# Sensor Network Lifetime Maximization Via Sensor Energy Balancing: Construction and Optimal Scheduling of Sensor Trees

Ling Shi*, Agostino Capponi†, Karl H. Johansson‡ and Richard M. Murray*

***Abstract*— In this paper we consider state estimation carried over a sensor network. A fusion center forms a local multi-hop tree of sensors and fuses the data into a state estimate. A set of sensor trees with desired properties is constructed, and those sensor trees are scheduled in such a way that the network lifetime is maximized. The sensor tree construction and scheduling algorithms are shown to have low polynomial time complexity which lead to efficient implementation in practice. The scheduling algorithm is also shown to return the optimal solution. Examples are provided to demonstrate the algorithms.**

## I. Introduction

### A. Background

Wireless sensor networks have attracted much attention in the past few years and this area of research brings together researchers from computer science, communication, control, etc [1]. A typical wireless sensor network consists of a large number of sensor nodes and some base stations [2]. Sensor nodes are usually battery powered and have limited processing capabilities. They interact with the physical world and collect information of interest, e.g, temperature, humidity, pressure, air density, etc. Depending on the Media Access Control (MAC) and routing protocols, as well as the available resources (network bandwidth, node energy, etc), the collected data are transmitted to their final destination, usually a fusion center, at appropriate times.

Sensor networks haven been identified as one of the most important technologies in the 21st century [3], and they have a wide range of applications, including environment and habitat monitoring, health care, home and office automation and traffic control [4]. Although tremendous progress has been made in the past few years in making sensor network an enabling technology, many challenging problems remain to be solved, e.g, network topology control and routing, collaborative signal collection and information processing, and synchronization [5].

In particular any practical design must fully consider the constraints posed by the limited processing capability and energy supply of each individual sensor. We investigated such constraints in [6] by looking at LQG control over a wireless sensor network. We presented a sensor tree reconfiguration algorithm to meet a specified level of control performance in such a way that the total energy usage of the active sensor nodes in the tree is minimized.

However when a sensor node is not a leaf node, it not only needs to send a measurement data packet, but also needs to receive and forward data packets from its child nodes. As receiving a packet also costs considerable amount of energy [4], in general those sensor nodes that are closer to the fusion center consume more energy than those that are far away. Consequently, the former sensor nodes die quickly than the latter ones. Define the network lifetime to be the first time that any one of the sensors dies due to running out of battery. In [6], although the total sensor energy consumption is minimized, maximization of network lifetime is not guaranteed.

The main contribution of this paper are the construction of a set of *good* sensor trees which have different energy costs of individual sensors and scheduling of these sensor trees in such a way that the network lifetime is maximized.

### B. Related Work

The rapid developments of wireless and sensor technologies enable drastic change of the architecture and embedded intelligence in these systems. The theory and design tools for these systems with spatially and temporally varying control demands are not well developed, but there are a lot of current research.

Kalman filtering under certain information constraints, such as decentralized implementation, has been extensively studied [7]. Implementations for which the computations are distributed among network nodes is considered in [8]–[10]. Kalman filtering over lossy networks is considered in [11], [12]. The interaction between Kalman filtering and how data is routed on a network seems to be less studied. Routing of data packets in networks are typically done based on the distance to the receiver node [13]. Some recent work addresses how to couple data routing with the sensing task using information theoretic measures [14]. An heuristic algorithm for event detection and actuator coordination is proposed in [15]. For control over wireless sensor networks, the experienced delays and packet losses are important parameters. Randomized routing protocols that gives probabilistic guarantees on delay and loss are proposed in [16], [17].

A compensation scheme in the controller for the variations on the transport layer that such routing protocols give rise is

∗: Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91106. Email: {shiling, murray}@cds.caltech.edu.

†: Computer Science, California Institute of Technology, Pasadena, CA 91106. Email: acapponi@cs.caltech.edu.

‡: School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. Email: kallej@ee.kth.se.

Tel: (626) 395-2313, Fax: (626) 395-6170.

