Minimum Power Configuration for Wireless Communication in Sensor Networks

GUOLIANG XING
City University of Hong Kong
CHENYANG LU
Washington University
YING ZHANG and QINGFENG HUANG
Palo Alto Research Center (PARC), Inc.
and
ROBERT PLESS
Washington University in St. Louis

This article proposes the *minimum power configuration* (MPC) approach to power management in wireless sensor networks. In contrast to earlier research that treats different radio states (i.e., transmission/reception/idle) in isolation, MPC integrates them in a joint optimization problem that depends on both the set of active nodes and the transmission power. We propose four approximation algorithms with provable performance bounds and two practical routing protocols. Simulations based on realistic radio models show that the MPC approach can conserve more energy than existing minimum power routing and topology control protocols. Furthermore, it can flexibly adapt to network workload and radio platforms.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Routing and layout*

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Sensor networks, minimum power configuration, ad hoc networks, energy efficiency, wireless communications

Part of this article was published in *Proceedings of the MobiHoc* [Xing et al. 2005]. This work was funded in part by the National Science Foundation under ITR Grant CCR-0325529 and NeTS-NOSS Grant CNS-0627126, and by the DARPA under Contract F33615-01-C-1904.
Authors' addresses: G. Xing, Department of Computer Science, City University of Hong Kong, Hong Kong, email: glxing@cityu.edu.hk; C. Lu and R. Pless, Department of Computer Science and Engineering, Washington University, St. Louis, MO 63130; email: {lu,pless}@cse.wustl.edu; Y. Zhang and Q. Huang, Palo Alto Research Center (PARC), Inc., 3333 Coyote Hill Road, Palo Alto, CA 94304; email: {yzhang, ghuang}@parc.com.

ACM Reference Format:
Xing, G., Lu, C., Zhang, Y., Huang, Q., and Pless, R. 2007. Minimum power configuration for wireless communication in sensor networks. ACM Trans. Sens. Netw. 3, 2, Article 11 (June 2007), 33 pages. DOI = 10.1145/1240226.1240231 http://doi.acm.org/10.1145/1240226.1240231

1. INTRODUCTION

Many wireless sensor networks (WSNs) must aggressively conserve energy in order to operate for extensive periods without wired power sources. Since wireless communication often dominates the energy dissipation in a WSN, several promising approaches have been proposed to achieve power-efficient multihop communication in ad hoc networks. *Topology control* protocols [Rodoplu and Meng 1999; Ramanathan and Hain 2000; Narayanaswamy et al. 2002; Kawadia and Kumar 2003; Li et al. 2001, 2003; Alzoubi et al. 2003] aim to reduce the overall transmission power of a network by adjusting the transmission range at each node, while still preserving necessary network properties (e.g., connectivity). *Power-aware routing* protocols [Singh et al. 1998; Doshi et al. 2002; Doshi and Brown 2002; Chang and Tassiulas 2000; Sankar and Liu 2004] choose appropriate transmission ranges and routes to conserve energy used for multihop packet transmission. Both topology control and power-aware routing focus on reducing power consumption when the radio interface is actively transmitting/receiving packets. Such approaches alone are often insufficient, however, because radio interfaces (e.g., the CC1000 radio on Mica2 motes [Crossbow 2003] and WLAN cards [Chen et al. 2001]) also consume nonnegligible power, even if they are running in idle state. *Sleep management* [Chen et al. 2001; Xing et al. 2005; van Dam and Langendoen 2003; Zheng and Kravets 2003; Chipara et al. 2000; IEEE 1999; Ye et al. 2002; Polastre et al. 2004] has been proposed to reduce the energy wasted in an idle state by turning off radios when not in use.

Clearly, a WSN needs to reduce the energy consumed in each of the radio's power states (i.e., transmission, reception, and idle) in order to minimize its energy consumption. This requires a WSN to effectively apply all of the aforementioned approaches. As we will show in this article, however, the correlations between the different approaches are dependent on the network load and hence cannot be combined in a straightforward fashion. For example, when network workload is low, the energy consumption of a WSN is dominated by the idle state. In such a case, scheduling nodes to sleep saves the most energy. It is therefore more energy efficient for active nodes to use long communication ranges, since this will require fewer nodes to remain awake in order to relay packets. Conversely, short radio ranges may be preferable when the network workload is high, as the radio tends to spend more time in the transmission and reception states. In this article, we propose a novel approach, called *minimum power configuration (MPC)*, that minimizes the aggregate energy consumption in all power states. In sharp contrast to earlier research that treated topology control, power-aware routing, and sleep management in isolation, MPC provides a unified approach that integrates them as a joint optimization problem in which the power configuration of a network consists of a set of active nodes and the transmission power of those nodes.

This article makes the following key contributions. First, we show through analysis that the minimum power configuration of a network is inherently dependent on the data rates of sources in the network (Section 3). Second, we provide a new problem formulation that models the energy conservation in a WSN as a joint optimization problem that considers the overall energy consumption from all power states of the radio according to the network workload (Section 4). Third, we show that the minimum power configuration problem is NP-hard, and then propose four approximation algorithms with provable performance bounds compared to the optimal solution (Section 5). Fourth, we propose two distributed protocols minimum power configuration protocol (MPCP) and minimum active subnet protocol (MASP) (Section 6). The key advantage of MPCP is that it can flexibly adapt to a wide range of radio platforms by taking into consideration the power characteristics of the radio while MASP is a more efficient protocol that is only suitable for radios with high idle power. Finally, our analysis is validated by detailed simulations based on a realistic radio model [Zuniga and Krishnamachari 2004] (e.g., asymmetric and probabilistic radio links) of the Mica2 motes.

2. RELATED WORK

Numerous solutions have been proposed for conserving energy in wireless ad hoc (sensor) networks in literature. These protocols can be classified into roughly three approaches, namely, topology control, power-aware routing, and sleep management. We summarize the limitations of each after providing a brief overview of the existing works of each approach.

—*Topology control*: Topology control preserves the desirable properties of a wireless network (e.g., K-connectivity) through reduced transmission power. A comprehensive survey on existing topology control schemes can be found in Stankovic et al. [2003]. We review several representative works here. In the scheme proposed in Rodoplu and Meng [1999], a node chooses to relay through other nodes only when less power is used. The network can be shown to be strongly connected if every node has links to only those nodes that are within its “enclosure,” as defined by a relay region. Ramanathan and Hain [2000] proposed two centralized algorithms to minimize the maximal power used per node while maintaining the (bi)connectivity of the network. Two distributed heuristics were also proposed for mobile networks in Ramanathan and Hain [2000], although they may not necessarily preserve network connectivity. Two algorithms are proposed in Kawadia and Kumar [2003] and Narayanaswamy et al. [2002] to maintain network connectivity using minimal transmission power. CBTC [Li et al. 2001] preserves network connectivity using the minimum power that can reach some node in every cone of size smaller than $5\pi/6$. A local topology called localized Delaunay triangulation is shown to have a constant stretch factor with respect to the original network [Alzoubi et al. 2003]. Li et al. proposed a MST-based topology control scheme which preserves the network connectivity and has bounded node degrees [Li et al. 2003]. The problem of maximizing network lifetime under topology control is studied in Calinescu et al. [2003].

—*Power-aware routing*: Power-aware routing minimizes the total transmission energy consumed by a packet on its network route. Singh et al. proposed five power-aware routing metrics to reduce energy consumption and extend system lifetime [Singh et al. 1998]. The implementation of a minimum energy routing protocol based on DSR was discussed in Doshi et al. [2002] and Doshi and Brown [2002]. An online power-aware routing scheme is proposed to optimize system lifetime in Li et al. [2001]. Chang and Tassiulas studied the problem of maximizing the lifetime of a network with known data rates [Chang and Tassiulas 2000]. Chang and Tassiulas [2000] formulated the problem of choosing routes and transmission power of each node to maximize the system lifetime as a linear programming problem and discussed two centralized algorithms. Sankar and Liu [2004] formulated maximum lifetime routing as a maximum concurrent flow problem and proposed a distributed algorithm. More recently, Dong et al. [2005] studied the problem of minimum transmission energy routing in the presence of unreliable communication links.

—*Sleep management*: Recent studies showed that significant energy savings can be achieved by turning radios off when not in use. There are two basic approaches, namely, scheduling- and backbone-based sleep management. In the scheduling-based approach, nodes turn on their radios only in scheduled slots. The active slots of different nodes can be synchronous [IEEE 1999; Ye et al. 2002], or asynchronous [Zheng et al. 2003; Polastre et al. 2004; Ergen 2002; Hohlt et al. 2004]. In addition, several adaptive sleep schemes dynamically adjust the schedules based on traffic activities [Ye et al. 2002; van Dam and Langendoen 2003; Zheng and Kravets 2003; Chipara et al. 2000]. Backbone-based sleep management can improve network performance by maintaining a backbone composed of a small number of active nodes, while scheduling the other nodes to operate in low duty cycles to conserve energy [Chen et al. 2001; Xu et al. 2000, 2001; Xing et al. 2005].

None of the aforementioned three approaches optimize the energy consumption of all radio states. Topology control and power-aware routing reduce the transmission energy of wireless nodes and do not consider the idle energy. Sleep management can reduce the idle energy by scheduling idle nodes to sleep, but does not optimize the transmission energy. In summary, existing approaches suffer from the following two major drawbacks. First, the existing approaches are only suitable for limited network conditions, as they only minimize the energy consumption under partial radio states. Power-aware routing and topology control are effective only when the network workload is so high that the transmission energy dominates the overall energy consumption of the network. Similarly, sleep management is effective only in lightly loaded networks where idle energy dominates the overall energy consumption. Second, the existing schemes may yield very different performance characteristics among different radio platforms. For example, although sleep management may considerably reduce the energy consumption when the idle power of the radios is high

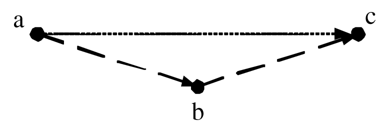


Fig. 1. Two communication paths from a to c: a → c or a → b → c.

relative to the communication power, it is less effective for radios that have low idle power.

In this article, we show through analysis that the network configuration that minimizes the total radio energy depends on workload as well as radio characteristics. Recently, Dong [2005] independently pointed out that there exist workload-dependent tradeoffs between topology control and power-aware routing in order to minimize both idle and transmission energy consumption. However, the problem of minimizing the total energy is left unaddressed. In this article, we formalize the problem and propose several approximate algorithms, as well as practical distributed protocols. To the best of our knowledge, this work is the first that aims to minimize the total energy consumption of all radio states in a network.

3. AN ILLUSTRATING EXAMPLE

In this section, we illustrate the basic idea of our approach with a simple example. We focus on the energy consumption of radios, since they tend to be the major source of power dissipation in wireless networks. We will show that when the total energy from each of the different radio states is considered, the optimal network configuration depends on the radio characteristics and data rates of the network. A wireless radio can work in one of the following states: transmitting, receiving, idle, and sleeping. The corresponding power consumptions are represented by $P_{tx}(d)$, P_{rx}, P_{id}, and P_s, respectively, where d is the Euclidean distance of the transmission.

