Minimum Power Configuration in Wireless Sensor Networks

Guoliang Xing; Chenyang Lu
Washington University in St. Louis
St. Louis, MO 63130
{xing,lu}@cse.wustl.edu

Ying Zhang; Qingfeng Huang
Palo Alto Research Center (PARC) Inc.
3333 Coyote Hill Road
Palo Alto, CA 94304

Robert Pless
Washington University in St. Louis
St. Louis, MO, 63130
{pless}@cse.wustl.edu

ABSTRACT

This paper proposes the *minimum power configuration (MPC)* approach to energy conservation in wireless sensor networks. In sharp contrast to earlier research that treats topology control, power-aware routing, and sleep management in isolation, MPC 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 powers of the nodes. We show through analysis that the minimum power configuration of a network is inherently dependent on the data rates of sources. We propose several approximation algorithms with provable performance bounds compared to the optimal solution, and a practical Minimum Power Configuration Protocol (MPCP) that can dynamically (re)configure a network to minimize the energy consumption based on current data rates. Simulations based on realistic radio models of the Mica2 motes show that MPCP can conserve significantly more energy than existing minimum power routing and topology control protocols.

Categories and Subject Descriptors

C.2.2 [**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

Keywords

Sensor Networks, Minimum Power Configuration, Ad-Hoc Networks, Energy Efficiency, Wireless Communications

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.
Copyright 2005 ACM 1-59593-004-3/05/0005 ...$5.00.

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 multi-hop communication in ad hoc networks. *Topology control* [1, 2, 3, 4, 5, 6, 7] aims to reduce the transmission power by adjusting nodal radio transmission ranges while preserving necessary network properties (*e.g.*, connectivity). *Power-aware routing* protocols [8, 9, 10, 11, 12] choose appropriate transmission ranges and routes to conserve energy used for multi-hop packet transmission. Both topology control and power-aware routing focus on reducing the power consumption when the radio interface is actively transmitting/receiving packets. However, such approaches alone are often insufficient because radio interfaces (*e.g.*, Mica2 motes [13] and WLAN cards [14]) also consume non-negligible power even when idle. *Sleep management* [15, 14, 16, 17] has been proposed to reduce the energy wasted in an idle state. A sleep management protocol turns off redundant nodes while only keeping a small number of *active* nodes as relays for multi-hop transmission.

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 the above approaches. However, as we will show in this paper, the correlations between different approaches are dependent on the network workload and hence cannot be combined in a straightforward fashion. For example, when network workload is low, the power consumption of a WSN is dominated by the idle state. In such a case, scheduling nodes to sleep saves the most power. Hence, it is more power-efficient for active nodes to use long communication ranges since it will require fewer nodes to remain awake to relay packets. Conversely, short radio ranges may be preferable when the network workload is high as the radio spends more time in the transmission and reception. In this paper, 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 powers of the nodes.

This paper makes the following key contributions. First, we show through an example that the MPC 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 power consumption in all power states according to the network workload (Section 4). Third, we show that the minimum power configuration problem is NP-hard, and then propose several approximation algorithms with provable performance bounds compared to the optimal solution (Section 5). Fourth, we present a practical distributed Minimum Power Configuration Protocol (MPCP) that can dynamically (re)configure a network based on data rates (Section 6). Finally, we provide simulation results that showed that MPCP can save significantly more energy than minimum power routing protocols. Our simulations are based on realistic radio models (*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. They can be roughly classified into three approaches, namely topology control, power aware routing and sleep management. We summarize the limitations of them after providing a brief overview of the existing works of each approach.

Topology control: Topology control preserves desirable properties of a wireless network (*e.g.*, K-connectivity) through reduced transmission powers. A comprehensive survey on existing topology control schemes can be found in [18]. We review several representative works here. In the scheme proposed in [1], a node chooses to relay through other nodes only when less power is used. The network is shown to be strongly connected if every node only keeps the links with the nodes in its "enclosure" defined by the relay regions. Ramanathan proposed two centralized algorithms to minimize the maximal power used per node while maintaining the (bi)connectivity of the network [2]. Two distributed heuristics were also proposed for mobile networks in [2], although they may not necessarily preserve the network connectivity. Two algorithms are proposed in [4, 3] to maintain network connectivity using the minimal transmission power. CBTC [5] preserves the network connectivity using the minimum power that can reach some node in every cone of 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 [6]. Li et al. proposed a MST-based topology control scheme which preserves the network connectivity and has bounded node degrees [7]. The problem of maximizing network lifetime under topology control is studied in [19].