The work by L. Shi and R. M. Murray is supported in part by AFOSR grant FA9550-04-1-0169. The work by A. Capponi is supported by a fellowship granted by the Social and Information Sciences Laboratory at Caltech. The work by K. H. Johansson was supported by the Swedish Research Council and the Swedish Foundation for Strategic Research.

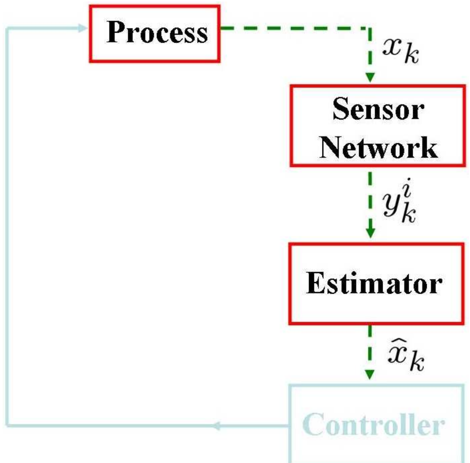


Fig. 1. State Estimation Using a Wireless Sensor Network

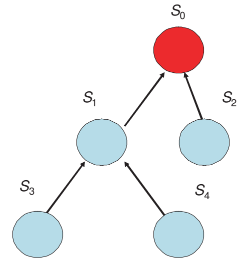


Fig. 2. An Example of a Sensor Tree

presented in [18]. A robust control approach to control over multi-hop networks is discussed in [19].

The rest of the paper is organized as follows. In Section II, the sensor tree construction and scheduling problems are formulated, and some previous results on optimal estimation over sensor trees and tree energy minimization problems are reviewed. In Section III, we propose an algorithm to construct a set of sensor trees. In Section IV, we solve the problem of sensor tree scheduling via linear programming. Examples are given in Section V to demonstrate the algorithms developed. Concluding remarks are given in Section VI.

## II. Problem Set-up and Previous Work

### A. Plant and Sensor Models

Consider the problem of state estimation over a wireless sensor network (Fig. 1). The process dynamics is described by

$$x_k = Ax_{k-1} + w_{k-1}, \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state of the process and $w_k \in \mathbb{R}^{n_x}$ is the process noise which is white Gaussian, zero-mean and with covariance matrix $Q \in \mathbb{R}^{n_x \times n_x}, Q \geq 0$.

A wireless sensor network is used to measure the state. When $S_i$ takes a measurement of the state in Eqn (1), it returns

$$y_k^i = H_i x_k + v_k^i, \tag{2}$$

where $y_k^i \in \mathbb{R}^{m_i}$ is the measurement, $v_k^i \in \mathbb{R}^{m_i}$ is the measurement noise which is white Gaussian, zero-mean and with covariance matrix $\Pi_i \in \mathbb{R}^{m_i \times m_i}, \Pi_i > 0$.

Each sensor can potentially communicate via a single-hop connection with a subset of all the sensors by adjusting its transmission power. Let us introduce a sensor $S_0$, which we denote as the fusion center and consider a tree $T$ with root $S_0$ (see Fig. 2). We suppose that there is a non-zero single-hop communication delay, which is smaller than the sampling time of the process. All sensors are synchronized in time, so the data packet transmitted from $S_i$ to $S_0$ is delayed one sample when compared with the parent node of $S_i$.

### B. Sensor Energy Cost Model

We assume that the sensor nodes are battery powered. Sensors spend energy in many ways, *i.e.*, packet transmission and reception, idle listening, computing, etc [4]. By appropriately designing the MAC protocol such as TDMA protocol, packet transmission and reception dominate the total energy usage. Define $e_{tx}^i$ as the energy cost for $S_i$ sending a measurement packet to its parent node and $e_{rx}^i$ as the energy cost for $S_i$ receiving a measurement packet from one of its children. The transmission power $e_{tx}^i$ typically grows rapidly with the distance to the receiver [1], and $e_{rx}^i$ is about the same for each sensor. Without loss of generality, we write $e_{rx}^i = e_{rx}$. Given a tree $T$ representing the sensor communications with $S_0$, the total energy cost is then given by

$$e(T) = \sum_{S_i \in T} e_{tx}^i + |T| e_{rx} \tag{3}$$

where $|T|$ denotes the number of nodes in tree $T$.

