Supplementary File of the Paper "On Maximizing the Lifetime of Wireless Sensor Networks Using Virtual Backbone Scheduling"

Yaxiong Zhao, Jie Wu, *Fellow, IEEE*, Feng Li, *Member, IEEE*, and Sanglu Lu, *Member, IEEE*

1 The proof of the NP-Hardness of the MLBS problem

The decision version of the MLBS problem is defined as follows:

Definition 1 (Decision problem of MLBS): Given a graph $G = (V, E)$ and a constant K, does a schedule of backbones, $\{\langle B_1, T_1\rangle, \cdots, \langle B_p, T_p\rangle\}$, exist such that $T_1 + T_2 + \cdots + T_p \geq K$ and for each vertex $v_i \in V$, v_i appears in B_1, B_2, ..., B_p and the total energy consumed by v_i is at most L_i? Here, L_i is the initial energy of sensor v_i.

We call the above problem the K-MLBS problem. Thus, the MLBS problem is the optimization problem of the K-MLBS problem, which is to find a schedule of maximum K. We show in Theorem 1 below that the K-MLBS problem is NP-Complete.

Theorem 1: K-MLBS is NP-Complete.

Proof: We assume, without generality, that each sensor node has L units of initial energy. Given a schedule of backbones, we can verify, in time linear to the number of sensor nodes in the network, whether or not:

- They are CDSs of the given network.
- The sum of their working time is larger than K.
- The energy consumed by each sensor node that belongs to any of the backbones at the end of the network lifetime is less than its initial energy.

Hence, K-MLBS is in NP. In order to prove the NP-Hardness of the MLBS problem, we reduce from the *Maximum Set Cover* (MSC) problem, as defined in [3], to it. We quote the definition of the decision version of the MSC problem from [3] here:

- *Yaxiong Zhao and Jie Wu are with the Computer and Information Sciences Department of Temple University, E-mail: {yaxiong.zhao, jiewu@}temple.edu.*
- *Feng Li is with the School of Engineering and Technology, IUPUI: Indiana University-Purdue University Indianapolis, E-mail: fengli@iupui.edu.*
- *Sanglu Lu is with the State Key Laboratory for Novel Software Technology, Nanjing University, P. R. China, E-mail: sanglu@nju.edu.cn.*

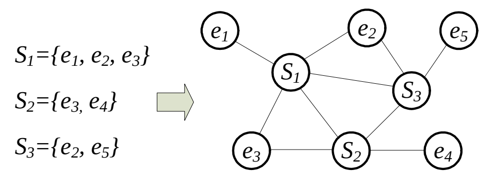


Fig. 1. The transformation of a K-MSC instance to a K-MLBS instance. Each node in the graph corresponds to a set or an element of the K-MSC problem. All of the nodes that represent the sets are fully connected. A node that represents a set in the transformed graph connects with all of the nodes that represent all of its elements.

Definition 2 (The Decision Version of the MSC problem): Given a collection C of subsets of a finite set R, find a family of set-covers, $S_1, ..., S_p$ with time weights, $t_1, ..., t_p \in [0, 1]$, such that $t_1 + ... + t_p \geq k$, and for each subset s in C, s appears in $S_1, ..., S_p$ with a total weight of at most 1, where 1 is the life time of each sensor node.

We denote the decision version of the MSC problem as the K-MSC problem. The K-MSC problem has quite a similar structure to the K-MLBS problem. We then transform an instance of the K-MSC problem to an instance of the K-MLBS problem.

For a given instance of the K-MSC problem, we construct a graph, $G(V, E)$, with the following operations:

- V=$\{S_1, S_2, ..., S_n, e_1, e_2, ..., e_m\}$;
- E=$\{\langle S_i, S_j\rangle | S_i, S_j \in S\} \cup \{\langle S_i, e_j\rangle | S_i \in S, e_j \in \mathbb{E}, and e_j \in S_i\}$;

In Fig. 1, a transformation based on the above rules is given. Each vertex in G is labeled a set in S or an element in E. An edge connects two vertices in G if: 1) one of the vertexes is labeled an element and the other is labeled a set, and the element is in the set; 2) two vertexes are all labeled sets. For each MSC instance, there is one and only one such graph that corresponds to it.

