This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

VBS: Maximum Lifetime Sleep Scheduling for Wireless Sensor Networks Using Virtual Backbones

Yaxiong Zhao, Jie Wu
Dept. of Comp. and Info. Sci.
Temple University
Philadelphia, PA 19122
Email: {yaxiong.zhao, jiewu}@temple.edu

Feng Li
School of Eng. and Tech.
IUPUI
Indianapolis, IN 46202
Email: fengli@iupui.edu

Sanglu Lu
Dept. of Comp. Sci. and Tech.
Nanjing University
Nanjing, P. R. China 210093
Email: sanglu@nju.edu.cn

***Abstract*—Wireless sensor network (WSN) applications require redundant sensors to guarantee fault tolerance. However, the same degree of redundancy is not necessary for multi-hop communication. In this paper, we present a new scheduling method called *virtual backbone scheduling* (VBS). VBS employs *heterogeneous scheduling*, where backbone nodes work with duty-cycling to preserve network connectivity, and non-backbone nodes turn off radios to save energy. We formulate a *maximum lifetime backbone scheduling* (MLBS) problem to maximize the network lifetime using this scheduling model. Because the MLBS problem is NP-hard, two approximation solutions based on the *schedule transition graph* (STG) and *virtual scheduling graph* (VSG) are proposed. We also present an *iterative local replacement* (ILR) scheme as an distributed implementation of VBS. The path stretch problem is analyzed in order to explore the impact of VBS on the network structure. We show, through simulations, that VBS significantly prolongs the network lifetime under extensive conditions.**

***Index Terms*—Wireless sensor networks (WSNs), virtual backbone, NP-hard.**

I. Introduction

The sensors in wireless sensor networks (WSNs) are expected to work on batteries for several months or a few years. The sensors' energy consumption is the paramount concern in WSNs. Energy saving techniques are critical to the success of WSNs. Currently, duty-cycling is a well-studied and widely-used power saving technique [1], [2], [3], [4]. A limitation of existing duty cycling approaches is that all the sensors' cycles are identical, which does not consider the redundancy in WSNs.

On the other hand, many WSN applications require redundant sensors for the fault tolerance of applications. However, the same degree of redundancy is not necessary for multi-hop communications. The reason is two-fold: 1) sensors are immobile, 2) traffic is low. So, the error rate in wireless transmissions can be very low [5]. Thus, we can form backbone(s) to maintain the connectivity without sacrificing the reliability in wireless communications. Virtual backbones and (minimum) connected dominating sets (CDS/MCDS) are used interchangeably throughout this paper.

We define the network lifetime as the shortest lifetime of the sensors. Although a single backbone reduces the overall energy consumption of the network, it has no effects on the network lifetime. It is necessary to rotate backbones in order to actually prolong the network lifetime. An intuitive solution is to construct multiple disjoint CDSs and let them work alternatively. It is formulated as a *connected domatic partition* (CDP) problem in [6]. However, this approach is restricted for two reasons. First, some of the sensors have not been added into any backbone. These nodes have never worked at all. Their energy is literally wasted. If we can incorporate all the sensors in backbone rotation, the results should be better. Another problem is that if some backbone nodes, having less initial energy, are forced to be selected to form multiple independent backbones, the entire backbone's lifetime is reduced, as well as the network lifetime.

Considering these observations, we propose *virtual backbone scheduling* (VBS). Similar to CDP, VBS schedules multiple backbones to work alternatively. In VBS, however, backbones can be overlapped. Additionally, VBS employs *heterogeneous scheduling*, where non-backbone sensors shut down their radios to save energy. We provide an example in Fig. 1 to illustrate the benefit of this scheduling model. The left figure shows a network of five sensors and one sink, where the stack beside each node represents its energy. Assuming that sensors consume 1 unit of energy per unit of time, then each sensor can work 3 units of time. We see that there is only one independent CDS formed by $\{sink, 0, 1\}$, $\{sink, 0, 3\}$, or $\{sink, 1, 3\}$. The network lifetime, using CDP based scheduling, is 3 units of time. Whereas in VBS, we schedule backbone $\{sink, 0, 1\}$ work for 1, $\{sink, 0, 3\}$ work for 1, and $\{sink, 1, 3\}$ work for 2 units of time, which yields 4 units of lifetime. This corresponds to a 33% lifetime increase! This simple example demonstrates that VBS partitions nodes at a finer-grained level of granularity, therefore it is able to fully utilize the battery power of the sensors in the network.