### C. Previous Work

In [6], the following two problems are solved.

*Problem 2.1:* Given a tree $T$ representing sensor communications with $S_0$, what is the optimal state estimate $\hat{x}_k(T)$ and its associated steady state error covariance $P_\infty(T)$ computed at $S_0$?

The following result is obtained.

*Theorem 2.2:* [6] Consider a sensor tree $T$ with depth $h$.

1) $\hat{x}_k$ and $P_k$ can be computed from $h$ Kalman filters as

$$\begin{aligned}
& (\hat{x}_{k-h+1}, P_{k-h+1}) \\
= \ & \mathbf{KF}(\hat{x}_{k-h}, P_{k-h}, Y_k^{k-h+1}, C_h, R_h) \\
& \vdots \\
& (\hat{x}_{k-1}, P_{k-1}) \\
= \ & \mathbf{KF}(\hat{x}_{k-2}, P_{k-2}, Y_k^{k-1}, C_2, R_2) \\
& (\hat{x}_k, P_k) \\
= \ & \mathbf{KF}(\hat{x}_{k-1}, P_{k-1}, Y_k^k, C_1, R_1)
\end{aligned}$$

[1] An estimate of $e_{tx}^i$ can be be computed based on the considered wireless technology. A common model is that if the distance between $S_i$ and $\mathrm{Par}(S_i)$ is $d_i$, then $e_{tx}^i = \beta_i + \alpha_i (d_i)^{n_i}$, where $\beta_i$ represents the static part of the energy consumption and $\alpha_i (d_i)^{n_i}$ the dynamic part. The path loss exponent $n_i$ is typically between 2 and 6.

2) If the limits exist, $P_\infty$ satisfies[2]

$$P_\infty = \tilde{g}_{C_1} \circ g_{C_2} \circ \cdots \circ g_{C_{h-1}}(P) \qquad (4)$$

where $P$ is the unique solution to $g_{C_h}(P) = P$.

*Problem 2.3:* How should the tree $T$ be established such that $e(T)$ is minimized yet $P_\infty(T) \leq P_{\text{desired}}$?

A *Tree Reconfiguration Algorithm* is proposed in [6] such that a final tree $T'$ is returned via a finite iterative reconfiguration of the given initial tree $T_0$. $T'$ has the property that $P_\infty(T') \leq P_{\text{desired}}$ and $T'$ approximates the minimum energy tree.

### D. Problems of Interest

The drawback of the *Tree Reconfiguration Algorithm* is that it does not consider the energy consumption of each individual sensor, and those sensors that are closer to the fusion center usually consume more energy than those that are far away. Consequently, the former sensor nodes die quickly than the latter ones, which make the overall network lifetime small. We are therefore interested in solving the following problems.

*Problem 2.4:* How can we generate a set of *good* trees with different energy consumption for each individual sensor node?

*Problem 2.5:* Given a set of good trees

$$\mathcal{T} = \{T_j : j = 1, \cdots, M\}$$

how can we schedule these trees in such a way that the network lifetime is maximized?

In Section III, we propose a *Tree Construction Algorithm* that solves Problem 2.4 and in Section IV, we solve Problem 2.5 via linear programming.

## III. Tree Construction

The proposed *Tree Construction Algorithm* consists of two main subroutings which are the *Random Initialization Algorithm* and the *Topology Improvement Algorithm*. The overall algorithm is presented in Fig. 3.

### A. Random Initialization Algorithm

Define the following quantities.

- $\mathcal{S}_{j-hop} \triangleq \{S_i : S_i \text{ is } j\text{-hop away from } S_0\}$.
- $\mathcal{S}^c \triangleq \{S_i : S_i \text{ is not included in } T \text{ yet}\}$.

