper, we present SEES (for "Smart Energy Exploitation and Sharing"), which is a *single-session* flow routing solution (requiring only a single pair of transmitter/receiver at each AFN). SEES seeks to maximize network lifetime through energy sharing among the AFNs and following the *max-min* concept. Simulation results show that the SEES algorithm, albeit employing single-session flow routing, can match closely the maximum network lifetime performance obtained by an optimal multi-session flow routing solution.**

***Index Terms*— Network lifetime, energy constraint, power control, energy sharing, flow routing, max-min, wireless sensor networks**

I. INTRODUCTION

Wireless sensor networks have recently found many new applications that could have significant impact throughout our society. In this paper, we consider two-tiered wireless sensor networks that can be deployed for various sensing applications. These networks consist of a number of sensor clusters and a base-station. Each cluster is deployed around a strategic location and consists of a number of wireless *application sensor nodes* (ASNs) and one *aggregation and forwarding node* (AFN). Each ASN is used to capture and transmit data stream to an AFN while the AFN performs in-network processing by aggregating all correlated information within the cluster (which is also known as "fusion"). The AFN then sends the composite data stream to the base-station via single or multi-hop transmission.

The most important performance measure for wireless sensor networks is *network lifetime.* For two-tiered wireless sensor networks, whenever an AFN runs out of energy, the sensed information from that local area is lost. Therefore, the definition for network lifetime would be the time from network activation to any AFN failure. Since the lifetime of each AFN heavily depends on its energy consumption behavior, and the major source of power consumption at an AFN attributes to its radio communication, it is essential to devise strategies that can minimize radio-related power consumption. One promising approach to maximizing network lifetime is to control the power level of radio transmitter. Since the power level of a radio transmitter directly affects its coverage, it is important to utilize the relaying capability among the AFNs to forward data streams.

This paper investigates the optimal network flow routing (through power control) among the upper-tier AFNs to maximize the network lifetime. In [4], Chang and Tassiulas formulated the network lifetime problem into a linear programming problem which results in a multi-session flow routing solution. Under this approach, the number of transmitters on each AFN must be $O(N)$, where N is the total number of AFNs served by one base-station. Clearly, this fact poses a scalability problem that is of great concern in practical deployment.

In this paper, we are interested in exploring flow routing solutions by limiting each AFN to be equipped with only a single pair of transmitter/receiver. Under this approach, each AFN can only transmit to a single destination at any time. This approach thus gives a so-called *single-session* flow routing solution. There are several reasons why we are interested in such a single-session solution. The single-session solution requires to fully exploit dynamic power control capability at each AFN, which is a much

This research is supported in part by ONR under grant number N00014-03-1-0521.

0-7803-8140-0/03$17.00 (C) 2003 IEEE

more challenging and interesting problem than static multi-session flow routing solutions that have been studied in prior efforts. More importantly, we believe that the single-session solution (based on dynamic power control) is a foundation to all related research efforts in optimal flow routing problems to maximize network lifetime.

We presents a *Smart Energy Exploitation and Sharing* (SEES) algorithm, which is a single-session flow routing solution. The main idea in SEES is an emulation of energy sharing based on the max-min concept: during each iteration, SEES identifies the AFN that currently has the lowest estimated lifetime and attempts to maximize its lifetime, by exploiting potential relaying nodes with a larger estimated lifetime. We show that SEES is a polynomial algorithm and matches closely the maximum network lifetime obtained in an optimal multi-session flow routing solution.

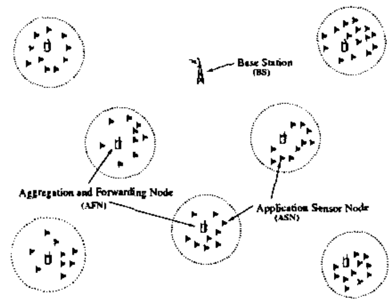
The remainder of this paper is organized as follows. In Section II, we provide a reference network model for wireless sensor networks and discuss its power consumption behavior. In Section III, we present the SEES algorithm. Performance results for SEES are given in Section IV. Section V reviews related work in wireless sensor networks and Section VI concludes this paper.