Our contributions in this paper are as follows:

- We propose a new scheduling method called VBS. We formulate the MLBS problem to find the optimal schedule in VBS.
- We present two centralized approximation algorithms and a decentralized implementation of VBS. The path stretch of several backbone construction algorithms is analyzed.
- We conduct extensive simulations to verify the performance of VBS.

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

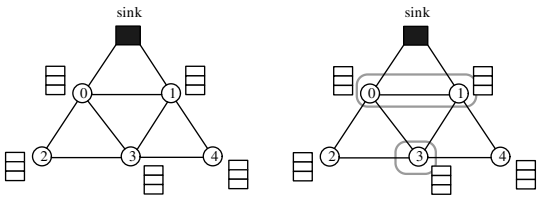


Fig. 1. A simple network consisting of five sensors and a sink, where each sensor has 3 units of energy. This graph only has one independent CDS, which is formed by $\{sink, 0, 1\}$, $\{sink, 0, 3\}$, or $\{sink, 1, 3\}$. VBS can archive a longer lifetime by scheduling overlapped backbones.

II. Related work

A survey on power saving techniques for multi-hop wireless networks is given in [7]. Sleep scheduling can be divided into synchronous [1] and asynchronous [8]. It has been proven in [9] that minimum end-to-end delay sleep scheduling is, in general, NP-hard. The energy-delay trade-off in tree shaped WSNs has been studied in [4]. In [3], five wakeup patterns used in previous work [2], [10], are summarized, and a multi-parent scheduling is proposed to reduce the forward and backward delays in data gathering WSNs. All of the above methods above do not use heterogeneous scheduling.

The connected dominating set is a widely used backbone construction method for wireless networks [11]. The minimum connected dominating set problem is NP-hard in Unit Disk Graph (UDG) [12]. Wu and Li's marking process (MP) [13], Wu and Dai's self-pruning rule [14], rules 1 & 2 [13], rule K [15], and the extended coverage condition [16] are representative distributed MCDS construction algorithms. In [17], several distributed algorithms are proposed to construct K-vertex connected K-coverage CDS (KCDS). Connected domatic partition (CDP) is a partition $\bigcup P_i$ of the nodes of a graph into disjoint sets, where each set P_i of the partition is a CDS. A recent work [6] presents a CDP based backbone rotation scheme.

III. Network model and problem definition

We consider static WSNs. Sensors are randomly placed in the field. There is only one sink in the network, which is always working and has an infinite power supply. The links between sensors are undirected. VBS works with duty-cycling. We define $T(T \geq 1)$ continuous cycles as a *round*. VBS rotates backbones at the end of each round. A node's *lifetime* is the time span from when it starts working to when its energy is depleted. Network lifetime is the minimum lifetime of all the sensors in the network.

A schedule in VBS is a set of backbones that working sequentially. It is represented by a set of tuples $\{\langle B_1, T_1\rangle, \cdots, \langle B_p, T_p\rangle\}$, where T_i is the working time of backbone B_i. The goal is to find a schedule with maximum network lifetime. The definition of MLBS is given as follows:

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

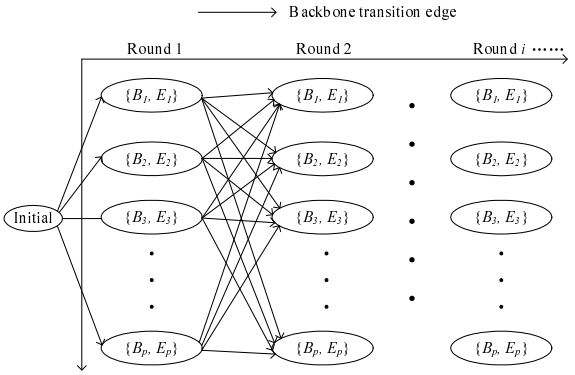


Fig. 2. The illustration of a STG.

We refer to the above problem as the K-MLBS problem. The MLBS problem is to find a schedule of maximum K. We show in Theorem 1 that K-MLBS is NP-complete. Thus, MLBS is NP-hard.