As shown in Figure 1, a, b, and c are three nodes located in 2D space. Here, a needs to send data to c at the rate of R bps. The bandwidth of all nodes is B bps. There are two network configurations to accomplish the communication between a and c: (1) a communicates with c directly using transmission range $|ac|$ while b remains sleeping; or (2) a communicates with b using transmission range $|ab|$ and b relays the data from a to c using transmission range $|bc|$. Minimizing the total energy of all nodes in the network is equivalent to minimizing the average power consumption of all radio states. We denote the average power consumption under the two configurations as P_1 and P_2, respectively. P_1 and P_2 can be computed as follows:

$$P_1 = \frac{R}{B} \cdot P_{tx}(|ac|) + \frac{R}{B} \cdot P_{rx} + 2\left(1 - \frac{R}{B}\right) \cdot P_{id} + P_s$$

$$P_2 = \frac{R}{B} \cdot (P_{tx}(|ab|) + P_{tx}(|bc|)) + \frac{2R}{B} \cdot P_{rx} + \left(3 - \frac{4R}{B}\right) \cdot P_{id}$$

Each term in P_1 or P_2 is the product of power consumption in a radio state and the fraction of time that the radio operates in this state. For example, in the

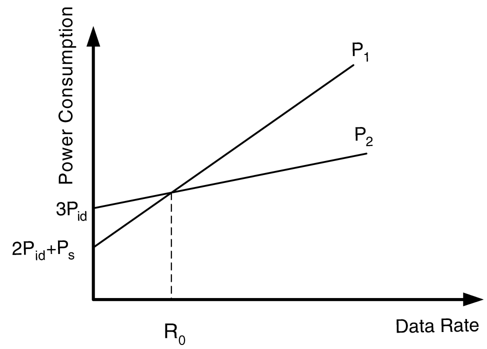


Fig. 2. Average power consumption vs. data rate.

first term of P_2, $P_{tx}(|ab|) + P_{tx}(|bc|)$ is the transmission power of nodes a and b, and $\frac{R}{B}$ is the fraction of time nodes a and b operate in transmission state. Similarly, the second term of P_2 represents the contribution of the reception power of nodes b and c. In the third term of P_2, P_{id} is the idle power, and $3 - \frac{4R}{B}$ is the sum of the fractions of time when nodes a, b, and c stay in the idle state. Specifically, node a is idle $1 - \frac{R}{B}$ of the time because it becomes idle when not transmitting to b, node b is idle $1 - \frac{2R}{B}$ of the time because it becomes idle only when neither transmitting to c nor receiving from a, and node c is idle $1 - \frac{R}{B}$ of the time because it becomes idle when not receiving from b.

For the given radio parameters and node locations, all symbols except R are constant in the expressions of P_1 and P_2. We plot P_1 and P_2 in Figure 2 under a possible setting of radio parameters and node locations. We can see that $P_1 > P_2$ when the data rate exceeds a threshold R_0 given by

$$R_0 = \frac{P_{id} - P_s}{P_{tx}(|ac|) - P_{tx}(|bc|) - P_{tx}(|ab|) + 2P_{id} - P_{rx}}. \tag{1}$$

To get a concrete estimation on R_0, we now apply the parameters of the CC1000 radio on Mica2 motes [Crossbow 2003] to Eq. (1). For a $433MHz$ CC1000 radio, the bandwidth is 38.4Kbps. There are a total of 31 transmission power levels, each of which leads to a different transmission range.[1] Suppose $P_{tx}(|ac|)$ is equal to the maximum transmission power 80.1 mW. $P_{tx}(|ab|)$ and $P_{tx}(|bc|)$ are equal to the medium transmission power 24.6 mW. P_{id}, P_{rx}, and P_s are 24 mW, 24 mW, and 6 μW, respectively. Using this information, it can be calculated that relaying through node b is more power efficient when the data rate is above 16.8Kbps.

This example leads to the following observations on the power-efficient network configuration: (1) When network workload is low, the energy consumption of a network is dominated by the idle state of the radio. In such a case, scheduling nodes to sleep saves the most energy. It is therefore wise to use long communication range between any two nodes in order to allow any nodes that would otherwise be used as relays to sleep. (2) When network workload is high, the transmission energy dominates the total energy consumption of a network. Since transmission power increases quickly with distance, using

[1]The actual transmission range of a radio also depends on environment and antennas.

shorter communication ranges that are relayed through multiple nodes saves more energy.

4. PROBLEM DEFINITION

We define our problem formally in this section. We first define several simple concepts. A node can either be *active* or *sleeping*. For any given time instance, an active node works in one of the following states: *transmitting*, *receiving*, or *idle*. The total energy consumption of an active node is equal to the sum of energy consumption in all states. The sleeping power consumption is orders of magnitude lower than active power consumption [Crossbow 2003; Chen et al. 2001]. In this article, we only consider the total active energy consumption in a network. We define the following notation.

(1) The maximal and minimal transmission power of each node is denoted by P_{tx}^{max} and P_{tx}^{min}, respectively. $P_{tx}(u, v)$ is the minimum power needed for successful transmission from node u to node v, $P_{tx}^{min} \le P_{tx}(u, v) \le P_{tx}^{max}$.
(2) $G(V, E)$ represents a wireless network. V includes all nodes in the network and E is defined as $E = \{(u, v) | (u, v \in V) \wedge (P_{tx}(u, v) \le P_{tx}^{max})\}$.
(3) P_{rx} and P_{id} represent the power consumption of a node in receiving and idle state, respectively.
(4) $S = \{s_i\}$ and $T = \{t_j\}$ represent a set of source and sink nodes, respectively. $I = \{(s_i, t_j, r_{i,j}) \mid s_i \in S,\ t_j \in T\}$ represents a set of traffic demands, where source s_i sends data to sink t_j at rate $r_{i,j}$.

In many sensor network applications such as periodic data collection [Mainwaring et al. 2002; Xu et al. 2004], a source is aware of its data rate. Alternatively, a source may estimate its average data rate online. We assume that the total workload in the network is lower than the network capacity, which is in turn much lower than node bandwidth in multihop wireless networks due to network contention and interference. We note that this assumption holds in many sensor network applications with low data rates. For instance, in the WSN deployed at Great Duck Island for habitat monitoring [Szewczyk et al. 2004], each mote only sends its sensor data to the base station every 20 minutes. Many other representative applications (e.g., precision agriculture and cargo tracking) also have low data rate.

The minimum power configuration (MPC) problem can be stated as follows. Given a network and a set of traffic demands, find a subnet that satisfies the traffic demands with minimum energy consumption. We note that minimizing the total energy consumption of a network is equivalent to minimizing the average power consumption of all nodes. We first consider the average power consumption of a node, assuming that the data path $f(s_i, t_j)$ from source s_i to sink t_j is known. To simplify the formulation, we introduce a virtual source node s_* and virtual sink node t_* to the network. Moreover, s_* sends data to each source s_i at the rate of $r_{i,j}$. Each sink t_j sends data to t_* at a rate of $r_{i,j}$. Note that the additional power consumption due to the introduction of s_* and t_* is constant for a given set of traffic demands. Now, the average power consumption $P(u)$ of any active node u (excluding s_* and t_*) can be computed as the weighted average of power consumption in transmitting, receiving, and idle states.

$$P(u) = \left(1 - 2\sum_{(u,v)\in f(s_i,t_j)} r_{i,j}\right)\cdot P_{id} + \sum_{(u,v)\in f(s_i,t_j)} r_{i,j}\cdot(P_{tx}(u,v) + P_{rx})$$
$$= P_{id} + \sum_{(u,v)\in f(s_i,t_j)} r_{i,j}\cdot(P_{tx}(u,v) + P_{rx} - 2P_{id}),$$

where $(u, v) \in f(s_i, t_j)$ represents that there exists a node v such that edge (u, v) is on the path $f(s_i, t_j)$. Based on the average power consumption of a node defined by the previous equation, the MPC problem can be defined as follows.

Definition 4.1 (*MPC Problem*). Given a network $G(V, E)$ and a set of traffic demands I, find a subgraph $G'(V', E')$ $(V' \subseteq V, E' \subseteq E)$ and a path $f(s_i, t_j)$ within G' for each traffic demand $(s_i, t_j, r_{i,j}) \in I$ such that the average power consumption $P(G')$ is minimal, where

$$P(G') = \sum_{u\in V'} P(u) = |V'|z + \sum_{u\in V'}\sum_{(u,v)\in f(s_i,t_j)} r_{i,j}\cdot C_{u,v} \tag{2}$$

and $C_{u,v}$ and z are defined as follows:

$$C_{u,v} = P_{tx}(u,v) + P_{rx} - 2P_{id} \tag{3}$$
$$z = P_{id} \tag{4}$$

From the preceding formulation, we can see that an edge (u, v) has cost $C_{u,v}$ for each unit of the data flowing through it, and each node has a fixed cost z that is independent of workload. We assume that all the data in the same flow takes the same path, that is, a flow is not splittable. Under such a consumption, one can show that network path $f(s_i, t_j)$ is the shortest path in graph G' with edge weight $C_{u,v}$. Eq. (2) can then be reformulated as follows:

$$P(G') = |V'|z + \sum_{(s_i,t_j,r_{i,j})\in I} r_{i,j}\cdot P(s_i,t_j), \tag{5}$$

where $P(s_i, t_j)$ represents the shortest path in $G'(V', E')$ with edge weight $C_{u,v}$. According to Eq. (5), the total power cost is equal to the sum of costs along the shortest path of each traffic demand and the total nodal costs.

When $\forall(u, v) \in E,\ P_{tx}(u, v) + P_{rx} = 2P_{id}$, the cost function of the MPC problem becomes $|V'|z$. When there is only one sink t in the network, the problem is equivalent to finding the minimum-weight Steiner tree in $G(V, E)$ with uniform edge weight z to connect the nodes in $S \cup \{t\}$. This special case of the minimum-weight Steiner tree problem is NP-hard [Garey and Johnson 1990]. As a result, a natural reduction from this problem can show that the MPC problem is also NP-hard.

Although polynomial solutions for the general MPC problem are unlikely to exist, the following nontrivial special cases of the MPC problem can be solved optimally in polynomial time.

(1) When $S \cup T = V$, every node in the network is either source or sink and hence needs to remain active. Thus the first term in (2) becomes $|V|z$, which is constant for a given network. In such a case, the solution is equivalent to finding the shortest paths with edge weight $r_{i,j} \cdot C_{i,j}$ connecting all sources to their sinks and hence can be solved in polynomial time.
(2) When $P_{id} = 0$, similar to the first case, the MPC problem can be solved optimally by shortest-path algorithms.