We claim that the K-MSC instance receives a YES answer if and only if the K-MLBS in the above graph G has a YES answer. To prove this, note that all nodes corresponding to S are mutually connected, and thus a CDS of G can be formed only by these nodes without affecting the maximum network lifetime. The K in the K-MLBS problem directly corresponds to that of the MSC problem. For the reverse case, it is obvious that a valid K-MSC solution also directly applies to the corresponding K-MLBS instance.Because the K-MSC problem is NP-Complete, as was proved in [3], we conclude that K-MLBS is NP-Complete. Therefore the MLBS problem, as an optimization version of the K-MLBS problem, is NP-Hard. □

2 Related work

2.1 Duty-cycling MAC protocols

Duty-cycling is built into most of the existing MAC protocols. S-MAC [20] introduced the idea of duty-cycling and scheduled sleeping into the MAC protocol. Sensor nodes follow a periodic active/sleep cycle, and the sensor nodes that are close to each other synchronize their active cycles together to reduce transmission delay. Sleep scheduling techniques can be classified into two categories: synchronous [12], [20] and asynchronous [1], [15]. Synchronous scheduling is more complicated, but offers lower communication delay. In synchronous scheduling, sensor nodes know their neighbors' wakeup times and try to find schedules that can minimize energy consumption and communication delays. For example, in S-MAC [20], sensor nodes overlap their active time with their neighbors' in order to reduce the waiting time to transfer messages. On the contrary, in asynchronous scheduling, sensor nodes choose schedules independently, knowing nothing about their neighbors. As a result, in asynchronous scheduling, a sender needs to send a persistent preamble to notify the receiver [15], [1] (sender-initiated), or receivers broadcast a notification to all of their neighbors to solicit potential senders (receiver-initiated) [16].

As for the theoretical aspects of the sleep scheduling problem, it has been proven in [13] that the problem of finding the sleep schedule with the minimum end-to-end delay is, in general, NP-Hard. However, optimal scheduling can be found in polynomial time for tree and ring networks. The energy-delay trade-off in tree-shaped WSNs has been studied in [5]. In [11], five wakeup patterns used in previous works [2], [8], [12] are summarized. All of the methods introduced above use homogeneous scheduling, which does not consider the redundancy in the network.

Channel assessment and detection are key techniques to determine the availability of links. B-MAC [15] addresses the *Low Power Listening* (LPL) problem, which helps a sensor node check the channel occupancy and determine its availability through simple operations. X-MAC [1] further reduces the energy consumption wasted in the preamble in B-MAC by using a smart probing technique.

2.2 Virtual backbone construction and rotation

A virtual backbone is comprised of a set of sensor nodes of a WSN and is used as a communication infrastructure. The *Connected Dominating Set* (CDS) is a widely-used abstract of the virtual backbone. The *Minimum Connected Dominating Set* (MCDS) problem is NP-Hard for both general graphs [9] and unit disk graphs [4]. Wu and Li's *Marking Process* (MP) [19], Wu and Dai's self-pruning rules [18], Rule K [17], and the extended coverage condition [6], are the representative algorithms. In [7], several distributed algorithms are proposed to construct a *K-vertex connected K-coverage CDS* (KCDS). We use these algorithms extensively in VBS.