Theorem 1: K-MLBS is NP-complete.

Proof: Due to lack of space, all the proofs of lemmas and theorems are omitted. Please refer to [18] for details. ■

IV. Solving the MLBS problem

Because MLBS is an NP-hard problem, we present three approximation algorithms in this section.

A. A near optimal approximation scheme

We propose a new concept called, *schedule transition graph* (STG), to model the MLBS problem. Fig. 2 presents a STG. The horizontal direction is for time, counted in rounds. The vertical axis is for backbones. Each state contains a backbone and the corresponding *energy levels* (defined later). An initial state connects with all the states in the first round as a starting point. Uni-directed backbone transition edges connect states in one round to those in the next round. The nodes in the backbone of the starting states consume fixed amounts of energy after each transition. No transition is allowed after any sensor is out of energy. A path in STG corresponds to a schedule in the network. The MLBS problem of a network is thus, to find the longest path in its corresponding STG.

1) The time span of STG: The maximum number of rounds in a STG is derived by dividing the sum of the initial energy of all nodes by the minimum energy consumed in each round. Backbone nodes consume a fixed amount of energy ϵ in each round. Because any backbone has more nodes than MCDS, suppose that the size of the MCDS is n, then the minimum energy consumption in each round is at least $n \times \varepsilon$. Denote E as the sum of the initial energy of all nodes. The maximum round number c is given by Eq. 1. Assuming that the initial energy of each sensor is a constant, E is in $O(|V|)$, and n is also in $O(|V|)$, so the round number is in $O(1)$.

$$c = \frac{E}{n \times \varepsilon} \tag{1}$$

Algorithm 1 STG based VBS

```
1: CUR_R ← 1      /*Current round*/
2: repeat
3:    CUR_R ← CUR_R + 1
4:    for Each state in CUR_R do
5:       Calculate the energy levels of current round
6:       Apply min() function and prune its return values
7:       Select the energy level with the maximum minimum energy
8:       if Still has multiple energy levels then
9:          Select the energy level with the maximum summation
10:      end if
11:   end for
12: until All the energy levels of the states in CUR_R are zero
```

2) Energy level: We define the energy level σ of a network $G(V, E)$ as a tuple $\langle E_1^r, E_2^r, \cdots, E_{|V|}^r \rangle$, where E_1^r is the residual energy of the i_{th} sensor in the network. Two energy levels, σ_1 and σ_2, satisfy $\sigma_1 \preceq \sigma_2$ if, for each $i \in \{1, 2, ..., |V|\}$ and $E_i^{r1} \in \sigma_1, E_i^{r2} \in \sigma_2$, there is $E_i^{r1} \leq E_i^{r2}$. $\sigma_1 < \sigma_2$ if $\sigma_1 \preceq \sigma_2$, and there is at least one i such that $E_i^{r1} < E_i^{r2}$. An energy level is *zero* if at least one element is zero. Zero energy levels are less than any non-zero energy level, and indicate the end of the network lifetime.

3) Enumerating backbones: We enumerate a polynomial number (in $|V|$) of backbones for a near optimal result. According to the concept of energy level, we can prune some backbones. If two backbones B_1 and B_2 satisfy $B_1 \subset B_2$, B_2 should not be included in the STG, because any possible transition directed to B_2 will not yield a longer lifetime than those directed to B_1.

4) The algorithm: The pseudo code is listed in Algorithm 1. The search starts from the initial state. After a backbone transition, the state's energy level is computed. Each state keeps the largest energy levels. A path terminates when its associated energy level is zero. The longest path is found when all paths are terminated. We define a function $min()$ to prune *invalid* redundant energy levels for each state:

Definition 2 ($min()$ function): $min(S{=}\{\sigma_1, \sigma_2, ..., \sigma_n\}) = \{\sigma | \sigma \in S$ and there is at least one element $\sigma' \in S$ such that $\sigma \preceq \sigma'\}$

The energy levels produced by this function are invalid, because strictly lower energy levels cannot produce a longer lifetime. $min()$ function can prune some redundant energy levels for each state. If there are still multiple energy levels left for a state, we only keep the one that has the maximum summation of all its elements.

5) Complexity: Suppose that λ backbones are constructed in the enumeration process, 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.