II. Network and Power Consumption Models

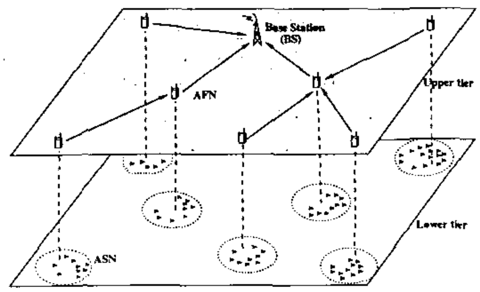
A. Network Reference Model

We focus on two-tiered architecture for wireless sensor networks.[1] Figures 1(a) and (b) show the *physical* and *hierarchical* topology for such networks, respectively. There are three types of nodes in the network: *application sensor nodes* (ASNs), *aggregation and forwarding nodes* (AFNs), and *base-station* (BS). The small and low-cost ASNs constitute the lower-tier of the network, and are densely deployed in groups (or clusters) at strategic locations for various sensing applications. The objective of an ASN is very simple: once triggered by an event (*e.g.*, detection of motion or biological/chemical agents), it starts to capture live information (video, audio, or scalar measurement), which it sends directly to the local AFN in one hop. It is worth pointing out that multi-hop routing among ASNs are not necessary due to the small distance between an ASN and its AFN. By deploying these inexpensive ASNs densely in clusters, and within proximity of a strategic location, it is possible to obtain a comprehensive view of the area by exploring the correlation among the scenes collected at each ASN. Furthermore, the reliability of area surveillance can also be improved through the redundancy among ASNs in the same cluster.

[1] The two tiered architecture is motivated by recent advances in *distributed source coding* (DSC) for sensor networks [5], [10].



(a) Physical topology.



(b) A hierarchical view.

Fig. 1. Reference model of tiered wireless sensor network.

For each cluster of ASNs, there is one AFN, which is different from an ASN in terms of both its physical properties and logical functions. The primary functions of an AFN are: 1) *data aggregation* (or "fusion") for flows coming from the local cluster of ASNs, and 2) *forwarding* (or relaying) the aggregated data streams toward the base-station. For data fusion, an AFN analyzes the content of each data stream received from ASNs and composes a complete view by exploiting the correlation among each individual scene. In addition, AFNs have an important networking function for the upper-tier architecture: it serves as the *relaying node* for other AFNs to forward data streams toward the base-station. Although an AFN is expected to be provisioned with much more energy than an ASN, it also consumes energy at a substantially higher rate (due to the wireless communication over large distances). Consequently, an AFN has a limited lifetime. Upon the depletion of energy at an AFN, we expect that the *coverage* for that particular area under surveillance is lost.

The last component within the two-tiered architecture is the base-station, which is basically the *sink* node for data streams generated at all AFNs. A base-station may be assumed to always have a sufficient battery provisioning, or its battery may be re-provisioned during its course of operation. Therefore, its power consumption is not a concern in our investigation.

In summary, the lower-tier ASNs are used for data acquisition. The upper-tier AFNs are used for data fusion and forwarding the aggregated data toward the base-station.

Although the AFNs and base-station locations are immobile, there is a great degree of flexibility in network flow routing. Power control of the transmitter at an AFN can determine radio signal's coverage, which in turn affects the network routing topology [9], [12], [15]. In this paper, we will fully explore dynamic power control capability for network flow routing to maximize network lifetime.

B. Power Consumption Model

A detailed power dissipation model for each component in a wireless sensor node can be found in [7]. For an AFN, the radio-related power consumption (*i.e.*, in transmitter and receiver) is a dominant factor [1]. The power dissipation at a transmitter can be modeled as:

$$p_t(s_i,\ s_k) = c_{s_i,s_k} r_o\ , \tag{1}$$

where $p_t(s_i,\ s_k)$ is the power dissipated at AFN s_i when it is transmitting data stream at rate r_o to AFN s_k. c_{s_i,s_k} is the power consumption cost of link (s_i, s_k) and

$$c_{s_i,s_k} = \alpha_{t1} + \alpha_{t2} d^n_{s_i,s_k}\ , \tag{2}$$

where α_{t1} is a *distance-independent* term, α_{t2} is a *distance-dependent* term, d_{s_i,s_k} is the distance between these two AFNs, n is the path loss index and $2 \le n \le 4$ [11]. Typical values for these parameters are $\alpha_{t1} = 45$ nJ/b and $\alpha_{t2} = 0.001$ pJ/b/m^4 ($n = 4$) [3]. In this paper, we use $n = 4$ in all of our numerical results.

The power dissipation at a receiver is [11]:

$$p_r = \alpha_r r_i\ , \tag{3}$$

where r_i (in b/s) is the incoming rate of received data stream. Typical value of α_r is 135 nJ/b [3].