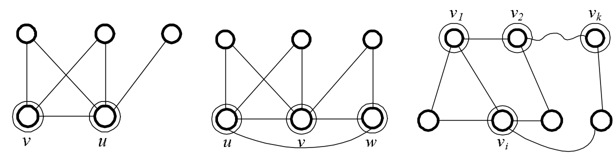
The idea of rotating multiple backbones to prolong the network lifetime has already been proposed to save energy in WSNs. A *Connected Domatic Partition* (CDP) is a partition of the sensor nodes of a graph into disjoint sets, where each set of the partition is a CDS. A recent work, [14], presents a CDP-based backbone rotation scheme. This work focused on designing distributed approximation algorithms for the CDP problem. We show, via simulations, that the CDP-based approach is limited in prolonging the network lifetime of WSNs.

The multi-parent method proposed in [11] is identical to the CDP-based approach in the algorithmic aspect, but it tries to reduce the communication delay instead of increase the network lifetime. The solution in [11] is based on linear programming and has no distributed implementation. Our work is inspired by [11] and [14], but we seek a different approach to the problem, and our proposed algorithms perform much better.

3 Preliminary on CDS construction algorithms

CDS is a well-studied problem in graph theory. A *Dominating Set* (DS) $\mathbb{D}$ is a set of vertices of a graph $G(V, E)$ that satisfy: each vertex in V is either in $\mathbb{D}$, or it is connected with a node in $\mathbb{D}$. A *Connected Dominating Set* (CDS) is a DS and is fully connected in the original graph. The *Minimum Connected Dominating Set* (MCDS) is the CDS that is formed by the minimum number of nodes.

CDS is a widely-used abstraction of the backbones for WSNs. The *Marking Process* (MP) [19], together with Rules 1&2 and Rule K, are the first distributed CDS construction algorithms. Rules 1&2 are extensions to the MP process. MP works as follows: u sets its marker to T (in-backbone) if there exist two neighbors, v and w of u, that are not directly connected. MP preserves the shortest path length in the induced graph. Rules 1&2 apply to the induced graph G' of MP. Rule 1, as depicted in Fig. 2(a), states that a marked node v on G' can be unmarked if one of its neighbors, u, covers all of its neighbors and u has higher priority. Rule 2 states that a marked node v can be unmarked if all of its neighbors are jointly covered by two of its connected neighbors u and w, and v has the lowest priority. This is depicted in



(a) Rule 1 (b) Rule 2 (c) Rule K

Fig. 2. The illustrations of three CDS construction rules. Double circled nodes are in the backbone.

Fig. 2(b), where v can be unmarked. Rule K in Fig. 2(c) states that for k connected marked nodes $\{v_1, v_2, ..., v_k\}$, v_i can be unmarked if it has the lowest priority and all of its neighbors are covered jointly by the connected subset $\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_k\}$. Here, the priority of nodes can be any attributes of nodes or their combinations, like the residual energy or node ID.

In the WSNs that use VBS, all messages are forwarded by backbone sensor nodes. As a result, the minimum path lengths between sensor nodes in the induced network is probably larger than that in the original network. This problem is called $path\ stretch$. In this section, we analyze the stretches of *Rule 1*, *Rule 2*, and *Rule K* (MP is not studied because it preserves the minimum path length). We define the *stretch factor* of the path connecting u and v as the ratio of the hop-count of the minimum length path connecting them in the induced graph to that in the original graph.

Rule 1 has a maximum stretch factor of 2. As shown in Fig. 2(a), v and its neighbors are covered by u, the path lengths between v and its neighbors increase by one. This property holds for any node pair, therefore, the stretch factor is 2. Fig. 2(b) shows Rule 2. Rule 2 states that if two marked nodes, u and w, are neighbors of a marked node v, v can be unmarked if all of its neighbors are covered by u and w and it has the lowest priority among three nodes. Its stretch factor is 3. Rule K's stretch factor is k because two directly connected nodes in the original graph is now connected by a chain of k-1 nodes in the worst case, resulting in a k-hop path.