B. Another centralized approximation scheme

1) Virtual Scheduling Graph (VSG): We define a *virtual node* (VN) of a sensor as a node that contains ε energy, and we call the original node the *ancestor*. ε is the energy consumed in each round by a backbone node. An ancestor of E_r energy is divided into $\lceil \frac{E_r}{\varepsilon} \rceil$ VNs. The VNs of the same ancestor form a *virtual group*. VNs in the same virtual group are indexed, as shown in Fig. 3. Two virtual groups are neighbors if their ancestors are neighbors in the original graph. A VN is *isolated* if it does not connect with any VNs in other virtual groups.

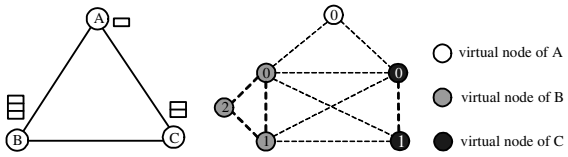


Fig. 3. Three nodes of different amounts of residual energy (left) and the corresponding VSG (right). Each VN is indexed. VNs are connected with increasing index order. Note that VN 2 of B is isolated because it has more energy and cannot fully connect with the VNs of A or C.

The virtual scheduling graph (VSG) $G_s(V', E')$ of a graph $G(V, E)$ using the following *VSG rules*:

- V' is all the VNs of V.
- Nodes of two neighboring virtual groups are connected with the increasing index order until one of the two group's VNs are all connected.
- VNs' priorities are using tuples of (degree, ID), where ID is the original ID plus the VN's ID.

Fig. 3 gives a VSG of a network of 3 nodes. We can view VSG as a collection of scattered snapshots of the original graph in each round. The reason to use a VSG is that it will be biased towards nodes with more energy. Nodes with more energy will produce "isolated" VNs. In this way, the CDS selection scheme is forced to pick nodes with more energy. This approach also prevents selecting nodes with scarce energy, but with a high degree.

2) Pseudo minimum connected dominating set (PMCDS): The definition of PMCDS is as follows. Theorem 2 states that a PMCDS of the VSG corresponds to a CDS of the original graph:

Definition 3 (PMCDS): A PMCDS of a VSG is a CDS that does not contain multiple VNs of the same ancestor.

Lemma 1: PMCDS can be computed in polynomial time using any MCDS approximation algorithm.

Theorem 2: Given a graph $G(V, E)$ and its corresponding VSG $G_s(V', E')$, suppose $\bigcup U_i'$ is a PMCDS of G', and $\bigcup U_i$ is the corresponding ancestors of $\bigcup U_i'$ in G, then $\bigcup U_i$ is a CDS of G.

3) The algorithm: We transform the MLBS problem of a WSN into a MCDS problem in its VSG. The pseudo code is shown in Algorithm 2. Algorithm 2 iteratively constructs PMCDSs of the VSG. Those backbone nodes are then removed from the VSG. We apply the VSG rule to preserve the correspondence between the original graph and the modified VSG after each iteration, so that Theorem 2 is still valid in the resultant VSG. When all of the VNs of any ancestor are removed, i.e. the energy of the ancestor is depleted, the algorithm ends.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

Algorithm 2 VSG based VBS

```
S ← {}
Construct the VSG G_s(V', E') of G(V, E)
repeat
    Construct a CDS C in VSG using rules 1 & 2 or rule K
    Construct the PMCDS C' from C
    Remove the highest indexed VNs of the ancestors whose VNs is in C' from G_s(V', E')
    Apply the VSG rules to eliminate edges between virtual groups of different sizes.
    Find the corresponding CDS C_i of C' in G
    Append ⟨C_i, T_i⟩ to S
until Any ancestor's VNs are all eliminated from G_s(V', E')
return S
```

Algorithm 3 Iterative local replacement

```
loop
    At the beginning of each round
    Compute the switching probability P_switch
    if Decide to switch then
        Collect or update the h-hop information
        Recalculate backbone using rules 1 & 2, or rule K
        Notify replacement nodes
    end if
end loop
```

4) Complexity: Recall that the maximum round number is c, which is determined by Eq. 1, and we need to construct at most as many backbones. So the time complexity of the VSG-based scheduling is $O(c|V|)$, where c is given in Eq. 1.