Power aware routing: Singh et al. proposed five power-aware routing metrics to reduce energy consumption and extend system lifetime [8]. The implementation of a minimum energy routing protocol based on DSR was discussed in [9, 10]. An online power-aware routing scheme is proposed to optimize system lifetime in [20]. Chang and Tassiulas studied the problem of maximizing the lifetime of a network with known data rates [11]. Chang et al. 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 [11]. Sankar et al. formulated maximum lifetime routing as a maximum concurrent flow problem and proposed a distributed algorithm [12].

Sleep management: Recent studies showed that significant energy savings can be achieved by turning wireless radios off when not in use. In this approach, only a small number of nodes remain active to maintain continuous service of a network and all other nodes are scheduled to sleep. ASCENT [15], SPAN [14], AFECA [16] and GAF [17] maintain network connectivity while CCP [21] maintains both network connectivity and sensing coverage. More recently, a sleep schedule algorithm is proposed in [22] to maximize the lifetime of network clustering.

None of the three above approaches optimizes the energy consumption of all radio states. Topology control and power aware routing reduce the transmission power of wireless nodes and do not consider the idle power. Sleep management can reduce the idle power by scheduling idle nodes to sleep, but does not optimize the transmission power. We show in this paper that significant energy reduction can be achieved by jointly optimizing the transmission power and sleep time of nodes based on the network workload.

3. AN ILLUSTRATING EXAMPLE

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

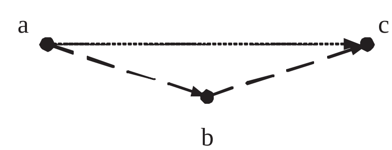

Figure 1: Two communication paths from a to c: $a \rightarrow c$ or $a \rightarrow b \rightarrow c$.

As shown in Fig. 1, a, b and c are three nodes located in 2D space. 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 $|bc|$. The total power consumption of the three nodes under the two configurations, P_1 and P_2, can be computed as follows:

$$P_1 = \frac{R}{B} \cdot P_{tx}(|ac|) + \frac{R}{B} \cdot P_{rx} + 2(1 - \frac{R}{B}) \cdot P_{id} + P_s$$
$$P_2 = \frac{R}{B} \cdot (P_{tx}(|ab|) + P_{tx}(|bc|)) + \frac{2R}{B} \cdot P_{rx} + (3 - \frac{4R}{B}) \cdot P_{id}$$

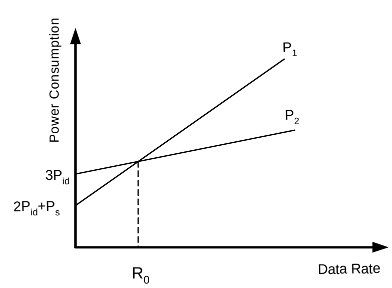

Figure 2: Total power consumption vs. data rate

The total power consumption under each configuration is computed as the sum of the power consumed by three radios in all working modes. For example, P_2 includes the transmission power of node

a and b, $\frac{R}{B} \cdot (P_{tx}(|ab|) + P_{tx}(|bc|))$, reception power of b and c, $\frac{2R}{B} \cdot P_{rx}$ and idle power of the three nodes, $(3 - \frac{4R}{B}) \cdot P_{id}$. For the given radio parameters and node locations, all terms except R are constant in the expressions of P_1 and P_2. We plot P_1 and P_2 in Fig. 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:

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

To get a concrete estimation on R_0, we now apply the radio parameters of Mica2 motes [13] to (1). For a $433MHz$ Mica2 radio, the bandwidth is 38.4 Kbps. There are a total of 30 transmission power levels, each of which leads to 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. It can be calculated that relaying through node b is more power efficient when the data rate is above 16.8 Kbps.