III. SEES: A Single-session Flow Routing Algorithm

Although it is easy to formulate a linear programming problem to find a flow routing solution with optimal network lifetime [4], there are two potential concerns with this approach. First and foremost, this approach gives a multi-session solution, which requires $O(N)$ transmitters at each AFN, where N is the number of AFNs. Clearly this leads to a scalability issue which may be of important concern in practice. Second, although such approach offers an optimal network flow routing solution, it hardly provides any insight on how energy should be shared among the AFNs to achieve maximum network lifetime. These issues motivate us to pursue a single-session flow routing solution.

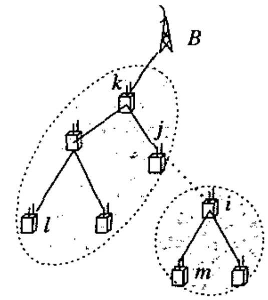


Fig. 2. A schematic diagram illustrating the SEES concept.

In this section, we develop a Smart Energy Exploitation and Sharing (SEES) algorithm to maximize network lifetime. This algorithm exploits intrinsic energy sharing behavior at each AFN and can achieve near-optimal results with very low computational complexity. At the conceptual level, SEES builds itself largely upon the concept of "max-min" [2]. Specifically, under the classical max-min theory, the objective is to achieve fairness in bandwidth sharing among flows in the network and a max-min algorithm always attempts to increase the rate of the session with the minimum rate. Similarly, for our network lifetime problem, we can identify the AFN that has the smallest *estimated* lifetime and use the max-min concept to maximize its estimated lifetime, by exploiting potential relaying nodes with a larger estimated lifetime.

Before describing how SEES works, we introduce the notion of *one-hop subtree*. Referring to Fig. 2, for AFN s_l, we call AFN s_k as the root of a one-hop subtree since AFN s_k is only one-hop toward the base-station B, and we call this subtree as s_l's one-hop subtree.

Under SEES, AFN s_i will use AFN s_j as a relaying node for a period of τ (*i.e.*, energy exploitation period at AFN s_j) if the following two criteria are met: (1) AFN s_j's estimated lifetime is longer than AFN s_i's; and (2) the estimated lifetime of all AFNs in s_j's one-hop subtree will not have an estimated lifetime shorter than the estimated lifetime of AFN s_i without the relaying. Figure 2 illustrates the lifetime-based max-min concept in SEES. AFN s_j is a potential relaying candidate for AFN s_i if criteria (1) is satisfied. AFN s_j then calculates the maximum amount of time τ that it can offer AFN s_i, using the criteria (2).

To increase the precision of SEES, we use a binary search technique to determine the network lifetime T. The initial lower bound for T can be set to the minimum of all one-hop lifetime when all AFNs transmitting directly to the base-station; the initial upper bound for T can be set to the minimum of all lifetime when all AFNs relaying to its nearest neighboring node (including the base-station). Let

```
0. T: Current target network lifetime
1. SysTime: System time
2. ELT_i: Estimated Life Time of AFN s_i.
3. ART_i: Available Relay Time AFN s_i gets from its relay.
4. DRT_i: Desired Relay Time AFN s_i wishes to have.
5. ECS_i: Energy Crisis Status of AFN s_i
6. RE_i: Remaining energy at AFN s_i
```

Fig. 3. Notations used in SEES algorithm.

```
//Initialization
    //For each AFN s_i, find its neighboring AFNs that are
    //closer to s_i than the base-station B
    for (i = 1; i <= N; i++) {
        build relaying node list R_i = {k|dist(s_i, s_k) < dist(s_i, B)}
        order R_i in increasing distance from AFN s_i,
        the last node is B; }
    SysTime = 0;
//Initialize with one-hop flow routing for each AFN
    for (i = 1; i <= N; i++)
        Compute ELT_i, DRT_i, ART_i, ECS_i for one-hop;
    SystemCheck();
//Build initial tree
    S = {i|s_i hasn't tried to use a relay};
    while (S != NULL) {
        Find s_i such that ELT_i = min(ELT_j|j ∈ S);
        UseRelay(i); }
    ImproveTree();
    SystemCheck();
//Main iteration
    while (SysTime < T) { //run to next event and rebuild tree
        Δ = min(min(ART_i), min(DRT_i|ECS_i == 0));
        SysTime = SysTime + Δ;
        update ELT_i, DRT_i, ART_i, ECS_i, RE_i;
        S = {i|s_k is the relaying node of s_i, ART_k == 0;
            or ECS_i == 0 and DRT_i == 0};
        while (S != NULL) {
            Find s_i such that ELT_i = min(ELT_j|j ∈ S);
            AFN s_i switches to use B as its relaying node;
            update ELT, DRT, ART, ECS in s_j's one-hop subtree;
            UseRelay(i); }
        ImproveTree();
        SystemCheck(); }
```