In the problem formulation, we assume that all data sources are known offline. This assumption may not be practical in many sensor network applications where data sources are usually triggered by asynchronous events (e.g., an object passing by) or a query submitted by users. In other words, the data sources in many scenarios arrive in an online fashion. In Section 5, we discuss both offline and online approximate algorithms for the MPC problem.

In our problem definition, the energy consumption of packet retransmissions on lossy communication links is ignored. Recent empirical studies show that lossy communication links are common in real sensor networks [Woo et al. 2003; Zhao and Govindan 2003]. In such a case, the communication quality between two nodes can be quantified by the packet reception ratio (PRR) [Zuniga and Krishnamachari 2004]. In this article, we assume that an automatic repeat request (ARQ) mechanism is used to deal with lossy links. A node with ARQ keeps retransmitting a packet until the packet is successfully acknowledged by the receiver or the preset maximum number of retransmissions is reached. To reflect the additional energy cost caused by retransmissions, the cost function defined in Eq. (2) can be revised as follows. Let $PRR(u, v, P_{tx})$ represent the PRR when u communicates with v using transmission power P_{tx}. Note that $PRR(u, v, P_{tx})$ depends on the quality of both forward and reverse links between u and v when an ARQ is used.[2] The expected transmission power cost when u communicates with v with P_{tx} on the lossy links can be estimated as $P_{tx}/PRR(u, v, P_{tx})$. Hence the most efficient transmission power that should be used by u to communicate with v is determined as follows:

$$P_{tx}(u, v) = arg\min \frac{P_{tx}}{PRR(u, v, P_{tx})}, \quad P_{tx}^{min} \leq P_{tx} \leq P_{tx}^{max} \tag{6}$$

We redefine $P_{tx}(u, v)$ in Eq. (3) of our problem formulation according to (6) when the communication links are lossy.

5. CENTRALIZED APPROXIMATION ALGORITHMS

We investigate approximate algorithms for the general MPC problem in this section. We first focus on the scenario where there is only one sink in the network. Each source s_i ($s_i \in S$) sends data to sink t at a data rate of r_i. We discuss the extension of some of our results to the scenario of multiple sinks in Section 5.3.

[2]Acknowledgment can be transmitted at a relatively high power level to reduce the number of retransmissions.

Input: $G(V, E)$, set $W = S \cup \{t\}$ and traffic demands I
Output: $G^{'}(V', E')$

(1) Create a complete graph M containing all nodes in W as follows. Each edge between two nodes in M is the shortest path between the two nodes in G under the edge cost D. For two sources s_i and s_j, $D_{u,v} = z + \frac{2r_i r_j}{r_i + r_j} C_{u,v},\ (u, v) \in E$. For a source s_i and sink t, $D_{u,v} = z + r_i C_{u,v},\ (u, v) \in E$.
(2) Find a matching of graph M that has at most half the cost of the minimum perfect matching, and has at most half of the number of total nodes.
(3) The nodes and edges of G defining each matched edge of M are added into G'. For each matched edge (s_i, s_j) in M, choose s_i to be the center with probability $r_i/(r_i + r_j)$, otherwise s_j will be the center. Change the data rate of the center as $r_i + r_j$.
(4) Each noncenter node in a matched edge of M is removed from W. Stop if S contains only the sink. Otherwise go to step 1 with the updated W.

Fig. 3. Matching-based algorithm (MBA) for MPC problem.

5.1 Matching-Based Algorithm

When there is only one sink and data flows are not splittable, the MPC problem has the same formulation as the *cost-distance* network design problem [Meyerson et al. 2000]. Those authors proposed a randomized approximation scheme [Meyerson et al. 2000] that has a best-known approximation ratio of $O(\lg k)$, with k being the number of sources. We briefly review the algorithm and propose an optimization that considerably improves the practical performance of the algorithm.

The Meyerson algorithm takes a graph $G(V, E)$ and outputs a subgraph $G'(V', E')$ that contains the paths from all sources to the sink.

The time complexity of the aforementioned algorithm is $O(k^2(m + n \lg n))$, where k, m, and n represent the number of sources, total number of edges, and nodes in G, respectively. As shown in Meyerson et al. [2000], the algorithm terminates after at most $O(\lg k)$ iterations and the expected cost introduced by the newly added edges in each iteration is at most constant times of the cost of the optimal solution. Hence, the approximation ratio of the algorithm must be $O(\lg k)$. We refer to this algorithm as *matching-based approximation (MBA)* in the rest of the article.

We note that an edge of G can lie on the matched edges of M in multiple iterations at step 3 of MBA. However, the fixed cost of each edge z is only counted once in the total cost of the solution (see Eq. (2)). This observation can lead to the following optimization to MBA. After the matching of M is found in step 2, we redefine the cost of each matched edge of G as $D_{u,v} = \frac{2r_i r_j}{r_i + r_j} C_{u,v}$. In other words, the fixed cost of each edge z is removed if the edge is matched. The intuition behind this consideration is that the matchings in following iterations will tend to reuse those edges of G that have been previously matched due to the cost reduction on these edges. Consequently, the total cost of the solution may be reduced by more path sharing. We refer to the MBA with this optimization as MBA-opt. Although MBA-opt does not improve the approximation ratio of MBA, we show in Section 5.5 that it can result in considerable improvement on the practical performance.

Input: $G(V, E)$, source set S, sink t and traffic demands I
Output: $G^{'}(V', E')$

(1) Initialize $G^{'}(V', E')$ to be empty.
(2) **foreach** s_i
  (a) Assign edge weights for $G(V, E)$ according to g_i.
  (b) Find the shortest path connecting s_i to t.
  (c) Add the shortest path found to $G^{'}$.
(3) **end**

Fig. 4. Shortest-path tree heuristic (STH).

Although MBA and MBA-opt have a good performance bound, they suffer from the following drawbacks. First, efficient distributed implementations of them are difficult to realize in large-scale sensor networks. In order to find the matching of the network graph (step 2 of MBA) in a distributed environment, complex coordination between nodes is needed [Wattenhofer and Wattenhofer 2004]. Secondly, MBA and MBA-opt are not applicable to the online scenario in which sources arrive dynamically because finding the matching of the network requires knowledge of all data sources. Finally, MBA and MBA-opt only work for the scenario in which there is a single sink in the network. Because of these drawbacks, we are forced to design other approximate algorithms that are more suitable to distributed and online implementations.

5.2 Shortest-Path Tree Heuristic (STH)

In this section, we discuss an approximation algorithm called the shortest-path tree heuristic (STH). The idea behind this heuristic is to balance the flow-dependent cost $(r_{i,j} \cdot C_{u,v})$ and the fixed nodal cost (z) of a graph using a combined cost metric. For convenience, we define a set of weight functions for edge (u, v).

$$g_i(u, v) = r_i \cdot C_{u,v} + z \tag{7}$$

Each weight function $g_i(u, v)$ defines a cost for edge (u, v) when the data flow from s_i travels through that edge. The pseudocode for STH is shown in Figure 4. At each iteration, STH simply finds the shortest path from one of the sources to the sink according to weight function (7). The output of STH is the union of all shortest paths found. Note that the cost of an edge needs to be updated during each iteration (step 2.a), since the cost depends on the data rate of the current source (according to Eq. (7)).

Figure 5 shows an example of the STH algorithm. Figure 5(a) shows an initial network without any flows. Figure 5 (b) and (c) show two iterations of STH. In each iteration, $G(V, E)$ is weighted according to g_i, and the shortest path from s_i to t is found. The output of STH is the graph composed of all shortest paths found. According to Eq. (2), the average power cost (excluding the cost of the sink) can be calculated to be 9.4.

Step 4 of the STH algorithm can be implemented using Dijkstra's [1959] shortest-path algorithm. The complexity of STH is $O(|S||E| \lg|V|)$. It can be seen that STH outputs the optimal solution for the two polynomial-time special cases of the MPC problem discussed in Section 4.

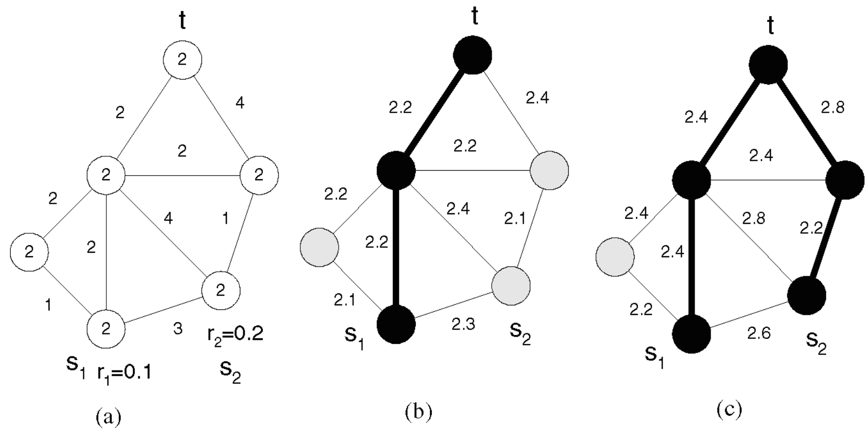


Fig. 5. (a) Initial network with edge weight $C_{u,v}$ and node weight $z = 2$ (shown on each node). (b) edge weights are defined by $r_1 \cdot C_{u,v} + z$. (c) edge weights are defined by $r_2 \cdot C_{u,v} + z$. The shortest paths from s_1, s_2 to t are highlighted in black.

Before we investigate the performance bound of STH for the general MPC problem, we define the following notation. We define a set of weight functions w_i for edge (u, v) as follows:

$$w_i(u, v) = r_i \cdot C_{u,v} \tag{8}$$

Here, $w_i(u, v)$ represents the cost of edge (u, v) when the data flow from s_i travels through (u, v). Let $P_G^x(u, v)$ represent the cost of the shortest path between nodes u and v in graph G under the weight function x. Then (2) can be reformulated as follows:

$$P(G') = \sum_i P_{G'}^{w_i}(s_i, t) + |V'|z \tag{9}$$

We have the following theorem regarding the performance of STH.

THEOREM 5.1. *The approximation ratio of STH is no greater than* $|S|$.