4 The complexity of the STG- and VSG-based centralized algorithms

4.1 The complexity of the STG-based algorithm

Suppose that λ backbones are constructed in the enumeration process, and the number of the states in each round is λ. Therefore, the $min()$ function runs in $O(\lambda^2)$, and the selection process runs in $O(\lambda)$. Backbones are constructed in $O(\lambda|V|)$. The round number is, at most, c. So, the algorithm runs in $O(c\lambda^2 + \lambda|V|)$, where c is determined by Eq. 1.

$$C = \frac{|V|\mathbb{E}}{n \times \varepsilon} \tag{1}$$

The round length trades off between the performance of the scheduling results and the complexity. By increasing the round length, we can reduce the scheduling complexity. As defined in Section 2.1 of the main paper, a round can be as short as a single cycle, while the STG gives the longest possible network lifetime. Because the energy consumed in each round increases with the round length, and according to Eq. 1, c decreases with an increase of the round length and so does the complexity of the STG-based scheduling.

4.2 The complexity of the VSG-based algorithm

In Algorithm 2 of the main paper, a CDS is constructed in each repetition. To construct a CDS, we need to examine, at most, $c|V|$ virtual nodes, where c is the maximum round number, as calculated in Eq. 1. In the examination of each virtual node, a $(c\triangle)^k$ operation is performed to calculate the status of the virtual node in question. Here, $\triangle$ is the maximum node degree in the original graph. The reason to multiply c, the maximum round number, is that each node will be transformed into c virtual nodes in the corresponding VSG. k is the scope of the construction algorithm, which means that the algorithm needs to check neighbors up to k-hop(s) away to determine the status of the current virtual node. Thus, the complexity of finding a CDS in a VSG is $O(c|V|(c\triangle)^k)$. Note that c is the maximal round number, and we need to construct, at most, just as many backbones. Hence, the time complexity of VSG scheduling is $O(c^2|V|(c\triangle)^k)$. We can see that the maximum number of rounds c, which is directly determined by the length of each round, has a similar effect here as in the STG-based algorithm.

5 Discussion and extensions

We discuss several issues in the design and implementation of the VBS. We then elaborate on some potential extensions to the VBS.

5.1 Discussion

The implementation of VBS can be centralized or distributed. Although centralized algorithms are generally thought of as more costly than distributed ones, it is not the case in static WSNs. This is because: firstly, a topology change is rare. This is because only the destruction and the energy depletion of the sensor node can change the topology permanently. These two events seldom happen in civilian WSN applications, which are the WSN applications we consider in this paper. Secondly, the sink has a continuous power supply and more powerful computing capability. By putting all of the computation tasks at the sink, we can reduce the overhead of the sensor nodes. Finally, with careful scheduling, the sink can estimate the network conditions without actually collecting messages from sensor nodes, which significantly reduces the overhead. According to these three facts, we should consider STG and VSG when a better performance is more favorable than a smaller overhead. On the other hand, centralized algorithms may be quite complicated because they involve a large amount of information.

In STG-based and VSG-based scheduling, the obtained schedule needs to be disseminated to all of the sensor nodes in the network. The sink broadcasts messages containing a bitmap of all nodes' statuses (backbone or non-backbone) and corresponding working rounds. The actual wakeup patterns are determined by other techniques [11]. A backbone node appends its schedule in the message and re-broadcasts it. A non-backbone sensor node records its parent's schedule and discards the message. For distributed construction, nodes' statuses are determined locally. When new nodes are added into the backbone, they send messages to their neighbors to report their schedules.

VBS has a subtle problem. Since non-backbone sensor nodes turn off their radio in each round, the traffic directed to them will not be delivered within each round. Thus, non-backbone nodes can be reached only at the beginning of each round. Along these lines, the worst-case delay for the traffic directed to non-backbone nodes is the length of a round. In reality, sensor-node-to-sink traffic (data gathering) is most common in WSNs; therefore, we only consider maintaining the connectivity from sensor nodes to the sink in this paper. Another interesting property is that the CDP-based approach in [14] and the multi-parent technique in [11] can be seen as special cases of the VBS.