Fig. 4. Main components in SEES algorithm.

```
//Extend the lives of AFNs that will die before next event
void ImproveTree()
{
    D = {i|s_k is the relaying node of s_i, ELT_i < ART_k
        and ELT_i < T - SysTime};
    while (D != NULL) {
        Find s_i such that ELT_i = min(ELT_j|j ∈ D);
            AFN s_i switches to use B as its relaying node;
            update ELT, DRT, ART, ECS in s_i's subtree;
            UseEmergentRelay(i); }
}
//Find next relaying node
void UseRelay(int i)
{
    j=next relaying node of s_i;
    while (s_j != B) {
        if ((ELT_j > max(ELT_i, T - SysTime))
            and (CheckCircle(j, i)==0))
            if ((ECS_i == -1) and (ECS_j != -1))
                continue;
            else if (TryRelay(j, i)==1) {
                record s_i use s_j;
                Update(j, i);
                break; }
        j=next relaying node of s_i; }
}
```

Fig. 5. Auxiliary functions used in SEES algorithm.

T be the average of the lower and upper bounds, then SEES returns an answer of whether or not a network flow routing solution can exceed this T. Depending on the answer, we can narrow the range of T and let T be the average of the new lower and upper bounds and proceed to the next iteration. The iteration terminates once the range of upper and lower bounds is less than a chosen threshold.

During each iteration, we use the max-min concept to build a network flow routing solution. Figure 4 shows the key components of SEES, with notations and some auxiliary functions given in Figs. 3, 5 and 6, respectively. There are four variables associated with each AFN s_i, namely: *Estimated Lifetime* (ELT_i), *Desired Relaying Time* (DRT_i), *Available Relaying Time* (ART_i), and *Energy Crisis Status* (ECS_i). ELT is the current estimated remaining lifetime of AFN s_i. ART is the maximum available time that AFN s_i can be used as a relaying node for its current children. DRT refers to the desired time that AFN s_i wants to use current relaying node, so that following two criteria can be met: (1) AFN s_i can support itself to meet the network lifetime T, and (2) the ART at AFN s_i can also meet T. ECS is a status flag used to indicate the remaining energy level and can be -1, 0, and 1. When ECS_i is 1, it indicates a "crisis" situation where there is at least one AFN within s_i's subtree that has $ELT < T - SysTime$. When ECS_i is -1, it indicates that there does not exist an AFN in s_i's one-hop subtree that has $ELT < T - SysTime$. When ECS_i is set to 0, it indicates a mixed situation: there is no AFN in s_i's subtree has $ELT < T - SysTime$ but there is at least one AFN within s_i's one-hop subtree has $ELT < T - SysTime$ (see Fig. 2).

For each AFN s_i, we denote set R_i containing all the AFNs that are within the radius from AFN s_i to the base-station. It is worth pointing out that, for AFN s_i, only nodes in R_i may be chosen as relaying nodes. We sort nodes in R_i in an increasing order of distance from AFN s_i and include the base-station B as the last node in the set R_i.

At time 0, we examine the one-hop case to see if each AFN can meet or exceed the given T. If yes, we are done and can update the range of T and move onto the next iteration. Otherwise, we move onto the phase of building up an initial tree that offer a better lifetime.