PROOF. Let $P(G')$ and $P(G'_{min})$ represent the total cost of G' found by STH and the optimal solution, respectively. The total cost of the shortest paths found by STH in G' with weight g_i is greater than in $P(G')$ because the idle power z of each node in G' might be counted multiple times. We have

$$P(G') \leq \sum_i P_{G'}^{g_i}(s_i, t). \tag{10}$$

Since STH finds the shortest paths in G with weight g_i and $G'_{min} \subset G$, we have

$$\sum_i P_{G'}^{g_i}(s_i, t) \ \leq \ \sum_i P_{G'_{min}}^{g_i}(s_i, t). \tag{11}$$

Consider the total cost of the shortest paths from s_i to t in G'_{min} with weight g_i. This cost is greater than the optimal solution $P(G'_{min})$, since weight z might be

Input: $G(V, E)$, source set S, sink t and traffic demands I
Output: $G^{'}(V', E')$

(1) Initialize $G^{'}(V', E')$ to be empty.
(2) Label all nodes as *asleep*.
(3) $W = S$.
(4) **while** $W \neq \phi$
 (a) Find $s_i \in W$ that has the shortest distance in $G(V, E)$ to t with edge weight $h_i(u, v)$.
 (b) Add the shortest path from s_i to t in $G^{'}$.
 (c) Label all nodes on the path as *active*.
 (d) $W = W - s_i$.
(5) **end**

Fig. 6. Incremental shortest-path tree heuristic (ISTH).

counted multiple times for each node in $G^{'}_{min}$. It can be seen that z is counted at most $|S|$ times for each node (which occurs when a node lies on paths from all sources to the sink). Thus we have

$$\begin{aligned}\sum_i P^{g_i}_{G^{'}_{min}}(s_i, t) &\leq \sum_i P^{w_i}_{G^{'}_{min}}(s_i, t) + |S|(|V^{'}|)z \\ &\leq |S|\left(\sum_i P^{w_i}_{G^{'}_{min}}(s_i, t) + (|V^{'}|)z\right). \\ &= |S|P(G^{'}_{min}) \end{aligned} \tag{12}$$

From Eqs. (10) to (12), we have

$$P(G^{'}) \leq |S|P(G^{'}_{min}). \qquad \square$$

5.3 Incremental Shortest-Path Tree Heuristic (ISTH)

In STH, the function used to weight the network is different for each source. Consequently, the shortest path from a source to the sink is not affected by whether shortest paths are already established for other sources. Intuitively, this does not seem efficient, since sharing an existing path could lead to lower nodal costs. Suppose that we are finding the shortest path from s_i to t and that all the shortest paths from $s_j (0 < j < i)$ to t have already been found. If any edge on the existing paths is reused by the new path, the incremental cost is $r_i \cdot C_{u,v}$. This cost does not include the nodal cost z, since it has been counted by the existing paths. In other words, the edge weights on existing paths should not include the nodal cost z. Based on this observation, we propose the following algorithm, called the *incremental shortest-path tree heuristic (ISTH)*, which finds that path from each source to the sink with the minimal *incremental* cost. The pseudocode of ISTH is depicted in Figure 6. During its execution, the algorithm maintains a subgraph $G^{'}$ that contains those paths from sources to sink that have been visited so far. In each iteration, ISTH finds the remaining source node that is closest to, but not connected to, the sink in $G^{'}$. It then adds the shortest path from that node to the sink into $G^{'}$. For convenience, we refer

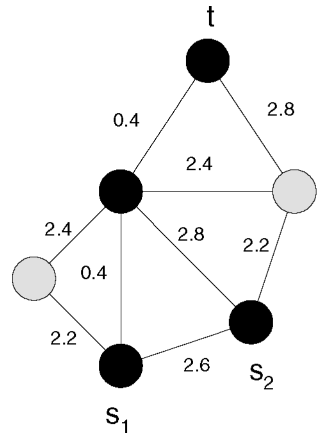


Fig. 7. The shortest path from s_2 to t shares an edge with the existing shortest path from s_1 to t.

to the state of those nodes already in $G^{'}$ as active. Once a node becomes active (i.e., included in $G^{'}$), the cost of any edge originating from it is decreased by z to reflect the incremental cost incured by the edge when a new flow travels through it. Formally, when ISTH finds the shortest path from source s_i to the sink, the edge cost is defined by the following function:

$$h_i(u, v) = \begin{cases} r_i \cdot C_{u,v} & \text{u is } \textit{active} \\ r_i \cdot C_{u,v} + z & \text{otherwise} \end{cases} \tag{13}$$

Figure 7 shows the second iteration of an example of ISTH in which the shortest path from s_1 to t has been found. The first iteration of the example is the same as that of STH, shown in Figure 5(b). The total weights on the shortest path from s_1 to t in Figure 7 are smaller than those in Figure 5(c), since the nodal cost z is not included. Consequently, different from the case of STH where two paths must always be disjoint (as shown in Figure 5(c)), the shortest path from s_2 to t shares an edge with the existing path. The total number of nodes used is therefore decreased, resulting in less idle energy consumption. According to Eq. (2), the average power cost in this example (excluding the cost of the sink) can be calculated to be 7.6. This value is smaller than the one obtained for the solution to STH. It can easily be seen that this solution is optimal for this example.

We now prove that the approximation ratio of ISTH is at least as good as that of STH.

THEOREM 5.2. *The approximation ratio of ISTH is no greater than $|S|$.*

PROOF. Let $P(G^{'})$ and $P(G^{'}_{min})$ represent the total cost of $G^{'}$ found by ISTH and the optimal solution, respectively. Here, $P(G^{'})$ equals the sum of the costs of all shortest paths found by ISTH. We have

$$P(G^{'}) \;=\; \sum_i P^{h_i}_{G^{'}}(s_i, t).$$

According to Eqs. (13) and (7), $h_i \leq g_i$. Hence, the incremental cost found by ISTH at each iteration must be no greater than that found by STH. We have

$$\sum_i P_{G'}^{h_i}(s_i, t) \leq \sum_i P_{G'}^{g_i}(s_i, t). \tag{14}$$

According to (14), (11), and (12), we have

$$P(G^{'}) \leq |S| P(G^{'}_{min}). \quad \square$$

As mentioned earlier, when $\forall(u, v) \in E, C_{u,v} = 0$, the MPC problem is equivalent to finding the minimum-weight Steiner tree connecting all the sources and the sink in G with uniform edge weight z. In ISTH, once a shortest path is found, the weights on the path become zero. Finding a subsequent shortest path from a source to the sink is therefore equivalent to finding the shortest path to any node on the existing path. In such a case, ISTH is equivalent to the minimum-weight Steiner tree heuristic with an approximation ratio of 2 [Gilbert and Pollak 1968]. This result suggests that ISTH yields good performance when idle energy dominates the total energy consumption of a network. Such a situation occurs when network workload or transmission/reception power is low. Similar to STH, algorithm ISTH finds the optimal solution for the two polynomial-time special cases of the MPC problem.

At each iteration of ISTH (see Figure 6), the data source closest to the sink is chosen for processing from among all remaining sources. Since this operation requires knowing about every source in the network, it can not be implemented online. A straightforward modification to handle online sources is to process one new source at each iteration of the algorithm. Although this modification will likely result in average performance degradation, the approximation ratio of ISTH, $|S|$ (where S is the set of sources), remains unchanged. This holds true because the proof of Theorem 5.2 does not require any particular sequence for the processing of sources. This property allows ISTH to preserve its performance bound in online scenarios.

We have been focusing on the scenario involving a single sink in this section. As STH and ISTH are based on pairwise, shortest-path heuristics, they can easily be extended to a scenario containing multiple sinks. It can be shown that the approximation ratio of both algorithms still holds using similar proofs.

5.4 Constant-Ratio Approximation Algorithm

Although the STH and ISTH algorithms described previously do find the optimal solution for the two polynomial-time special cases of the MPC problem, their known approximation ratio is equal to the number of source nodes in the network for the general MPC problem, causing them to scale relatively poorly when the number of sources becomes large. In this section, we seek an algorithm with a constant approximation ratio. We show in the following theorem that a minimum-weight Steiner tree algorithm will lead to a constant approximation ratio for MPC problem when the ratio of maximal transmission power to idle power is bounded.

THEOREM 5.3. *Let H be the best approximation algorithm to the minimum-weight Steiner tree problem that has an approximation ratio β. If $\forall(u,v) \in E, C_{u,v} \leq \alpha z$, the solution by executing H in G with the uniform edge weight z has an approximation ratio $(1+\alpha)\beta$ to the optimal solution of the MPC problem.*

PROOF. Suppose $G'_{min}(V'_{min}, E'_{min})$ and $G'(V', E')$ are optimal solutions to the minimum-weight Steiner tree problem and the solution of algorithm H, respectively. Since H has an approximation ratio of β and all edges have the same weight z, we have

$$|V'| - 1 = |E'| < \beta|E'_{min}| = \beta(|V'_{min}| - 1). \tag{15}$$

Let $P(G')$ and $P(G'_{min})$ represent the cost of G' and $P(G'_{min})$, respectively, in the MPC problem. We ignore weight z for the constant sink node in both $P(G')$ and $P(G'_{min})$. Doing so neither affects the quality of G' nor the optimality of G'_{min}. We have

$$\begin{aligned} P(G') &= \sum_i \sum_{(u,v)\in f(s_i,t)} r_i \cdot C_{u,v} + (|V'| - 1)z \\ &\leq \sum_{(u,v)\in E'} \left(C_{u,v} \cdot \sum_i r_i \right) + (|V'| - 1)z. \end{aligned} \tag{16}$$

where $f(s_i, t)$ represents the shortest path with edge weight $C_{u,v}$ from s_i to t. Based on the assumption that the total workload in the network is lower than network capacity $\sum_i r_i \leq 1$, we have

$$\begin{aligned} P(G') &\leq \sum_{(u,v)\in E'} C_{u,v} + (|V'| - 1)z \\ &\leq \sum_{(u,v)\in E'} \alpha z + (|V'| - 1)z \\ &= |E'|\alpha z + (|V'| - 1)z \\ &= (|V'| - 1)(1+\alpha)z, \end{aligned} \tag{17}$$

According to Eqs. (15) and (17), we have

$$\begin{aligned} P(G') &< \beta(|V'_{min}| - 1)(1+\alpha)z \\ &< (1+\alpha)\beta \left((|V'_{min}| - 1)z + \sum_i P^{w_i}_{G'_{min}}(s_i, t) \right) \\ &= (1+\alpha)\beta P(G'_{min}). \end{aligned}$$

□

Theorem 5.3 shows that the Steiner-tree-based algorithm performs better when the ratio of communication power to idle power, namely α, is low. The intuition behind this result is that the algorithm only minimizes the idle energy and ignores the transmission/reception energy of the radio, hence results in more energy reduction when idle energy constitutes a bigger portion of the total energy consumption, that is, α is low. Therefore, Theorem 5.3 indicates that the Steiner-tree-based algorithm is particularly suitable for radios with high idle power. Theorem 5.3 also shows that the performance of the algorithm

Input: $G(V, E)$, source set S, sink t and traffic demands I
Output: $G^{'}(V', E')$

(1) Set the weight of every edge in $G(V, E)$ to z.
(2) $V' = t$
(3) $W = S$.
(4) **while** $W \neq \phi$
 (a) Find $s_i \in W$ that has the shortest distance to $G^{'}$ with edge weight z.
 (b) Add the shortest path found in the previous step to $G^{'}$.
 (c) $W = W - s_i$.
(5) **end**

Fig. 8. The Gilbert minimum Steiner tree algorithm.

is dependent on β—the best approximation ratio of minimum Steiner tree algorithms. Approximate algorithms of the minimum Steiner tree problem have been studied extensively [Robins and Zelikovsky 2000]. The best-known approximation ratio is about 1.5 [Robins and Zelikovsky 2000]. According to the measurements of the CC1000 radio on Mica2 motes [Crossbow 2003], $\alpha \approx 2.3$. The approximation ratio of the scheme discussed in this section is therefore about 5 on the CC1000 radio.