5.2 Extensions

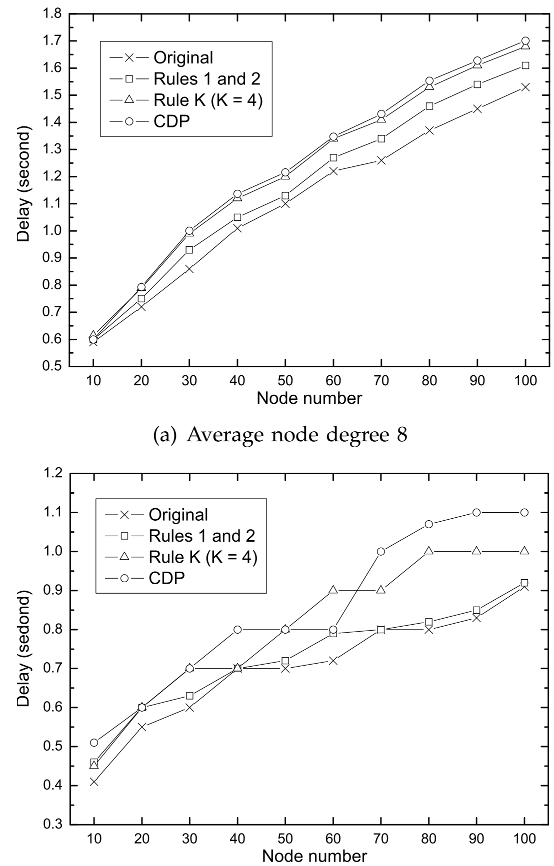
We introduce some potential extensions in this part. Although we use MP together with Rules 1&2 and Rule K in this paper, we can see that the use of different CDS construction algorithms is independent of the execution of the STG/VSG/ILR-based scheduling. This property means that we can plug different CDS construction algorithms into the VBS for various kinds of application requirements. For example, if the application demands high reliability, we can construct a *K-vertex-connected M-coverage Dominating Set* (KMCDS) in order to provide the extra fault tolerance. Or, we can explicitly optimize the delay between sensor nodes and the sink by using algorithms that have bounded stretches.

VBS can also be used to find the minimum delay scheduling with a lifetime constraint. We first apply STG/VSG-based scheduling to find a candidate schedule. Then, we shrink the network lifetime to the required value and increase the working frequency by the same ratio. It is clear that the resultant schedule will meet the lifetime constraint, and its working cycle is minimized.

6 Supplementary simulation results

6.1 Worst-case delay

For the monitoring application, the worst-case delay of detecting some rare event is determined by the worst case communication delay from sensor nodes to the sink.



(a) Average node degree 8

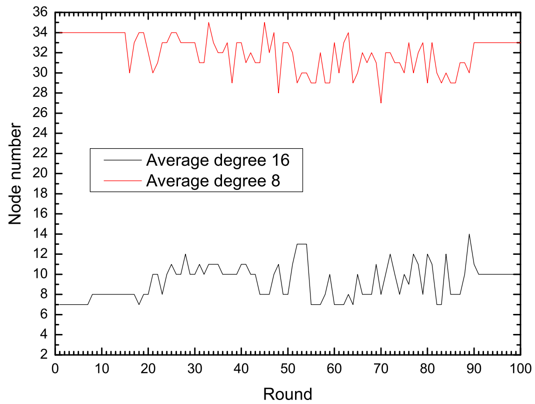
(b) Average node degree 16

Fig. 3. The worst-case delay from sensor nodes to the sink in the sparse and dense networks. The average degree of the sparse network is 8, and that of the dense network is 16. The networks are fully connected.