This example leads to the following observations on the power-efficient network configuration: 1) When network workload is low, power consumption of a network is dominated by idle state of radios. In such a case, scheduling nodes to sleep saves the most power. Hence, it is wise to use long radio ranges for communication since more idle nodes in the network can sleep. 2) When network workload is high, the transmission power dominates the total power consumption of a network. As the transmission power increases quickly with distance, communicating with short radio ranges saves more power.

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 modes: *transmitting*, *receiving* and *idle*. The power consumption of an active node is equal to the sum of the power consumption in all working modes. The sleeping power consumption is orders of magnitude lower than active power consumption [13, 14]. In this paper, we only consider the total active power 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, $P_{tx}^{min} \leq P_{tx}(u, v) \leq P_{tx}^{max}$.
2. $G(V, E)$ represents a wireless network. V includes all nodes in the network and E is defined as $E = \{(u, v)|(u, v \in V) \wedge (P_{tx}(u, v) \leq P_{tx}^{max})\}$.
3. P_{rx} and P_{id} represent the power consumption of a node in receiving and idle mode, 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, *e.g.*, periodic data collection, a source is aware of its data rate. Alternatively, a source may estimate its average data rate online. We assume that the total traffic demands are smaller than the bandwidth of any node, *i.e.*, $\sum r_{i,j} \leq 1$ where $r_{i,j}$ represents the data rate between source s_i and sink j normalized by the effective node bandwidth [2]. This assumption is applicable to many sensor network applications with low data rates.

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 power consumption. Before we present the formal definition of MPC problem, we first consider the power consumption of a node assuming 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. 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 power consumption of any active node u (excluding s_* and t_*), $P(u)$, can be computed as the sum of power consumed by transmitting, receiving and idle state at node u:

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

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 power consumption of a node defined by the above equation, MPC problem can be defined as follows.

DEFINITION 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 total power cost $P(G')$ is minimal, where*

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

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

$$\begin{aligned} C_{u,v} &= P_{tx}(u, v) + P_{rx} - 2P_{id} \quad (3) \\ z &= P_{id} \quad (4) \end{aligned}$$

From the above formulation, we can see that an edge (u, v) has a 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 data. We assume that all the data in the same flow takes the same path, *i.e.*, 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 $C_{u,v}$. (2) can be reformulated as follows:

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

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

[2]We assume $\sum r_{i,j} \leq 1/2$ when the communication is multi-hop since the bandwidth of intermediate nodes is consumed by both data transmission and reception.

where $P(s_i, t_j)$ represents the shortest path in $G'(V', E')$ with edge weight $C_{u,v}$. According to (5), the total power cost is equal to the sum of the cost 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 minimum weight steiner tree problem is NP-hard [23]. As a result, a natural reduction from this problem can show that MPC problem is also NP-hard.

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

1. When $S \cup T = V$, *i.e.*, 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 our problem definition, the power consumption of packet retransmissions on lossy communication links is ignored. Recent empirical studies show that lossy communication links are common in real sensor networks [24, 25]. In such a case, the communication quality between two nodes can be quantified by packet reception ratio (PRR) [26]. In this paper, we assume 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 power cost caused by retransmissions, the cost function defined in (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[3]. 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:

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

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

5. CENTRALIZED APPROXIMATION ALGORITHMS

We investigate approximate algorithms for the general MPC problem in the this section. We first focus on the scenario where there is only one sink in the network in this section. 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.

[3]In our design, an acknowledgment is always transmitted at a relatively high power level to reduce the number of retransmissions.

5.1 Matching based Algorithm

When there is only one sink and data flows are not splittable, MPC problem has the same formulation as the *cost-distance* network design problem [27]. Meyerson et. al proposed a randomized approximation scheme [27] that has the best known approximation ratio $O(\lg k)$, where k is 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 an subgraph $G'(V', E')$ that contains the paths from all sources to the sink.

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 non-center 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.

Figure 3: Matching based algorithm (MBA) for MPC problem

The time complexity of the above 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 [27], the algorithm terminates after at most $O(\lg k)$ iterations and the expected cost introduced by the new added edges in each iteration has a constant ratio to the cost of the optimal solution. Hence the approximation ratio of the algorithm is $O(\lg k)$. We refer to this algorithm as *matching based approximation (MBA)* in the rest of the paper.