The intuitive idea of the *Random Initialization Algorithm* is that $\mathcal{S}_{j-hop}, j = 1, \cdots, h$ are randomly determined in sequence until all $S_i$'s are included in the tree.

After the execution of the *Random Initialization Algorithm*, an initial tree of depth $h$ is constructed with $|\mathcal{S}_{j-hop}| = n_j, j = 1, \cdots, h$, and $\sum_{j=1}^{h} n_j = N$.

If $n_1 = N$, then the algorithm returns the star tree, i.e, all sensor nodes connect to $S_0$ directly. The complexity of the algorithm is easy seen to be $O(N)$.

[2]Due to the space limitation, we omit the lengthy definitions of $C_i$, etc which can be found in [6]. Readers may find the theorem stated here more complete than the original one.

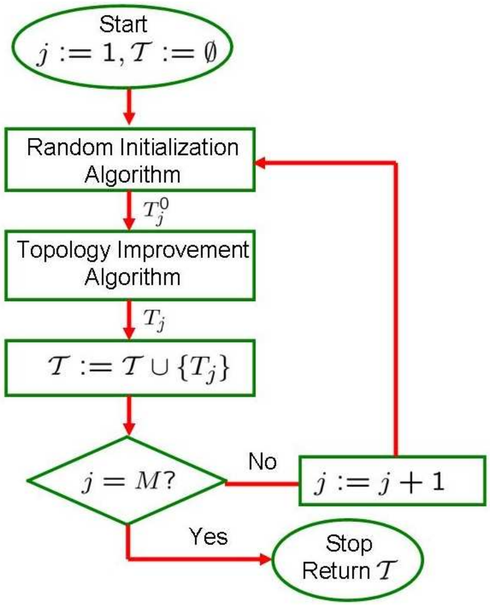


Fig. 3. Tree Construction Algorithm

**Algorithm 1** Random Initialization Algorithm

$h := 0$
$T := \{S_0, \emptyset\}$
$\forall j \; \mathcal{S}_{j-hop} := \emptyset$
$\mathcal{S}^c = \{S_1, \cdots, S_N\}$
**while** $(\mathcal{S}^c \neq \emptyset)$ **do**
  $h := h + 1$
  Pick $n_h$ from $(1, |\mathcal{S}^c|)$ uniformly at random.
  $l := 1$
  **while** $(l \leq n_h)$ **do**
    Pick any $S_p \in \mathcal{S}^c$ and any $S_q \in \mathcal{S}_{(h-1)-hop}$ uniformly at random.
    Connect $S_p$ to $S_q$.
    $\mathcal{S}^c := \mathcal{S}^c \setminus \{S_p\}$
    $T := T \cup \{S_p, (S_p, S_q)\}$
    $\mathcal{S}_{(h)-hop} := \mathcal{S}_{(h)-hop} \cup \{S_p\}$
    $l := l + 1$
  **end while**
**end while**

### B. Topology Improvement Algorithm

Since the previous algorithm randomly constructs the initial tree, some sensor communication paths may be established inefficiently, i.e, some sensors use more energy yet have more hops to communicate with $S_0$. *The Topology Improvement Algorithm* aims to remove this inefficiency.

When $S_i$ is connected to $S_p$, we define $\tau_{ip}$ as the number of hops between $S_i$ and $S_0$, and $e_{ip}$ as the transmission energy cost of $S_i$. Similarly we define $\tau_{i0}$ and $e_{i0}$ for $S_i$ in the initial tree constructed by the *Random Initialization Algorithm*.