Figure 8 shows a simple minimum Steiner algorithm proposed by Gilbert and Pollak [1968]. At step 4(a), the shortest path from a source s_i to G' is the shortest among the shortest paths from s_i to all nodes in G'. The algorithm has an approximation ratio of 2 [Gilbert and Pollak 1968]. In Section 6.2, we will discuss the design of a distributed protocol, called MASP, based on the Gilbert Steiner algorithm. The rationale of employing this algorithm instead of more complex ones with better approximation ratios is that this algorithm admits an efficient distributed implementation.

The Gilbert algorithm (see Figure 8) can be extended as follows to the scenario where sources arrive online. At step 4(a) of each iteration, a shortest path is found to connect the new source to the subgraph (composed of the sink and existing sources) before being added to the existing subgraph. The output is the subgraph composed of all sources and their respective found paths. This scheme has been shown to have an online approximation ratio of $\lg|S|$ to the minimum Steiner tree problem, where S is the set of nodes to be connected [Imase and Waxman 1991]. According to Theorem 5.3, the approximate ratio of this online algorithm for the MPC problem is $(1 + \alpha)\ \lg|S|$.

5.5 Performance Evaluation

In this subsection, we evaluate through simulations the average performance of the centralized approximate algorithms we presented in previous subsections. As discussed in Section 5.3, STH likely performs worse than ISTH and hence is not evaluated in this section.

We implement MBA, MBA-opt, ISTH, and the Gillbert Steiner tree algorithm (referred to as Steiner hereafter) in a network simulator. To evaluate the effectiveness of other energy conservation approaches to our problem, we

Table I. Radio Transmission Parameters

Tx Power (dBm)	Radio Range(m)	Current Consumption (mA)
−20	5	8.6
−10	18	10.1
0	50	16.8
5	68	25.4

also implemented two baseline algorithms called transmission-power minimum spanning tree (TMST) and transmission-power shortest-path tree (TSPT). TMST finds the minimum spanning tree of the network where each edge is weighted by the minimum transmission power of that edge. We choose TMST as a baseline algorithm for performance comparison, since distributed MST has been shown an effective topology-control algorithm [Li et al. 2003]. Similarly, TSPT finds the shortest-path tree of the network when weighted by transmission power, and this technique has been previously proposed as an efficient power-aware routing scheme [Singh et al. 1998].

We use the parameters of the CC1000 radio on Mica2 motes in the simulation. There is no packet loss in the simulation environment. The node bandwidth is 40Kbps. In the simulation, only those nodes that lie on the communication paths between sources and the sink remain active (i.e., the state of their radios is transmitting, receiving, or idle). All noncommunicating nodes run in sleeping state. The power consumption of the radio in receiving, idle, and sleeping states is 21 mw, 21 mw, and 6 μw, respectively [Crossbow 2003]. The actual radio range of the CC1000 on Mica2 motes varies, depending on environmental factors and transmitting power. We set the parameters of the radio range and transmitting power according to empirical measurements presented in Alessio [2004], which are listed in Table I. When a node communicates with a neighbor, it always uses the minimum radio range that can reach that neighbor. At the beginning of the simulation, a communication path from each source to the sink is found. The nodes on the communication paths remain active and all other nodes are put to sleep. The simulation time for each algorithm is 1,000 seconds. 200 nodes are randomly distributed in a $500m \times 500m$ region. The results in this section are the average of 10 different network topologies.

Figure 9 shows the total energy consumption of the network when the number of flows varies from 1 to 100. The data rate of each flow is 0.2Kbps. We can see that MBA-opt, ISTH, and Steiner significantly outperform the other algorithms. The good performance of Steiner and MBA-opt are expected because of their good approximation ratios. Interestingly, ISTH yields similar performance to MBA-opt and Steiner, although ISTH's known approximation ratio is worse. This result is due to the following facts. First, the performance bound of ISTH is derived under worst-case scenarios, which do not exhibit in the simulation. Second, although the aggregate data rate of all flows in the simulation is up to half of the network bandwidth, the data rate of each individual flow is very low. As a result, the active nodes on data routes remain idle most of the time. In such a case, ISTH minimizes the number of active nodes, resulting behavior similar to Steiner (i.e., g_i in Eq. (13) is close to zero). Figure 9

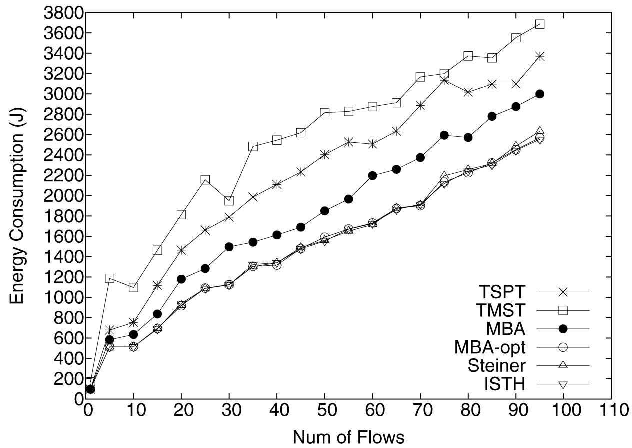


Fig. 9. Energy consumption vs. number of flows. The data rate of each flow is 0.2Kbps.

also shows the effectiveness of our optimization to the MBA algorithm, as discussed in Section 5.1. TSPT and MST result in considerably higher energy consumption than the aforementioned algorithms, since they only consider transmission power and ignore idle power.

The results in this section show that the average performance of ISTH and Steiner is similar to that of MBA-opt. As both ISTH and Steiner are based on the shortest-path algorithm, they have a more efficient distributed implementation than MBA-opt. We now turn our attention to the distributed implementation of ISTH and Steiner.

6. DISTRIBUTED PROTOCOLS

In this section, we present the design and implementation of two distributed routing protocols: minimum power configuration protocol (MPCP) and minimum active subnet protocol (MASP). These protocols are based on the centralized algorithms ISTH and Steiner (presented in Section 5), respectively. We focus on a "many-to-one" routing scenario in our discussion, since it is the most common communication paradigm in sensor networks. MPCP and MASP can be easily extended to support more general routing scenarios.

6.1 Minimum Power Configuration Protocol

In this section, we present the design of the minimum power configuration protocol (MPCP). MPCP finds power-efficient routes for the communicating nodes in a network based on the distributed implementation of the ISTH algorithm with online extentions (discussed in Section 5.3).

Shortest-path-based routing mechanisms have been extensively studied. We adopt the destination sequenced distance vector routing (DSDV) protocol [Perkins and Bhagwat 1994] as our implementation framework. DSDV is based on a distributed implementation of the Bellman-Ford shortest-path algorithm [Bertsekas and Gallager 1987]. A node in DSDV advertises its current routing cost to the sink by broadcasting *route update* messages. A node sets that

Table II. A Routing Table

Data Rate Packets/s	Next Hop	Cost	Seq
2.1	5	28.9	8
1	7	8.9	6
0.5	15	18.3	8
0.1	30	8.2	12

neighbor which has minimum cost for the sink as its parent and rebroadcasts its updated cost if necessary. DSDV can avoid the formation of routing loops by using sink-based sequence numbers for route updates. The routing cost of a node in DSDV is its hop count to the sink. The routing cost of a node in MPCP, however, depends on the operational state of the node (active or power saving), as well as the data rates of those flows that travel through the node. We now discuss in detail the core components of MPCP.

6.1.1 *Node States and Routing Table.* In our design, a node operates in either *active* or *power-saving* mode. A node in power-saving mode remains asleep most of the time and only periodically wakes up. This simple sleep schedule is similar to several existing power-saving schemes such as SMAC [Ye et al. 2002]. Initially, all nodes operate in power-saving mode. When a source node starts sending data to the sink, a power-efficient routing path from source to sink is found by the distributed ISTH algorithm. All nodes on the routing path are activated to relay data from the source to the sink. All other nodes remain in power-saving mode to reduce energy consumption. Similarly, an active node switches to power-saving mode if all the data flows traveling through it disappear.

Each node in the network maintains a routing table that contains the routing entries and status of neighbors. Since the routing cost to the sink varies with the data rate of the source, we need to store an entry for each data rate in the network. Specifically, an entry in the routing table of node u includes the following fields: $< r_i,\ next\_hop,\ cost,\ and\ seq >$, where r_i is the data rate of source s_i, *next_hop* is the neighbor node with the minimum cost for the sink, *cost* is the cost of node u for the sink through *next_hop*, and *seq* is a sequence number originated by the sink. Table II shows a routing table of an active node.

One simple method for obtaining source rates is to let each source flood the network with its rate information before finding a route to the sink. This approach incurs too much overhead, however, when a network is composed of many nodes. To reduce the overhead, only those data rates with significant difference are kept in the routing table. When a new source node starts sending data, it chooses that next hop node (from a routing table entry) which has the data rate closest to its own . The new data rate will then be propagated to other nodes if it is significantly different from the ones stored in their table.

6.1.2 *Route Updates.* According to cost function (13), the routing cost from a node to its neighbors in MPCP depends on data rate and the change of the node's state (active or power saving). As a result, a new round of route updates will be triggered by any of the following events: (1) a link is broken;

(2) the data rate of an existing flow changes; or (3) a data flow is started or completed.

A node detects a broken link when multiple transmissions fail. The process of route updates caused by a broken link is similar to that of DSDV. A node advertises its routing information by broadcasting a route update packet to its neighbors. After receiving an update from a neighbor, a node calculates its new cost to the sink at each data rate specified in the update, and updates its routing table. This new cost is equal to the sum of the link cost to the neighbor (defined by Eq. (13)) and the cost of the neighbor included in the update. A node sets a timeout after the arrival of the first route update in this round so as to wait for more updates from other neighbors. If there exist entries in the routing table that have a cost reduction above a threshold after the timeout, the node broadcasts a route update packet containing these entries to advertise its updated routing information.

We now discuss in detail the route updates caused by the change of data rate and start/completion of a data flow. When a source node changes its data rate to a value differing significantly from the data rates stored in the routing table, the source node notifies the sink by including the new rate in its data packets. Once the sink sees the new rate, it broadcasts a route update with a new sequence number to the network. The routing tables of nodes are updated when the route update is broadcast throughout the network. Consequently, the source with the new data rate may choose a better route due to updated routing information. When the workload of the network is dynamic, multiple rounds of route updates may be initiated at the same time, resulting in high network contention. To reduce the overhead of route updates in such a case, the sink can include several default data rates in its initial route updates, based on the estimation of source rates. From then on, a new round of route updates is initiated only when the data rate of a flow changes to a value significantly different from the default ones.