We note that an edge of G can lie on the matched edges of M in different 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 (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 $D_{u,v} = \frac{2r_i r_j}{r_i + r_j} C_{u,v}$. That is, 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 the 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.

In general, efficient distributed implementation of MBA and MBA-opt is difficult in large-scale sensor networks. In order to find the matching of the network (step 2 of MBA) in a distributed environment, complex coordinations between nodes are needed [28]. Furthermore, MBA and MBA-opt are only applicable to the scenario with one sink node. We next seek more general approximation algorithms that are more suitable to distributed implementations.

5.2 Shortest-path Tree Heuristic (STH)

In this section, we discuss an approximation algorithm called shortest-path tree heuristic (STH). The idea is to balance the flow dependent cost ($r_{i,j} \cdot C_{u,v}$) and the fixed nodal cost (z) 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 the edge.

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
3. Assign edge weights for $G(V, E)$ according to g_i.
4. Find the shortest path connecting s_i to t.
5. Add the shortest path found to $G^{'}$.
6. **end**

Figure 4: Shortest-path Tree Heuristic (STH)

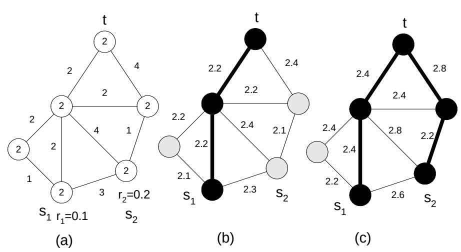

Figure 5: (a) Initial network with edge weight $C_{u,v}$ and node weight $z = 2$ (shown on each node). (b) edge weights are defined by $r_1 \cdot C_{u,v} + z$. (c) edge weights are defined by $r_2 \cdot C_{u,v} + z$. The shortest paths from s_1, s_2 to t are highlighted in black.

Fig. 5 shows an example of STH algorithm. Fig. 5(a) shows an initial network without any flows. Fig. 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 the shortest paths found. According to (2), the total power cost (excluding the cost of the sink) can be calculated to be 9.4.

Step 4 of STH algorithm can be implemented using Dijkstra's 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 MPC problem discussed in Section 4.

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:

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

$w_i(u, v)$ 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 node 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 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 $P(G^{'})$ 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:

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

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 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 the paths from all the sources to the sink). Thus we have:

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

From (10) to (12), we have:

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

□

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 there are already shortest paths established for other sources. Intuitively, this is not efficient since sharing an existing path can lead to lower nodal costs. Suppose we are finding the shortest path from s_i to t and all the shortest paths from $s_j (0 < j < i)$ to t have been found. If any edge on the existing paths is reused by the new path, the additional cost is $r_i \cdot C_{u,v}$ that does not include the nodal cost z since it has been counted by the existing paths. That is, the edge weights on the existing paths should not include the nodal cost z. Based on this observation, we propose the following algorithm called *incremental shortest-path tree heuristic (ISTH)* that finds the minimal incremental cost for each new path. Similar to STH, in each iteration, ISTH finds a shortest path for a new source. We define the states of the nodes on existing paths to be *active*. We define the following set of weight functions for convenience:

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

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** $W \neq \Phi$
5. Choose a random s_i from W and find the shortest path from s_i to t with edge weight $h_i(u,v)$ in $G(V,E)$.
6. Add the shortest path found to $G^{'}$.
7. Label all nodes on the path as *active*.
8. $W = W - s_i$.
9. **end**

Figure 6: Incremental Shortest-path Tree Heuristic (ISTH)

Fig. 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 Fig. 5(b). The total weights on the shortest path from s_1 to t in Fig. 7 are smaller than those in Fig. 5(c) since the nodal cost z is not included. Consequently, different from the case of STH where two paths are disjoint as shown in Fig. 5(c), the shortest path from s_2 to t shares a edge with the existing path. Hence the total number of nodes used become smaller resulting less idle power consumption. According to (2), the total power cost (excluding the cost of the sink) can be calculated to be 7.6 which is smaller than the solution of STH. It can be easily seen that this solution is optimal for this example.