We consider modifying the path of $S_i$ ($S_i \in \mathcal{S}_{j-hop}, j \geq 2$) in the initial tree only if there exists $S_p \in \mathcal{S}_{j-hop}, j \leq$

$\tau_{i0}-1$ such that either $e_{ip} < e_{i0}$ or $e_{ip} = e_{i0}$ and $\tau_{ip} < \tau_{i0}$, in which cases, we reconnect $S_i$ to $S_p$. The first condition corresponds to reducing the energy consumption of $S_i$ yet not making the hops between $S_i$ and $S_0$ larger; the second condition corresponds to making the hops between $S_i$ and $S_0$ smaller yet not increasing its energy consumption. Define $F_i$ as the indicator function for $S_i$. $F_i = 1$ means that $S_i$ has already been examined for possible improvement and not otherwise. The full algorithm is presented below.

**Algorithm 2** TOPOLOGY IMPROVEMENT ALGORITHM

```
∀i F_i := 0
∀S_i ∈ S_{j-hop}, j ≤ 1, F_i := 1
while ∃F_i = 0 do
    F_i := 1
    for all S_p ∈ S_{(j)-hop}, j ≤ τ_{i0} − 1 do
        compute (τ_{ip}, e_{ip}).
    end for
    remove all (τ_{ip}, e_{ip}) such that e_{ip} > e_{i0}
    let S_q be the one in the remaining sensors that has
    the least τ_{ip}.
    if e_{iq} < e_{i0} or (e_{iq} = e_{i0} and τ_{iq} < τ_{i0}) then
        reconnect S_i to S_q
        update S_{(j)-hop}, j ≤ τ_{i0}
    end if
end while
```

Notice that $F_i$ is set to be 1 for all $S_i \in \mathcal{S}_{j-hop}, j \leq 1$, as for those sensor nodes that are 1 hop away from $S_0$, no improvements can be made that further reduce the energy consumption (and maintain the same hop numbers) or reduce the hop numbers.

The worst case complexity of the algorithm is easily seen to be $O(N^2)$. Therefore the overall complexity of the *Tree Construction Algorithm* in the worst case is $O(MN^2)$.

## IV. OPTIMAL TREE SCHEDULING

In Section III, we construct a set of initial trees, which, as an input to the *Tree Reconfiguration Algorithm* in [6], produces a set of trees $\mathcal{T}$ such that for any $T_j \in \mathcal{T}, j = 1, \cdots, M$,

$$P_\infty(T_j) \leq P_{\text{desired}}.$$

Let us define $e_{ij}$ as the total energy cost of $S_i$ in $T_j$, *i.e.*,

$$e_{ij} = e_{tx}^{ij} + e_{rx}^{ij}$$

where $e_{tx}^{ij}$ and $e_{rx}^{ij}$ are the energy costs for $S_i$ transmitting and receiving a data packet in $T_j$ respectively. Further define $\Pi_i$ as the initial energy level of $S_i$. As Problem 2.5 stated, we would like to schedule $T_j$ in such a way that the network lifetime is maximized. Without loss of generality, we assume that $T_j$ is used for $t_j$ times in sequence and this is repeated afterwards. Thus

$$\frac{\Pi_i}{\sum_{j=1}^M t_j e_{ij}}$$

gives the maximum cycle that each $S_i$ can operate before its battery is fully consumed. As a result, the network lifetime $L$ can be computed as

$$L = \min_i \frac{\sum_{j=1}^M t_j \Pi_i}{\sum_{j=1}^M t_j e_{ij}} \tag{5}$$

We can therefore write Problem 2.5 as

*Problem 4.1:*

$$\max_{(t_1,\cdots,t_M)} \min_i \frac{\sum_{j=1}^M t_j \Pi_i}{\sum_{j=1}^M t_j e_{ij}}$$

subject to

$$t_j \geq t_{\min}, j = 1, \cdots, M$$

where $t_j \geq t_{\min}$ is added to make sure the estimation will enter steady state after some transient times.