Route updates may also be triggered when a new data flow appears. If the new flow has a data rate significantly different from those stored in the routing table, a round of route updates is initiated, as discussed earlier. In addition, the appearance of a new flow may activate a node previously running in power-saving mode and reduce the cost of the node to its neighbors (see Eq. (13)). As shown in Figure 10, a new data flow from source node A activates nodes A, B, and C before it meets the existing routing path at a junction node D (D may be the sink node). Nodes A, B, and C then lower their routing costs after being activated. In such a case, to reduce the number of route updates, only that node preceding the junction node initiates the route update, since it has the minimum cost to the sink among all nodes on the new path. In Figure 10, node C will broadcast a route update with a new sequence number and reduced routing costs in order to initiate a round of route updates. Nodes B, A, and others having reduced routing costs to the sink participate in the route update process that has been initiated by C. Note that route updates initiated in this way only involve a subset of nodes in the network, since many nodes (e.g., those closer to the sink) will not participate in the route update process due to the lack of reduction in their routing costs.

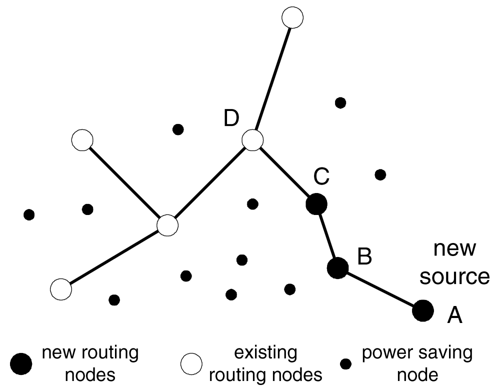


Fig. 10. Node A is a new source. The junction node C will initiate a round of route updates.

Similar to the appearance of a new flow, the disappearance of an existing flow may also cause route updates. In such a case, nodes on the existing routing path switch to power-saving mode after some timeout, resulting in higher routing costs (see Eq. (13)). Again, that node preceding the junction node initiates the route update process by advertising the new routing costs.

6.1.3 *Link Estimation.* In real wireless sensor networks, a routing protocol often suffers from dynamic and lossy communication links. Empirical study shows that the reliability of routing protocols can be significantly improved by only keeping "good" neighbors, such as, those with high packet perception ratios (PRRs), in neighborhood tables [Woo et al. 2003]. A simple way to obtain the PRR of a link is by profiling the link characteristics offline. Alternatively, the PRR can be obtained from online link estimators [Woo et al. 2003; Chipara et al. 2006]. For example, nodes can broadcast periodic beacon messages as well as the PRR of a link to a neighbor being estimated by counting the number of messages received from that neighbor. Further discussion on this issue is beyond the scope of this article.

6.2 Minimum Active Subnet Protocol

We now present the design of the minimum active subnet protocol (MASP) that finds a Steiner tree connecting all sources in the network to the sink using the minimum number of active nodes. The MASP is also based on DSDV and has a similar design to MPCP, as both protocols are based on the shortest-path algorithm. We will now discuss the major difference between MPCP and MASP.

In MASP, a node in power-saving mode incurs a routing cost of P_i (idle power).[3] Once a data flow travels through a node, it becomes active and its routing cost reduces to zero. In other words, routing among active nodes is free. As a result, when a new source arrives, finding the shortest path from that node to the sink is equivalent to finding the shortest path to any active node.

[3]Since the routing cost is the same for all power-saving nodes, one can use any positive number as the routing cost.

Unlike MPCP, the routing cost of a node in MASP does not depend on data rates. This independence reduces the storage overhead of the routing table at each node, as well as the network bandwidth used by route updates. Each entry of a routing table in MASP contains <*next_hop*, *cost*, *seq*>. The route updates of MASP can be triggered by either a broken link, or the start or completion of a data flow. Route updates triggered by link failures are similar to DSDV, while the updates triggered by sources are similar to MPCP. Moreover, MASP is expected to generate fewer routing updates than MPCP because the change in data rates does not affect the routing cost of MASP. In other words, MASP ignores data rates because it only minimizes idle energy. As shown in our simulation results presented in Section 7, MASP is only suitable for radios with high idle power.

7. EXPERIMENTATION

7.1 Simulation Environment

Low-power wireless radios used by real sensor network platforms (e.g., Berkeley motes) are known to have highly irregular communication ranges and probabilistic link characteristics [Zhao and Govindan 2003]. The simplistic assumptions on wireless radio propagation made by a network simulator may cause the simulation results to differ significantly from real-world experimental results [Kotz et al. 2004]. Accurate simulation of the characteristics of real wireless radios with different transmission powers is key to evaluating the realistic performance of our protocols. Because of this we took a link-layer model developed by USC [Zuniga and Krishnamachari 2004] and implemented it for use with the Prowler network simulator [Simon 2007]. We also added improved routing support to this model based on work done during the Rmase project [Zhang 2007]. Experimental data showed that the USC model can simulate highly unreliable links in Mica2 motes [Zuniga and Krishnamachari 2004]. In our simulations, the packet reception ratio (PRR) of each link is governed by the USC model according to the distance between the two communicating nodes and the transmission power. The MAC layer in Prowler employs a simple CSMA/CA scheme without RTS/CTS, which is similar to the B-MAC protocol [Polastre et al. 2004] in TinyOS. To improve communication reliability in the lossy simulation environment, we implemented an ARQ (automatic repeat request) scheme that retransmits a packet if an acknowledgment is not received after some preset timeout. The maximum number of retransmissions before dropping a packet is 8. Prowler is a Matlab-based network simulator that employs a layered event-driven structure similar to TinyOS. Using such a simulator allows us to easily implement new network modules (such as the link model from USC) and to port our protocols to Berkeley motes in the future.

7.2 Simulation Settings

For performance comparison, in addition to MPCP and MASP, we have implemented two baseline protocols: minimum transmission (MT) routing [Woo et al. 2003] and minimum transmission power (MTP) routing. These have similar components to MPCP, except for the cost metrics. MT is shown to be more reliable than the hop-count-based routing scheme when given a lossy network [Woo et al. 2003]. A node in MT chooses the next hop node with the minimum expected number of transmissions to the sink. All communication links in the original MT protocol use the same transmission power. A link between nodes u and v in MT has a cost of $\frac{1}{PRR(u,v)}$. To take advantage of variable transmission power, we modified the link cost of MT to $\frac{1}{PRR(u,v,P_{tx}(u,v))}$, where $P_{tx}(u, v)$ is defined in Eq. (6). A node in MTP chooses the next hop node with minimum total expected transmission power to the sink. The cost of a link between u and v in MTP is equal to $\frac{P_{tx}(u,v)}{PRR(u,v,P_{tx}(u,v))}$. Except in consideration of unreliable links, MTP is similar to the minimum power routing schemes studied in Doshi et al. [2002] and Doshi and Brown [2002].

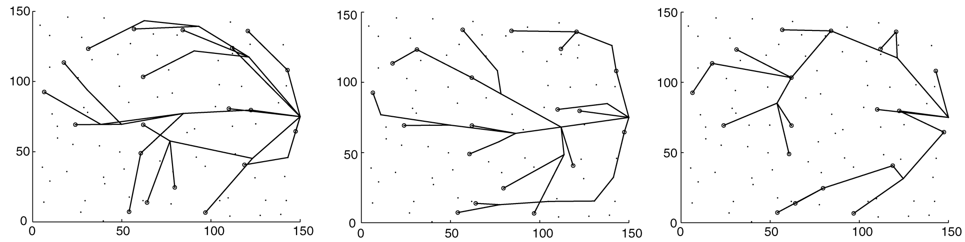
In each simulation, 100 nodes are deployed in a $150m \times 150m$ region divided into 10×10 grids. A node is randomly located within each grid. Source nodes are randomly chosen. The sink is located at (150, 75) to increase the hop count from some of the sources. The radio bandwidth is 40Kbps. Power parameters of the radio are set according to the empirical measurements of the CC1000 radio on Mica2 motes [Shnayder et al. 2004] as follows. The CC1000 radio is capable of transmitting data at 31 power levels ranging from -20 dBm to 10 dBm. To simplify our design, we chose 10 power levels from the total of 31 levels. The corresponding current consumption ranges from 3.7 mA to 21.5 mA. The receiving and idle current is 8 mA. Each simulation lasts for 400 seconds. Each source sends packets at a randomly chosen interval of $10 \sim 14$ seconds, which corresponds to an average data rate of between 68.5 to 96 bps. The number of sources varies from 5 to 30, which results in a total data rate of 0.4 to 2.4Kbps at the sink. Real-world experiments show that the maximum effective multihop bandwidth of Mica2 motes can barely reach 6Kbps due to channel contention and lossy wireless links [He et al. 2004], which conforms to our observation in simulations.

During the initialization state, a source node starts sending data at some random time after its route to the sink is found. After this initialization phase, a node that does not lie on any communication path will enter power-saving mode automatically, as discussed in Section 6. The power-saving mode has a period of 10 seconds and an active window of 1 second. The data packet size is 120 bytes. A routing update packet is 40 bytes. The results in this section are the average of 5 different network topologies.

7.3 Performance of MPCP

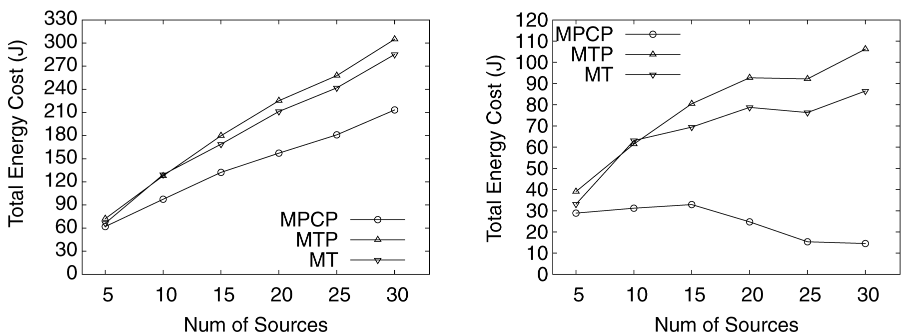
In this section, we evaluate the performance of MPCP. Since the performance of MASP varies with a platform-dependent parameter α (see Section 5.4), we compare it with MPCP under different platform parameters in Section 7.4.

Figure 11 shows the routing topologies produced by different protocols in a typical run with 20 sources. The circles in the figure represent data sources and small dots represent other nodes. We can see that the topologies produced by MT and MTP are similar and both have over 33 active nodes on the communication paths. In contrast, MPCP activates only 24 nodes, that is, 4 more



(a) MT (33 active nodes) (b) MTP (35 active nodes) (c) MPCP (24 active nodes)

Fig. 11. Routing topologies of different protocols with 20 sources.



(a) total network energy (b) total energy, excluding idle energy of sources

Fig. 12. Energy consumption of different protocols.

nodes besides the data sources that must remain active. As the number of data sources increases, MPCP can effectively reuse more active sources on different communication paths and hence further reduce the number of active nonsource nodes. For example, MPCP activates only 1 nonsource node when there are 30 sources. This result clearly illustrates that MPCP can significantly reduce the energy wasted for idle listening by sharing active nodes on different communication paths.