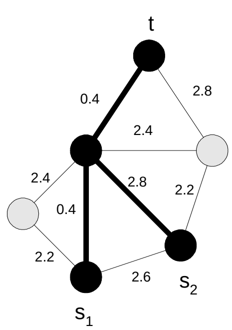

Figure 7: The shortest path from s_2 to t shares a edge with the existing shortest path from s_1 to t.

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

THEOREM 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. $P(G^{'})$ equals the sum of the costs of all shortest paths found by ISTH. We have:

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

According to (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:

$$\sum_i P^{h_i}_{G'}(s_i,t) \quad \leq \quad \sum_i P^{g_i}_{G'}(s_i,t) \qquad (14)$$

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

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

□

As we 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. Hence finding a subsequent shortest path from a source to the sink is equivalent to finding the shortest path to any node on the existing path. In such a case, ISTH follows exactly with a heuristic of minimum weight Steiner tree with approximation ratio of 2 [29]. This result suggests that ISTH yields good performance when the idle power dominates the total power consumption of a network, which occurs when network workload is low or transmission power is small. Similar to STH, ISTH finds the optimal solution for the two polynomial-time special cases of MPC problem.

We have been focusing on the scenario of single sink in this section. As STH and ISTH are based on pairwise shortest-path heuristics, they can be easily extended to the the scenario of 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 before find the optimal solution for two polynomial-time special cases of MPC problem, their known approximation ratio equals the number of source nodes in the network for the general MPC problem. Hence STH and ISTH may not scale well when the number of sources is large. In this section, we seek the algorithm with 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 3. *Let H be the best approximate algorithm to 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 MPC problem.*

PROOF. Suppose $G^{'}_{min}(V^{'}_{min}, E^{'}_{min})$ and $G^{'}(V^{'}, E^{'})$ are the optimal solution 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:

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

Let $P(G^{'})$ and $P(G^{'}_{min})$ represent the cost of $G^{'}$ and $P(G^{'}_{min})$ in MPC problem. We ignore the constant sink node weight z in both $P(G^{'})$ and $P(G^{'}_{min})$, which does not affect the quality of $G^{'}$ or the optimality of $G^{'}_{min}$. We have:

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

where $f(s_i, t)$ represents the shortest path with edge weight $C_{u,v}$ from s_i to t. Based on the assumption $\sum_i r_i \leq 1$, we have:

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

According to (15) and (17), we have:

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

□

The best known approximation ratio to minimum weight Steiner tree problem is about 1.5 [30]. According to the measurements of Mica2 motes, $\alpha \leq 2.3$ [13]. Hence the performance ratio of the approximation scheme discussed in this section is about 5.

5.5 Performance Evaluation

In this subsection, we evaluate the practical performance of MBA, MBA-opt and ISTH under realistic network settings. The performance of the steiner tree based algorithm discussed in Section 5.4 depends on the ratio of maximum transmission power to the idle power, which varies with wireless network platforms. As discussed in Section 5.3, STH is likely worse than ISTH. Hence the steiner tree based algorithm and STH are not evaluated in this section.

We implement MBA, MBA-opt and ISTH in a C++ based network simulator. To evaluate the effectiveness of other power conservation approaches to our problem, we also implement 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 the edge. We choose TMST as a baseline algorithm for performance comparison since distributed MST has been shown to be effective in topology control [7]. Similarly, TSPT finds the shortest path tree of the network weighted by transmission powers, which has been previously proposed as an efficient power aware routing scheme [8].

We use the radio parameters of Mica2 Motes in the simulation. There is no packet loss in the simulation environment. The node bandwidth is 40 Kbps. In the simulation, only the nodes that lie on the communication paths between sources and the sink remain active (*i.e.*, the working mode of their radios is either transmitting, receiving or idle). All non-communicating nodes run in sleeping state. The power consumption of radio in receiving, idle and sleeping modes are $21\ mw$, $21\ mw$ and $6\ \mu w$, respectively [13]. The actual radio range of Mica2 motes varies with environment and

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

Table 1: Radio transmission parameters

transmitting power. We set the parameters of radio range and transmitting powers according to the empirical measurements presented in [31], which is listed in Tab 1. When a node communicates with a neighbor, it always uses the minimum radio range that can reach the neighbor. At the beginning of the simulation, a communication path from each source to the sink is found. All the nodes on the communication paths remain active and all other nodes are put to sleep. The simulation time for each algorithm is 1000 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.