To solve Problem 4.1, we can write it equivalently as

$$\max_{(t_1,\cdots,t_M)} L$$

subject to

$$L \sum_{j=1}^M t_j e_{ij} \leq \sum_{j=1}^M t_j \Pi_i, i = 1, \cdots, N$$
$$t_j \geq t_{\min}, j = 1, \cdots, M$$

Notice that the first constraint involves both $L$ and $t_j$, so we cannot solve the problem via linear programming directly. Let us define

$$e_i^{\min} = \min_j \{e_{ij}\},$$

and

$$\bar{L} = \min_i \frac{\Pi_i}{e_i^{\min}},$$

then we obtain

$$\begin{aligned} L &= \min_i \frac{\sum_{j=1}^M t_j \Pi_i}{\sum_{j=1}^M t_j e_{ij}} \\ &\leq \min_i \frac{\sum_{j=1}^M t_j \Pi_i}{\sum_{j=1}^M t_j e_i^{\min}} \\ &= \min_i \frac{\Pi_i}{e_i^{\min}} \\ &= \bar{L} \end{aligned}$$

Given $L$, let us define $\mathcal{P}(L)$ as the feasibility problem to

$$\max_{(t_1,\cdots,t_M)} 1$$

subject to

$$L \sum_{j=1}^M t_j e_{ij} \leq \sum_{j=1}^M t_j \Pi_i, i = 1, \cdots, N$$
$$t_j \geq t_{\min}, j = 1, \cdots, M$$

Now $\mathcal{P}(L)$ can be solved via linear programming as follows.

$$\min_{(t_1,\cdots,t_M,u)} u$$

subject to

$$L\sum_{j=1}^{M} t_j e_{ij} \leq \sum_{j=1}^{M} t_j \Pi_i + u, i = 1,\cdots,N$$
$$t_j \geq t_{\min} - u, j = 1,\cdots,M$$

If the minimizers $(t_1^*,\cdots,t_M^*,u^*)$ satisfies $u^* \leq 0$, then the vector $(t_1^*,\cdots,t_M^*)$ satisfies the feasibility problem $\mathcal{P}(L)$. With the definition of $\mathcal{P}(L)$, we can find the solution to Problem 4.1 via the following *Binary Search Algorithm*.

**Algorithm 3** BINARY SEARCH ALGORITHM

```
t := 1
l := 1
u := x̄
L(0) := 1
L(t) := ⌈(l+u)/2⌉
while L(t) ≠ L(t−1) do
    if P(L(t)) is feasible then
        l := L(t)
        L(t) := ⌈(l+u)/2⌉
    else
        u := L(t)
        L(t) := ⌈(l+u)/2⌉
    end if
    t := t+1
end while
```

*Theorem 4.2:* The *Binary Optimal Search Algorithm* returns the optimal solution $L^*$ with worse case time complexity $O(\log \bar{L}) * O(\mathcal{P}(L))$, and the optimal scheduling $(t_1^*,\cdots,t_M^*)$ is obtained from solving $\mathcal{P}(L^*)$.

*Proof:* The time complexity of the algorithm is trivial to show, and we only need to show that if $\mathcal{P}(L^*)$ is feasible, then for any $L \leq L^*$, $\mathcal{P}(L)$ is also feasible. Since $\mathcal{P}(L^*)$ is feasible,

$$L^*\sum_{j=1}^{M} t_j^* e_{ij} \leq \sum_{j=1}^{M} t_j^* \Pi_i + u^*, i = 1,\cdots,N$$
$$t_j^* \geq t_{\min} - u^*, j = 1,\cdots,M$$

Thus the same $(t_1^*,\cdots,t_M^*,u^*)$ automatically satisfy

$$L\sum_{j=1}^{M} t_j^* e_{ij} \leq \sum_{j=1}^{M} t_j^* \Pi_i + u^*, i = 1,\cdots,N$$
$$t_j^* \geq t_{\min} - u^*, j = 1,\cdots,M$$

for any $L < L^*$. Hence $\mathcal{P}(L)$ is also feasible. ■

## V. EXAMPLES

Due to the space limitation, we only provide examples to demonstrate the *Tree Construction Algorithm* and leave it to future work to combine the *Tree Construction Algorithm* and the *Tree Reconfiguration Algorithm* in [6], and apply the scheduling algorithm presented in previous section to the resulting sensor trees.