The most important metric for our performance evaluation is energy consumption. For each protocol, we measure the difference between the total energy cost of the communicating network and that of an idle network where there is no communication activity and all nodes run in power-saving mode. This metric indicates the net energy consumed by a protocol due to the communication activities of the network. As shown in Figure 12(a), MPCP consumes considerably less energy than other protocols. As the number of sources increases, routing paths from different sources share more nodes under MPCP and MASP, resulting in more energy reduction in the idle state and better energy efficiency. The overall energy reduction of MPCP can be as high as 30% over MTP and 26% over MT.

Another interesting result in Figure 12(a) is that although MTP optimizes the transmission energy, it has a total energy cost similar to that of MT, even though MT makes simpler routing decisions based on the number of transmissions.

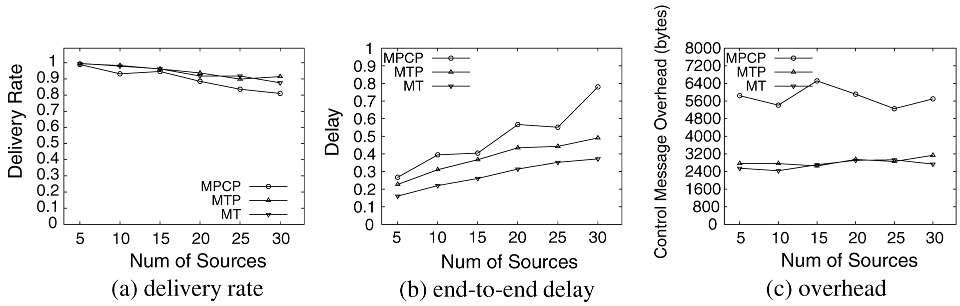
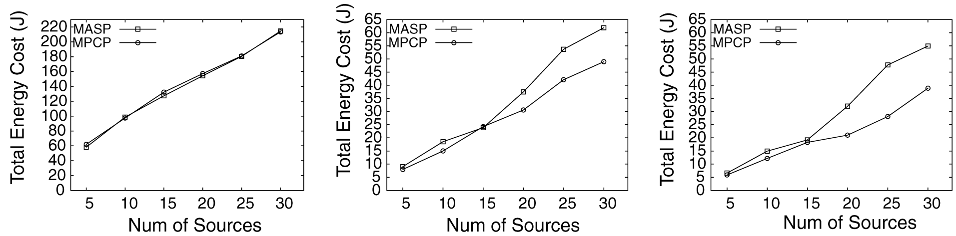


Fig. 13. Communication performance and overhead of different protocols.

As transmission power grows quickly with transmission distance, the routing paths found by MTP are likely to consist of more hops. Consequently, more nodes have to remain active on the routing paths, resulting in more energy wastage due to idle listening. On the other hand, although MT does not optimize transmission energy, its routings paths contain fewer hops and hence more nodes can run in power-saving mode. In contrast to MTP or MT, which only reduce the radio energy costs under partial working modes, MPCP effectively minimizes the total energy cost of radios based on data rates.

We observe that when the number of source nodes is large, most of the energy consumption is due to the idle listening of sources. This phenomenon reduces the difference in total energy consumption between different protocols. To focus our analysis on the energy consumption of nonsource nodes, we measure the difference between the total energy consumption of the network and that of the same network where there is no communication activity, that is, a network where all nonsource nodes remain in the power-saving mode but all source nodes remain in the idle state. This metric indicates the net energy consumption of communication activities *excluding* the idle listening of source nodes. Figure 12(b) shows that MPCP consumes at most 86% less energy than MT and 83% less than MTP. This result is consistent with the observation from the routing topology of MPCP in Figure 11(c), namely, that MPCP activates many fewer nonsource nodes by effectively sharing intermediate source nodes on different paths. Another interesting result in Figure 12(b) is that MPCP may consume less energy on intermediate nodes as the number of sources increases. This is because MPCP tends to route the data from a source through other sources that must remain active anyway. Reusing these sources results in lower routing costs to the sink. More intermediate nodes may, therefore, run in power-saving mode as the number of sources increases. We note that although the energy reduction by routing through other active sources is generally viable in the many-to-one communication pattern, it may be affected by the spatial distribution of sources in other scenarios.

Next we evaluate the communication performance of the various protocols. Figure 13(a) shows the data delivery ratio at the sink under different protocols. We can see that the delivery ratio of all protocols decreases slowly, the more sources there are in the network. MPCP delivers slightly less data than the other protocols when the number of sources exceeds 15. This occurs because



(a) idle current is 8mA (b) idle current is 0.365mA (c) idle current is 0.02mA

Fig. 14. Energy consumption on different platforms.

MPCP causes slightly higher network contention due to path sharing between different sources when the network workload is high.

We plot the average end-to-end delay of data packets in Figure 13(b). Not surprisingly, MT yields the shortest latency, since it finds the routing paths with fewer retransmissions. MPCP yields a higher latency when network workload becomes higher due to the network contention caused by path sharing between different sources.

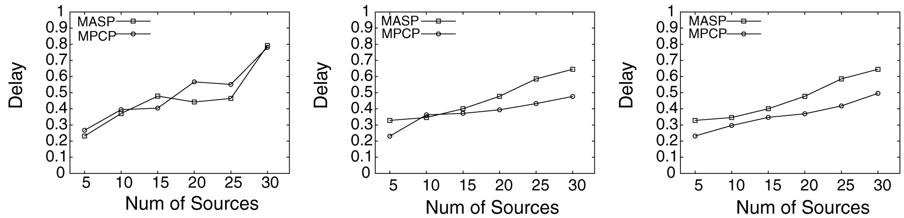
Finally, Figure 13(c) shows the overhead of different protocols in terms of the total number of useful bytes in all route update messages. The overhead of MT and MTP is similar and remains roughly constant as more sources appear. MPCP incurs a higher overhead because the appearance of a new source node changes the node states and routing costs (see Eq. (13)), triggering more route updates than MTP and MT. However, consistent with the discussion in Section 6, most route updates are triggered by the first several sources and hence the total number of updates remains roughly the same as the number of sources increases. This behavior allows MPCP to scale well to large-scale networks. Despite the additional overhead compared with MT and MTP, MPCP still achieves significantly less energy consumption, as shown in Figures 12(a) and (b).

7.4 Comparison of MPCP and MASP

As discussed in Section 6.2, MASP may incur a lower overhead than MPCP because it does not depend on information about the current set of sources and their data rates. A disadvantage of MASP, however, is that its energy performance depends on the power characteristics of the radio. We now compare the performance of MPCP and MASP with different radio characteristics.

With advancement in radio technology, the idle power of radio will continue to decrease in the future. To measure the impact of radio characteristics on MPCP and MASP, we simulate the two protocols using three different idle currents: 8 mA, 0.365 mA, and 0.02 mA. These three idle currents span three different orders of magnitude, hence allowing us to evaluate the energy performance of MPCP and MASP on a wide range of possible radio platforms. The transmission/reception current remains the same as the setting used in Section 7.2.

Figure 14 shows the energy consumption of MPCP and MASP. When the idle current is 8 mA, MASP consumes similar energy as MPCP, even though MASP only minimizes the number of active nodes and does not directly optimize the



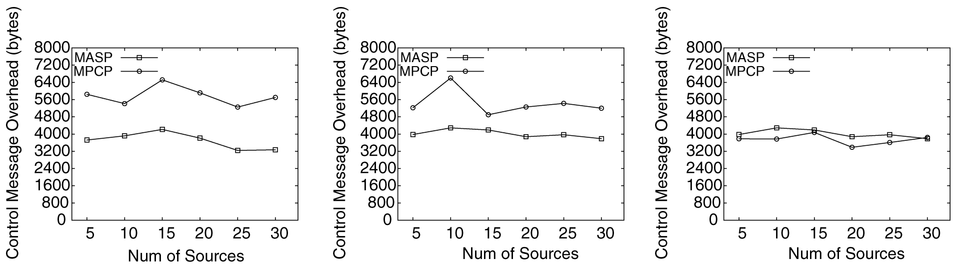
(a) idle current is 8mA (b) idle current is 0.365mA (c) idle current is 0.02mA

Fig. 15. End-to-end delay on different platforms.

overall energy consumption, as does MPCP. Hence, MPCP considerably outperforms MASP when the idle current is lower. This result can be explained as follows. First, the achievable maximum bandwidth on multihop networks is fairly low compared with an ideal radio bandwidth. For example, the practical maximum bandwidth of Mica2 motes can barely reach 6Kbps, due to channel contention and lossy wireless links [He et al. 2004]. This results in having only one-sixth of the ideal radio bandwidth. Consequently, most energy consumption is due to idle listening of nodes instead of transmission/reception when the idle current is 8 mA. In other words, the impact of data rates on overall energy consumption is limited when the idle current is high, making MPCP behave similarly to MASP, as discussed in Section 5.3. When the idle current is 0.365 mA or 0.02 mA, the transmission/reception energy dominates the total energy consumption. In such a case, the performance of MASP degrades significantly, as it only aims at minimizing the idle listening energy. This performance degradation of MASP is consistent with the analysis of the Steiner algorithm on which MASP is designed. As discussed in Section 5.4, the approximation ratio of the Steiner algorithm increases with α, which in turn increases as the idle current becomes lower. In contrast, MPCP yields a much better performance than MASP when the idle current is low because it always minimizes the total energy consumption of all radio states.

Figure 15 shows the end-to-end packet delay under MPCP and MASP. Consistent with the results on energy consumption, MPCP performs similarly to MASP when the idle current is 8 mA and considerably outperforms MASP when the idle current is 0.365 mA or 0.02 mA. When idle current is low, the routing cost under MPCP is dominated by the transmission/reception power (see Eq. (13)), resulting in a shortest-path-tree-like routing topology with more intermediate nodes than the Steiner-tree-like routing topology of MASP. A packet therefore travels fewer hops to the sink under MPCP, causing shorter end-to-end delay.

Finally, Figure 16 shows the overhead of MPCP and MASP in terms of the total number of useful bytes in all route update messages. We can see that MASP incurs significantly lower overhead than MPCP when the idle current is 8 mA. This is due to the fact that each route update of MPCP contains more routing information, as the routing cost depends on data rates. MPCP does, however, incur a lower overhead as the idle current decreases. In particular, MPCP incurs a overhead similar to MASP when the idle current is 0.02 mA. As the idle current decreases, the impact of node state (i.e., whether or not a node



(a) idle current is 8mA (b) idle current is 0.365mA (c) idle current is 0.02mA

Fig. 16. Routing overhead on different platforms.

is active) on routing cost decreases accordingly. As a result, the activation of nodes due to the appearance of new data flows causes fewer route updates. In contrast, MASP generates a similar number of route updates for all three idle currents because the routing cost of a node in that protocol only depends on its state (i.e., whether the node is active).