C. An iterative local replacement scheme

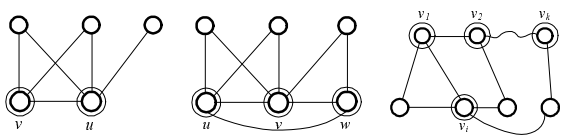
We employ a control-based scheme in *iterative local replacement* (ILR). A switching probability P_{switch} is computed by each backbone node at the end of the current round. A backbone node, that decides to switch, broadcasts a message to "hold" h-hop neighbors to stay awake to complete the replacement. It then notifies its replacement nodes after computation is completed. The switching probability is shown in the above equation. E_r is the residual energy, and E_c is the energy consumed since its last switch to backbone node.

$$P_{switch} = \frac{E_c}{E_r} \tag{2}$$

The pseudo-code of ILR is shown in Algorithm 3. ILR is executed at each backbone node at each round. In order to find its replacement, a backbone node uses distributed algorithms, such as rules 1 & 2 or rule K. The parameter h trades off between overhead and efficiency.

V. Discussion

The induced graph from CDS prolongs the shortest-paths between nodes. We define the *stretch factor* of the path connecting u and v as the ratio of its hop-count in the induced graph to that of the original graph. Rules 1 & 2 are extensions to the marking process (MP) [13]. 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. Rules 1 & 2 apply to the induced graph G' of MP. Rule 1 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 a higher priority. Rule 1 has a maximum stretch factor of 2. Because v and its neighbors are covered by u, the path lengths between v and its neighbors increase by one. Rule 2 states that, for marked nodes u and w are two neighbors of a marked node v, if all v's neighbors are covered by u and w, and v has the lowest priority among the 3 nodes, then v can be unmarked. It is clear that its stretch factor is 3. Rule K 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 its neighbors are jointly covered by connected subset $\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_k\}$. Rule K's stretch factor is k, because two directly connected nodes in the original graph may now be connected by a chain of k-1 nodes in the worst case, resulting in a k-hop path.



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

Fig. 4. The illustrations of three CDS construction rules.

VI. Simulation evaluation

We are interested in network lifetime and worst-case delay. We compare VBS with the CDP method proposed in [6], and the all-working scheduling, where all the sensors in the network work in duty-cycling. Sensors are randomly placed in a square area. The sink is placed at the center of the area. All nodes have the same transmission range. The number of nodes is varied to model different network densities and scale.

A. Network lifetime

We assume that 1 unit of energy is consumed in each round. Two configurations are considered: identical initial energy and imbalanced initial energy. We first consider the effect of network density. Sensors are deployed in a 500×500 area. Transmission range is fixed to 250. The number of nodes in the network ranges from 10 to 100 with a step of 10.

Fig. 5 gives the lifetime while nodes have 100 units of initial energy. The line labeled "original" represents the results when all nodes work. Rules 1 & 2, and rule K are used to construct backbones. The STG based approach has the best values; the VSG based algorithm is slightly worse. The poor performance of ILR is because it uses only local information. We also notice that rule K achieves slightly longer lifetimes. This is because it is able to construct smaller sized backbones, and the resultant backbones tend to be disjoint; thus, the energy consumption is more balanced and the network lifetime is longer. Fig. 6 presents the lifetime of the networks with uniform distributed initial energy. Each node is assigned an initial energy drawn uniformly from $[50, 100]$. Because the lifetime is determined by the node with the minimum energy, the achieved lifetime

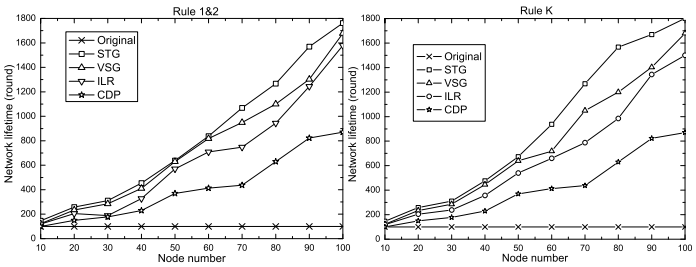


Fig. 5. Lifetime of networks with identical initial energy.