We consider the following example with 6 sensors communicating to $S_0$. The initial sensor topology is shown in Fig. 4.

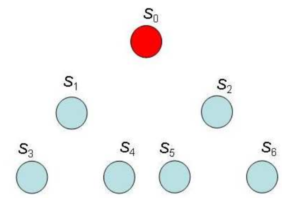


Fig. 4. Initial Sensor Topology

Let $d_{pq}$ denote the relative physical distance between sensor $S_p$ and $S_q$. Assume the transmission energy cost for $S_q$ when the receiving node is $S_q$ is given as $d_{pq}^2$, i.e, the larger the distance, the higher the energy cost.

Suppose $M = 3$ and we run the *Tree Construction Algorithm* three times. The following initial trees (Fig. 5 - 7) are returned.

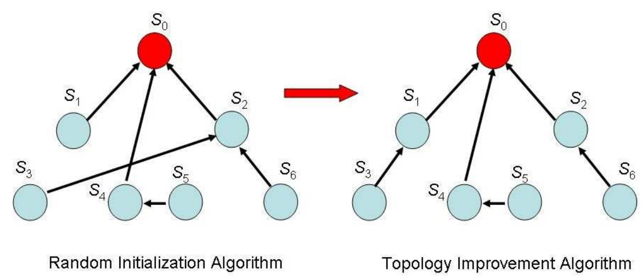


Fig. 5. Tree Construction Algorithm: 1st Round

In the first round, during the execution of the *Random Initialization Algorithm*

- $n_1 = 3, n_2 = 3$
- $\mathcal{S}_{1-hop} = \{S_1, S_2, S_4\}$
- $\mathcal{S}_{2-hop} = \{S_3, S_5, S_6\}$

Then the *Topology Improvement Algorithm* is executed and $S_3$ is reconnected to $S_1$ as its energy consumption is reduced.

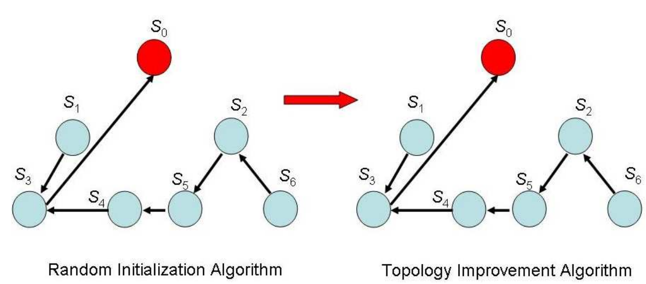


Fig. 6. Tree Construction Algorithm: 2nd Round

In the second round, during the execution of the *Random Initialization Algorithm*

- $n_1 = 1, n_2 = 2, n_3 = 1, n_4 = 1, n_5 = 1$
- $\mathcal{S}_{1-hop} = \{S_3\}$
- $\mathcal{S}_{2-hop} = \{S_1, S_4\}$
- $\mathcal{S}_{3-hop} = \{S_5\}$
- $\mathcal{S}_{4-hop} = \{S_2\}$
- $\mathcal{S}_{5-hop} = \{S_6\}$

Then the *Topology Improvement Algorithm* is executed, but in this case, no improvement is made.

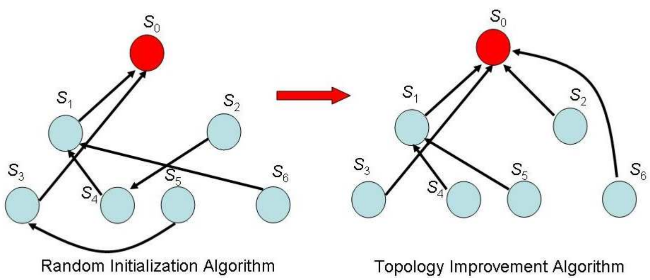


Fig. 7. Tree Construction Algorithm: 3rd Round

In the third round, during the execution of the *Random Initialization Algorithm*