The results in this section indicate that MPCP preserves satisfactory performance under a wide range of radio characteristics. When idle power of the radio is high, it reduces the energy wasted in the idle state by minimizing the number of active nodes. On the other hand, when the idle power of the radio is low, it saves energy by reducing transmission energy. This joint optimization approach adopted by MPCP enables flexible adaptation to different radio platforms. In contrast, MASP is only suitable for radios with high idle power and introduces less overhead than MPCP.

8. DISCUSSION

In this section, we discuss several limitations of this article and potential future work.

In our problem formulation, every node in the network operates in a constant state (active or sleeping) during communication. The simulation results in Figures 12(a) and 12(b) show that further energy savings can be achieved by reducing the idle time of active nodes (e.g., through sleep management). Moreover, the MPC problem could be solved optimally if there existed an *ideal* sleep management scheme that could schedule an active node to sleep whenever it becomes idle and wake up whenever data arrives. Data arrival times can, however, be highly unpredictable in a multihop communication environment, even with periodic data sources. Hence, scheduling actively communicating nodes to sleep may result in high communication delays or even data loss. We note that sleep scheduling schemes (e.g., ESSAT [Chipara et al. 2000], on-demand power management [Zheng and Kravets 2003], T-MAC [van Dam and Langendoen 2003]) that are adaptive to the traffic in the network are suitable for use with MPCP to further reduce the idle energy consumption of active nodes. with MPCP in future work.

While our approach mainly focuses on minimizing the total energy consumption of a network, it may not lead to maximal system lifetime. Nodes on shared routing paths found by MPCP deplete energy faster than other nodes, which may result in network partitions. We will extend MPCP to incorporate appropriate routing metrics (e.g., those based on node residual energy) to achieve more balanced energy dissipation and prolong network lifetime [Singh et al. 1998; Li et al. 2001]. Finally, while we have focused primarily on many-to-one workloads, MPCP can be extended to more general workload models with multiple sinks.

9. CONCLUSION

In this article we have proposed the minimum power configuration approach to minimizing the total energy consumption of WSNs. We first formulated the energy minimization problem as a joint optimization problem in which the power configuration of a network consists of a set of active nodes and the transmission power of these nodes. We have presented a set of approximation algorithms with provable performance bounds, and the practical MPCP protocol that dynamically (re)configures a network based on current data rates. We have also proposed a more efficient protocol, called MASP, that only minimizes the total number of active nodes in a network. Simulations based on a realistic radio model of Mica2 motes show that MPCP can conserve significantly more energy than representative topology control and power-aware routing schemes. Furthermore, while MASP is suitable for radios with high idle power, a key advantage of MPCP is that it yields satisfactory performance under a range of representative radio characteristics, allowing it to flexibly adapt to different radio platforms.

ACKNOWLEDGMENTS

This work is funded in part by the NSF under ITR Grant CCR-0325529 and NeTS-NOSS Grant CNS-0627126, and by the DARPA under Contract F33615-01-C-1904. We thank R. Ravi at Carnegie Mellon University for his insightful discussion on this work.

REFERENCES

ALESSIO, F. 2004. Sensor networks: Performance measurements with motes technology. Tech. Rep., University of Pisa, Italy.

ALZOUBI, K., LI, X.-Y., WANG, Y., WAN, P.-J., AND FRIEDER, O. 2003. Geometric spanners for wireless ad hoc networks. *IEEE Trans. Parallel Distrib. Syst. 14*.

BERTSEKAS, D. AND GALLAGER, R. 1987. *Data Networks*. Prentice-Hall, Upper Saddle River, NJ.

CALINESCU, G., KAPOOR, S., OLSHEVSKY, A., AND ZELIKOVSKY, A. 2003. Network lifetime and power assignment in ad-hoc wireless networks. In *Proceedings of the Annual European Symposium on Algorithms (ESA)*.

CHANG, J.-H. AND TASSIULAS, L. 2000. Energy conserving routing in wireless ad hoc networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. 2001. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*.

CHIPARA, O., HE, Z., XING, G., CHEN, Q., WANG, X., LU, C., STANKOVIC, J., AND ABDELZAHER, T. 2006. Real-Time power-aware routing in sensor networks. In *Proceeding of the IEEE International Workshop on Quality of Service (IWQoS)*.

CHIPARA, O., LU, C., AND ROMAN, G.-C. 2000. Efficient power management based on application timing semantics for wireless sensor networks. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*.

CROSSBOW. 2003. Mica and mica2 wireless measurement system datasheets. http://72.14.209.104/searchPq=cache:uw-483FWGsUJ:www.xbow.com/Products/Product-pdf-Files/Wireless-pdf/MICA2-Datasheet.pdf+Crossbow+Micatandtmicaz+wireless+measurement+system+data+sheet&-hl=en&-ct=clnk&-cd=1&-gl=us

DIJKSTRA, E. W. 1959. A note on two problems in connexion with graphs. *Numer. Math. 1*, 269–271.

DONG, Q. 2005. Maximizing system lifetime in wireless sensor networks. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*.

DONG, Q., BANERJEE, S., ADLER, M., AND MISRA, A. 2005. Minimum energy reliable paths using unreliable wireless links. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

DOSHI, S., BHANDARE, S., AND BROWN, T. X. 2002. An on-demand minimum energy routing protocol for a wireless ad hoc network. *SIGMOBILE Mob. Comput. Commun. Rev. 6*, 3, 50–66.

DOSHI, S. AND BROWN, T. X. 2002. Minimum energy routing schemes for a wireless ad hoc network. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

ERGEN, S. C. 2002. Pedamacs: Power efficient and delay aware medium access protocol for sensor networks. M. S. thesis, University of California at Berkeley.

GAREY, M. R. AND JOHNSON, D. S. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman.

GILBERT, E. N. AND POLLAK, H. O. 1968. Steiner minimal trees. *SIAM J. Appl. Math. 16*, 1–29.

HE, T., KRISHNAMURTHY, S., STANKOVIC, J. A., ABDELZAHER, T., LUO, L., STOLERU, R., YAN, T., GU, L., HUI, J., AND KROGH, B. 2004. Energy-Efficient surveillance system using wireless sensor networks. In *Proceedings of the 9th International Conference on Mobile System, Applications, and Services (MobiSys)*.

HOHLT, B., DOHERTY, L., AND BREWER, E. 2004. Flexible power scheduling for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*.

IEEE. 1999. Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Standard 802.11*.

IMASE, M. AND WAXMAN, B. 1991. Dynamic Steiner tree problem. *SlAM J. Discrete Math. 4*, 3, 369–384.

KAWADIA, V. AND KUMAR, P. R. 2003. Power control and clustering in ad hoc networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

KOTZ, D., NEWPORT, C., GRAY, R. S., LIU, J., YUAN, Y., AND ELLIOTT, C. 2004. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*.

LI, L., HALPERN, J. Y., BAHL, P., WANG, Y.-M., AND WATTENHOFER, R. 2001. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing*.

LI, N., HOU, J. C., AND SHA, L. 2003. Design and analysis of an MST-based topology control algorithm. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

LI, Q., ASLAM, J., AND RUS, D. 2001. Online power-aware routing in wireless ad-hoc networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*.

MAINWARING, A., CULLER, D., POLASTRE, J., SZEWCZYK, R., AND ANDERSON, J. 2002. Wireless sensor networks for habitat monitoring. In *Proceeding of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*. 88–97.

MEYERSON, A., MUNAGALA, K., AND PLOTKIN, S. 2000. Cost-Distance: Two metric network design. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*.

NARAYANASWAMY, S., KAWADIA, V., SREENIVAS, R. S., AND KUMAR, P. R. 2002. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *Proceedings of the European Wireless Conference*.

PERKINS, C. E. AND BHAGWAT, P. 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM Conference on Communications Architectures, Protocols and Applications*.

POLASTRE, J., HILL, J., AND CULLER, D. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor System (SenSys)*.

RAMANATHAN, R. AND HAIN, R. 2000. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

ROBINS, G. AND ZELIKOVSKY, A. 2000. Improved Steiner tree approximation in graphs. In *SODA*.

RODOPLU, V. AND MENG, T. H. 1999. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Commun. 17*, 8.

SANKAR, A. AND LIU, Z. 2004. Maximum lifetime routing in wireless ad-hoc networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

SHNAYDER, V., HEMPSTEAD, M., RONG CHEN, B., ALLEN, G. W., AND WELSH, M. 2004. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*.

SIMON, G. Probabilistic wireless network simulator. http://www.isis.vanderbilt.edu/projects/nest/prowler/.

SINGH, S., WOO, M., AND RAGHAVENDRA, C. S. 1998. Power-Aware routing in mobile ad hoc networks. In *Proceedings of the 4th Annual ACM-IEEE International Conference on Mobile Computing and Networking*.

STANKOVIC, J. A., ABDELZAHER, T., LU, C., SHA, L., AND HOU, J. 2003. Real-Time communication and coordination in embedded sensor networks. *Proc. IEEE 91*, 7.

SZEWCZYK, R., MAINWARING, A., POLASTRE, J., ANDERSON, J., AND CULLER, D. 2004. An analysis of a large scale habitat monitoring application. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*.

VAN DAM, T. AND LANGENDOEN, K. 2003. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*.

WATTENHOFER, M. AND WATTENHOFER, R. 2004. Distributed weighted matching. In *Proceedings of the 18th Annual Conference on Distributed Computing (DISC)*.

WOO, A., TONG, T., AND CULLER, D. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*.

XING, G., LU, C., ZHANG, Y., HUANG, Q., AND PLESS, R. 2005. Minimum power configuration in wireless sensor networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

XING, G., WANG, X., ZHANG, Y., LU, C., PLESS, R., AND GILL, C. 2005. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. on Senor Netw. 1*, 1, 36–72.

XU, N., RANGWALA, S., CHINTALAPUDI, K. K., GANESAN, D., BROAD, A., GOVINDAN, R., AND ESTRIN, D. 2004. A wireless sensor network for structural monitoring. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*.

XU, Y., HEIDEMANN, J., AND ESTRIN, D. 2000. Adaptive energy-conserving routing for multihop ad hoc networks. Res. Rep. 527, USC. October.

XU, Y., HEIDEMANN, J., AND ESTRIN, D. 2001. Geography-Informed energy conservation for ad hoc routing. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*.

YE, W., HEIDEMANN, J., AND ESTRIN, D. 2002. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

ZHANG, Y. 2007. Routing modeling application simulation environment. http://www2.parc.com/spl/projects/era/nest/Rmase/.

ZHAO, J. AND GOVINDAN, R. 2003. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)* (Los Angeles, CA).

ZHENG, R., HOU, J. C., AND SHA, L. 2003. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM Press, New York, 35–45.

ZHENG, R. AND KRAVETS, R. 2003. On-Demand power management for ad hoc networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

ZUNIGA, M. AND KRISHNAMACHARI, B. 2004. Analyzing the transitional region in low power wireless links. In *Proceedings of the 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SeCon)*.

Received September 2005; revised May 2006; accepted January 2007