The initial tree is built upon an iteration of going through all AFNs that have not yet attempted to use a relaying node. Among these AFNs, we start with the one that has the minimum ELT and see if we can find a relaying node for this AFN. After we build the initial tree, some AFNs may have $ELT < T - SysTime$ and will run out of energy

```
//Find next emergent available relaying node
void UseEmergentRelay(int i)
{
    j=next emergent relaying node of s_i;
    while (s_j ≠ B) {
        if ((ELT_j > T − SysTime) and (CheckCircle(j, i)==0))
            if (TryRelay(j, i, 0)==1) {
                record s_i use s_j;
                Update(j, i);
                break; }
        j=next emergent relay of s_i; }
        if ((ELT_i < T − SysTime) and (ELT_i < ART_j)) {
            printf("Lifetime < T");
            exit(0); }
}
//When check==0, if after using this relaying node,
//s_j's one-hop subtree has an AFN ∈ D
//and its relaying node's new ART <new ART_j, return 0, else return 1
//When check==1, if after using this relaying node,
//s_j's one-hop subtree has an AFN
//whose new life < s_i's old life, return 0, else return 1
int TryRelay(int j, int i, int check=1)
{
    ∀ AFNs in s_j's one-hop subtree,
        compute new ELT, DRT, ART;
    if (check==1)
        if (∃s_k in s_j's one-hop subtree, new ELT_k < ELT_i)
            return 0;
        else
            return 1;
    else //check=0
        if (∃s_k ∈ D in s_j's one-hop subtree,
            s_r is s_k's relaying node, new ART_r < new ART_j)
            return 0;
        else
            return 1;
}
//s_i use s_j as its relaying node,
//update ELT, DRT, ART, ECS in their one-hop subtree
int Update(int j, int i)
{
    ∀ AFNs in s_j's one-hop subtree, copy new ELT, DRT, ART,
        compute ECS;
}
```

Fig. 6. Additional auxiliary functions.

before their relaying node ceases to support them, which brings the network lifetime less than the current T. Denote D the set of all these AFNs. An AFN s_j node is used as an "emergent" relaying node of AFN $s_i \in D$ for a period of τ if the following two criteria are met: (1) AFN s_j's estimated lifetime is longer than AFN s_i's; and (2) all AFNs in both set D and s_j's one-hop subtree will not be supported for a period shorter than τ. We try to find the next available emergent relaying node for such AFNs. Should we find that, after using new relaying nodes, their ELT are still below $T - SysTime$, we declare failure in finding a network flow routing schedule for this T and the algorithm terminates for this iteration.

The function `SystemCheck()` is used to make a comparison of the current estimated network lifetime and the target lifetime T. If the ELT for all AFNs in the network is no less than T, then there is no need to proceed further and we can declare that we have found a flow routing solution.

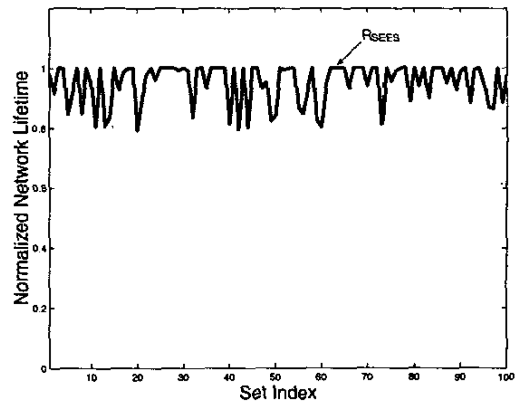


Fig. 7. Normalized network lifetime for 100 sets of experiments.

Otherwise, we move on to the main iteration. The function `CheckCircle()` is used to examine if there is any cycle in the network after using a relaying node.

In the main iteration, we use the lifetime-based max-min concept to extend the estimated lifetime at each iteration, each time with a new relaying tree. We then compare this newly estimated network lifetime with the target lifetime T to see if we need to proceed further. The program will terminate before $SysTime$ exceeds T if we limit $DRT, ART \leq T - SysTime$. Upon termination, we have two possible outcomes, either we find that the current estimated network lifetime for all AFNs is no less than $T - SysTime$ (success), or at least one AFN will deplete all of its energy before time T (failure).

IV. Performance Evaluation

We first discuss the complexity of SEES. In SEES, most computations take place when an AFN attempts to identify and uses a potential relaying node. During this operation, we need to examine a one-hop subtree ($O(n)$) and update each AFN in this one-hop subtree (if the AFN will join the subtree for relaying). Since we have a total of n AFNs and each AFN s will examine the nodes in R_s ($|R_s| \leq n$) at most twice, we will perform at most $2\sum_{i=0}^{n}|R_i| \leq 2n \cdot n = O(n^2)$ iterations of attempts of identifying potential relaying nodes. Therefore, the total computational complexity is $O(N^3)$.

In the following, we present numerical results to illustrate the performance of SEES. We randomly generate 100 sets of experiment data. The coordinate of base-station is (500, 500)(in meters). An AFN is placed randomly (following uniform distribution) at (x_i, y_i) with energy e_i, and each AFN generates data with rate F_i, where $0 \leq x_i, y_i \leq 1000$ (m), $100 \leq e_i \leq 1000$ (kJ), $1 \leq r_i \leq 10$ (kb/s). There are 20 AFNs. For each set of experiment, we can get the optimal lifetime T_{LP} through the linear programming approach.