- $n_1 = 2, n_2 = 3, n_3 = 1$
- $\mathcal{S}_{1-hop} = \{S_1, S_3\}$
- $\mathcal{S}_{2-hop} = \{S_4, S_5, S_6\}$
- $\mathcal{S}_{3-hop} = \{S_2\}$

After the *Topology Improvement Algorithm* is executed, $S_2, S_5, S_6$ are reconnected to $S_0, S_1, S_0$ respectively. However, in this case, we can do better by reconnecting $S_5$ to $S_2$ as the energy consumption of $S_5$ will be further reduced yet the hop number between $S_5$ and $S_0$ remains the same. The reason that $S_5$ is reconnected to $S_1$ instead is that $S_2$ initially has a larger hop number and hence $S_5$ is modified first according to the algorithm. Hence the *Topology Improvement Algorithm* only *improves* the tree returned by the *Random Initialization Algorithm* and does not necessarily produced the optimal tree. We leave it to future work to construct better algorithms.

## VI. Conclusion and Future Work

In this paper, we have considered the sensor tree construction and scheduling problems for state estimation over a wireless sensor network. A heuristic algorithm is proposed that constructs an initial set of sensor trees. An optimal tree scheduling algorithm having polynomial time complexity is proposed that maximizes the network lifetime.

There are a few extensions of the current work that we will pursue in the future which include combining the *Tree Construction Algorithm* with the *Tree Reconfiguration Algorithm* in [6]; closing the loop based on the estimation scheme; experimentally evaluate the algorithms developed in the paper; consider packet drops issues in the communication link which is often seen due to the nature of wireless communications.

**Acknowledgement:** The authors would like to thank Prof Magnus Egerstedt for his insightful comments on the work in [6] which partially motivated this work.

## References


[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of wireless sensor networks," *IEEE Computer, Special Issue in Sensor Networks*, Aug 2004.

[2] http://en.wikipedia.org/wiki/Wireless_sensor_network.

[3] "21 ideas for the 21st century," *Business Week*, pp. 78–167, Aug.30 1999.

[4] N. P. M. (Ed.), *Sensor Networks and Configuration*. Springer, 2007.

[5] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, Aug 2003.

[6] L. Shi, K. H. Johansson, and R. M. Murray, "Change sensor topology when needed: How to efficiently use system resources in control and estimation over wireless networks." Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, Dec 2007.

[7] D. Siljak, *Large-Scale Dynamic Systems: Stability and Structure*. North-Holland, New York, 1978.

[8] P. Alriksson and A. Rantzer, "Distributed kalman filtering using weighted averaging." Proc. 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006.

[9] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters." Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference, 2005.

[10] D. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed kalman filtering in sensor networks with quantifiable performance." Proceedings of the 4th International Conference on Information Processing in Sensor Networks, 2005.

[11] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[12] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "Networked control systems: Analysis and design," *To appear in the Proc. of IEEE, Special Issue on Networked Control Systems*, 2007.

[13] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1991.

[14] J. Liu, F. Zhao, and D. Petrovic, "Information-directed routing in ad hoc sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, April 2005.

[15] E. Ngai, M. Lyu, and J. Liu, "A real-time communication framework for wireless sensor-actuator networks." Aerospace Conference, 2006 IEEE, 2006.

[16] A. Bonivento, C. Fischione, and A. Sangiovanni-Vincentelli, "Randomized protocol stack for ubiquitous networks in indoor environment," *IEEE CCNC*, vol. 1, April 2006.

[17] W. Lai and I. C. Paschalidis, "Routing through noise and sleeping nodes in sensor networks: latency vs. energy trade-offs." Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, Dec 2006, pp. 551 – 555.

[18] E. Witrant, P. G. Park, M. Johansson, C. Fischione, and K. H. Johansson, "Predictive control over wireless multi-hop networks." IEEE Conference on Control Applications, Singapore, 2007. Submitted.

[19] A. Panousopoulou, G. Nikolakopoulos, A. Tzes, and J. Lygeros, "Experimental evaluation of a mobile ad-hoc networked (manet)controlled system." Proc. 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006.