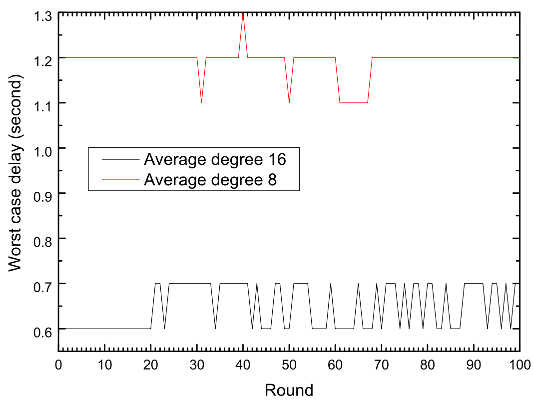
Therefore, in this part, we only study the worst-case delay from the sensor nodes to the sink.

The worst-case delay is obtained by measuring the delay from the farthest sensor node to the sink. We set the duty cycle to $1\ second$ and the per-hop delay between two duty-cycled nodes to $100ms$. Fig. 3 presents the worst-case delays in the dense networks and the sparse networks. The network field is $500 \times 500m^2$. We placed 100 sensor nodes in the field. In order to control the density, we gradually reduce the transmission range of all of the sensor nodes in the network until the required average node degree is reached. The average degrees of the dense and sparse networks are 16 and 8, respectively. The results shown in Fig. 3 are obtained by averaging the results of 10 randomly generated networks. Because the path stretch is only determined by the CDS algorithm, we show curves only of Rules 1&2, Rule K, CDP, and the original networks without applying any sleep scheduling algorithms.

As shown in Fig. 3, the worst-case delay increases with network size. Because the network size and the number of sensor nodes is fixed, the delay in the dense networks is much less than that in the sparse networks. Although



(a) Backbone size



(b) Worst-case delay

Fig. 4. The change of the backbone size and the worst-case delay of ILR in a network of 50 sensor nodes. The initial energy of all sensor nodes is 100 $joules$.

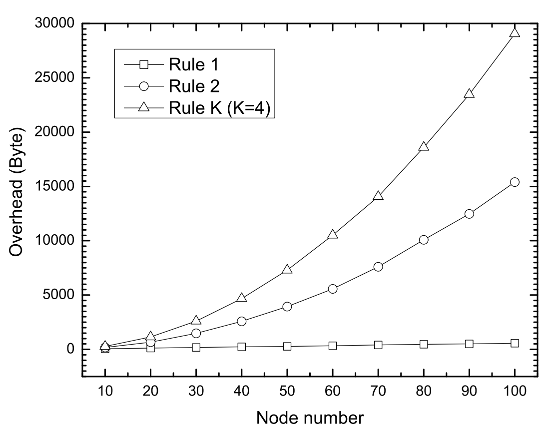


Fig. 5. The overhead measured in the size of messages sent by each sensor node.

the analysis in the Section 3 of the appendix indicates that the path stretches may potentially be very large, the stretch obtained in this set of simulations is actually small, which is within 20%. Considering the significant lifetime extension, the stretch is acceptable.

6.2 Microscopic Behavior and Overhead

In centralized STG and VSG scheduling, all sensor nodes report their one-hop neighbors and residual energy to the sink, which incurs an overhead proportional to the size of each sensor node's neighbor set. We assume that there is an initialization phase when the network is deployed. Taking the hello messages into consideration, the size of messages sent to the sink is irrelevant to duty-cycling.

However, measuring the overhead of ILR is complicated since each sensor node needs to send the topology beyond 1-hop to all of its 1-hop neighbors. Additionally, because of duty-cycling, the broadcast is implemented as multiple unicasts [10], which causes excessive overheads. This problem is exacerbated by the fact that the number of links within a certain hop-count grows beyond linearly with the hop-count.