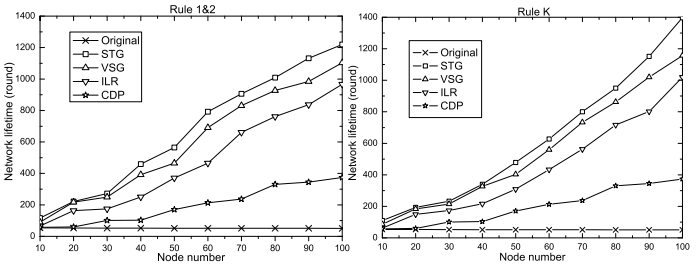


Fig. 6. Lifetime of networks with uniform distributed initial energy.

when all nodes work is nearly halved, as shown in the line labeled "original". Our proposed schemes still achieve much longer lifetimes.

B. Worst case delay

WSN applications are sensitive to the worst case delay. The worst case delay is measured from the farthest sensor to the sink. Suppose that the duty cycle is $1s$, and the per-hop delay between two duty-cycled nodes is $100ms$, we give the worst case delays in dense networks and sparse networks in Fig. 7. In order to control the density, the transmission range of nodes is adjusted until the given degree is achieved. The average degree of the dense network is 16; the sparse one is 8. We see that the stretch obtained in this set of simulations is within 20%. Considering the significant lifetime extension, this is acceptable. The traffic from sensors to the sink is the most common in WSNs. Additionally, because non-backbone nodes shut down the radio, the delay from the sink to non-backbone is indefinite. This flaw can be remedied by applying a lower wake-up frequency on non-backbone sensors.

VII. Conclusion and Future Work

In this paper, we presented a new scheduling method for WSNs called VBS. VBS combines virtual backbone and sleep scheduling, which achieves longer lifetimes for WSNs over existing methods. We defined the MLBS problem for WSNs to find the optimal schedule for VBS. Because the MLBS problem is NP-hard, two centralized approximation algorithms and a decentralized implementation are presented. We then analyzed the path stretches of several backbone construction heuristics. Simulation results prove that VBS significantly prolongs network lifetime under extensive conditions. In our future work, we will develop a more efficient distributed implementation of VBS.

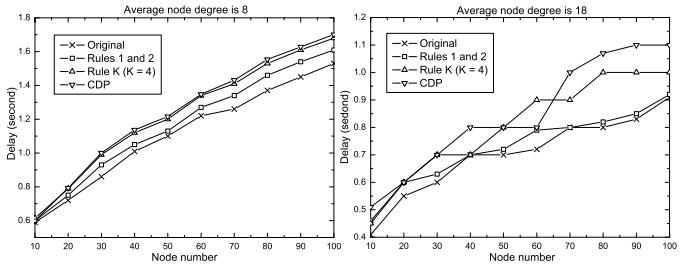


Fig. 7. The worst case delay in networks with different average degrees.

Acknowledgement

This work is supported in part by NSF grants CNS 0422762, CNS 0626240, CCF 0830289, and CNS 0948184.

References

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

[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*, pp. 20–27, 2005.

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

[4] 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.*, vol. 17, no. 2, pp. 570–581, 2009.

[5] J. W. Hui and D. E. Culler, "IP is dead, long live IP for wireless sensor networks," in *Proc. of ACM SenSys'08*, pp. 15–28, 2008.

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

[7] M. J. Miller and N. H. Vaidya, "Power save mechanisms for multi-hop wireless networks," in *Proc. of BROADNETS' 04*, pp. 104–114, 2004.

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

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

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

[11] K. Alzoubi, P.-J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *Proc. of HICSS '02*, p. 297, 2002.

[12] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1-3, pp. 165–177, 1990.

[13] J. Wu and H. L. 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.*, pp. 7–14, 1999.

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

[15] J. Wu, "Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links," *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 866–881, 2002.

[16] 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.*, pp. 908–920, IEEE Press, 2004.

[17] F. Dai and J. Wu, "On constructing k-connected k-dominating set in wireless ad hoc and sensor networks," *J. Paral. Distrib. Comput.*, vol. 66, no. 7, pp. 947–958, 2006.

[18] Y. Zhao, J. Wu, F. Li, and S. Lu, "Maximum lifetime scheduling for wireless sensor networks using virtual backbones," *Dept. of Comp. and Info. Sci. Technical Report, Temple University, available at http://www.cis.temple.edu/∼wu/tr-ss.pdf*, 2009.