Set	T_{SEES}	T_{LP}	Set	T_{SEES}	T_{LP}
1	45.75	46.51	11	43.97	54.35
2	58.91	64.35	12	42.66	42.66
3	19.00	19.00	13	71.40	88.41
4	31.93	31.94	14	55.98	67.04
5	79.17	93.52	15	88.20	88.20
6	60.45	65.74	16	40.23	43.40
7	31.33	31.33	17	24.78	25.18
8	65.97	77.70	18	30.43	30.43
9	23.61	23.63	19	14.44	14.44
10	34.08	35.93	20	23.41	29.55

TABLE I
NETWORK LIFETIME (DAY) PERFORMANCE FOR FIRST 20 SETS OF NUMERICAL RESULTS.

T_{SEES} is the network lifetime obtained through the SEES algorithm. For clarity, we present the normalized network lifetime for each experiment, *i.e.*, $R_{SEES} = T_{SEES}/T_{LP}$. Figure 7 shows R_{SEES} for these 100 experiments. The mean normalized network lifetime is 0.9501, variance is 0.0671, and the 95% confidence interval is [0.9394, 1]. In Table I, we list the first 20 sets of experiments.

V. RELATED WORK

Power control has been explored in the literature at different layers. Here, we briefly review the power control at *network* (routing) layer, which can be classified into two areas. The first area comprises of strategies to determine an optimal transmitter power to control the network *connectivity* (*e.g.*, [9], [12], [15]). A common theme in these strategies is to adjust each node's transmitter power so that different network connectivity topologies can be formed. In [9], Ramanathan and Rosales-Hain's objective is to keep the number of one-hop neighbors be bounded. In [12], [15], the authors aim to design distributed power control algorithms to achieve network connectivity.[2]

The second area could be called *power-aware routing*. Most schemes use the shortest path algorithm with a power-based metric, rather than a hop count-based metric (see *e.g.*, [6], [8], [13], [14]). In [13], Singh *et al.*, made some suggestions on metrics for power-aware routing, including energy consumed per-packet, time to network partition, variance in battery life of nodes, cost per packet, and node cost. However, energy-aware (*e.g.*, minimum-energy path) routing may not ensure good performance in *energy-constrained* applications. For example, using the most energy-efficient routes may still result in premature depletion of energy at certain nodes, which is not optimal in network lifetime performance.

[2] The notion of network lifetime in [15] is from connectivity perspective and is different from ours.

VI. CONCLUSIONS

In this paper, we explored the single-session flow routing solution to maximize network lifetime for wireless sensor networks. The main contribution of this paper is the development of SEES, a single-session flow routing algorithm, which exploits energy sharing by emulating the max-min concept. We show that SEES is a polynomial algorithm and matches closely the maximum network lifetime obtained under an optimal multi-session flow routing solution.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier)*, vol. 38, pp. 393-422, 2002.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Chapter 6, Prentice Hall, Englewood Cliffs, NJ, 1992.

[3] M. Bhardwaj and A.P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. IEEE Infocom*, pp. 1587–1596, 2002.

[4] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. IEEE Infocom*, pp. 22–31, Tel Aviv, Israel, March 2000.

[5] J. Chou, D. Petrovis, and K. Ramchandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in *Proc. IEEE INFOCOM 2003*, April 2003, San Francisco, CA.

[6] S. Doshi, S. Bhandare, and T.X. Brown, "An on-demand minimum energy routing protocol for a wireless ad hoc network," *ACM Mobile Computing and Communications Review*, vol. 6, no. 3, July 2002.

[7] W. Heinzelman, *Application-specific Protocol Architectures for Wireless Networks*, Ph.D. thesis, MIT, 2000.

[8] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless Ad-hoc networks," in *Proc. ACM MOBICOM 2001*, pp. 97–107, Rome, Italy.

[9] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. IEEE Infocom*, pp. 404–413, Tel Aviv, Israel, March 2000.

[10] K. Ramchandran, "Distributed sensor networks: opportunities and challenges in signal processing and communications," presentation at *NSF Workshop on Distributed Communications and Signal Processing for Sensor Networks*, Dec. 2002, Evanston, IL.

[11] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, New Jersey, 1996

[12] V. Rodoplu and T.H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, August 1999.

[13] S. Singh, M. Woo, and C.S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. ACM/IEEE Mobicom*, Dallas, TX, Oct. 1998.

[14] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, Nov. 2001.

[15] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. IEEE Infocom*, pp. 1388–1397, 2001.