To study the microscopic behaviors of ILR, we consider the size of the backbone in each round. The change of backbone size using ILR is shown in Fig. 4(a). The networks have 50 sensor nodes. Each sensor node has 100 joules of initial energy. The changing rate increases with the time. This corroborates the intention of the switching probability. The use of the threshold can freeze the replacement when the residual energy of sensor nodes is low. This is expressed in the vertical line near the end of the network lifetime. This is necessary, because in this situation, all of the sensor nodes do not have much energy left; the overheads of replacing sensor nodes may offset or even exceed its benefits.

Fig. 4(b) shows the change of the worst-case delay in the network with the same setting as the above one. Although the backbone size changes quite frequently, the delay does not. This is because the changes happen locally, so their impact on the network-wide property is limited. For example, the minimum hop-count from a sensor node to the sink may be increased in the next round, but the hop-count from the farthest sensor node to the sink can still be the same. We can see the vertical line near the end of the lifetime in Fig. 4(b), which is the result of the threshold-regulated switching probability.

We measure the average size of messages sent by each sensor node in this section. The messages record links within a certain hop-count, which are represented as a pair of node addresses of their two end-points. We assume that a sensor node is identified by a 6-byte MAC address. The results are shown in Fig. 5. Rule 1 needs topology information within 2-hop ranges, Rule 2 needs 3-hop, and Rule K needs (K+1)-hop. Fig. 5 indicates that duty-cycling forces each sensor node to send multiple times the same topology information to all its neighbors, which causes a significantly large overhead when the network becomes denser. This is indicated by the huge gap between three lines, meaning that the quadratically growing link count causes a huge amount of overhead for dense networks. Therefore, centralized algorithms is preferred when the network density is high.

REFERENCES

[1] M. Buettner, G. V. Yee E. Anderson, and R. Han. X-MAC: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proc. of ACM SenSys'06*, pages 307–320, 2006.

[2] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proc. of ACM IPSN '05*, pages 20–27, 2005.

[3] M. Cardei, M. T. Thai, Y. S. Li, and W. l. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of IEEE Infocom'05*, pages 123–131, 2005.

[4] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.

[5] R. Cohen and B. Kapchits. An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. *IEEE/ACM Trans. on Netw.*, 17(2):570–581, 2009.

[6] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. In *IEEE Trans. Parallel Distrib. Syst.*, pages 908–920. IEEE Press, 2004.

[7] F. Dai and J. Wu. On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. *J. Paral. Distrib. Comput.*, 66(7):947–958, 2006.

[8] T. V. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proc. of ACM SenSys' 03*, pages 171–180, 2003.

[9] M. R. Garey and D. S. Johanson. *Computers and Intractability: A guide to the theory of NP-Completeness*. Freeman, New York, NY, USA, 1979.

[10] Shuo Guo, Yu Gu, Bo Jiang, and Tian He. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In *Proc. of MobiCom '09*, pages 133–144. ACM, 2009.

[11] A. Keshavarzian, H. Lee, and L. Venkatraman. Wakeup scheduling in wireless sensor networks. In *Proc. of ACM MobiHoc'06*, pages 322–333, 2006.

[12] 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. IEEE IPDPS'04*, pages 224–235, 2004.

[13] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel. Delay efficient sleep scheduling in wireless sensor networks. In *Proc. of IEEE Infocom'05*, pages 2470–2481, 2005.

[14] C. Misra, R. Mandal. Rotation of cds via connected domatic partition in ad hoc sensor networks. *IEEE Trans. on Mobile Computing*, pages 488–499, 2009.

[15] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proc. of ACM SenSys'04*, pages 95–107, 2004.

[16] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proc. of ACM SenSys '08*, pages 1–14, 2008.

[17] J. Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Trans. on Parallel and Distributed Systems*, 13(9):866–881, 2002.

[18] J. Wu and F. Dai. A generic broadcast protocol in ad hoc networks based on self-pruning. In *Proc. of IEEE IPDPS'03*, pages 29–41, 2003.

[19] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Commun.*, pages 7–14, 1999.

[20] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proc. of IEEE Infocom'02*, pages 1567–1576, 2002.