Fig. 8 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 0.2 Kbps. We can see that MBA-opt consumes the least energy among all algorithms, which shows the effectiveness of our optimization to MBA discussed in Section 5.1. ISTH performs slightly worse than MBA-opt but better than MBA, although its known approximation ratio is worse than both MBA-opt and MBA. TSPT and MST lead to considerably more energy consumption than the above algorithms since they only consider transmission power and do not optimize the idle power consumption. Simulations with different node density and data rates show similar results. They are not shown due to space limitation.

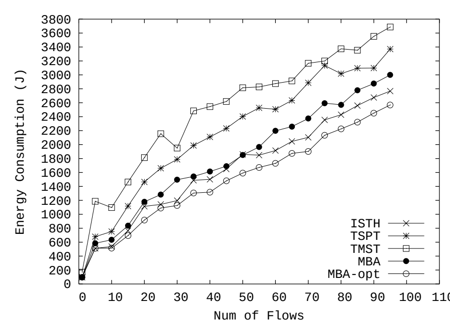

Figure 8: Energy consumption vs. num of flows. The data rate of each flow is 0.2 Kbps.

The results in this section show that the practical performance of ISTH is similar to MBA and MBA-opt. As ISTH is based on shortest-path algorithm, it has more efficient distributed implementation than matching-based MBA or MBA-opt. We now turn our attention to the design of a distributed protocol based on ISTH from the next section.

6. DISTRIBUTED PROTOCOL DESIGN

In this section, we discuss the design of minimum power configuration protocol (MPCP). MPCP can find the power-efficient routes for communicating nodes in a sensor network based on the distributed implementation of ISTH algorithm. We focus on "many

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

Table 2: A routing table

to one" routing scenario in our discussion since it is the most common communication paradigm in sensor networks. MPCP can be easily extended to support more general routing scenarios. In our design, a node operates in either *active* or *power saving* mode. A node in power saving mode remains asleep in most of the time and only periodically wakes up. This simple sleep schedule is similar to several existing power saving schemes such as SMAC [32]. Initially, all nodes operate in power saving mode. When a source node starts sending data to the sink, a power-efficient routing path from the source to the sink is found by 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 power consumption.

Routing mechanisms based on shortest path have been extensively studied previously. We adopt Destination Sequenced Distance Vector Routing (DSDV) protocol [33] as our implementation framework. DSDV is based on distributed implementation of Bellman-Ford shortest path algorithm. A node in DSDV advertises its current routing cost to the sink by broadcasting *route update* messages. A node sets the neighbor with the minimum cost to 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. However, the routing cost of a node in MPCP depends on the operational state (active or power saving) and the data rates of the flows that travel through the node. We now discuss in detail the core components of MPCP.

6.1 Routing Table

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 following fields: $< r_i,\ next\_hop,\ cost,\ seq >$ where r_i is the data rate of source s_i, $next\_hop$ is the neighbor node with the minimum cost to the sink, $cost$ is the cost of node u to the sink through $next\_hop$, seq is a sequence number originated by the sink. Tab. 2 shows a routing table of an active node.

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

6.2 Routing Cost Updates

A node advertises its current routing cost to the sink by broadcasting a route update message to its neighbors. A route update contains a list of data rates and the corresponding routing costs to the sink. After receiving an update from a neighbor, a node calculates its current cost to the sink for each data rate specified in the update, which is equal to the sum of the link cost to the neighbor (defined by (13)) and the cost of the neighbor included in the update. The node broadcasts a route update if the maximum reduction of its costs is above a threshold.

The process of routing cost updates can be triggered by the following events: (1) the quality of a link drops significantly; (2) the data rate of an existing flow changes; and (3) a data flow is started or completed. A node detects (1) when multiple transmissions fail. The process of route updates initiated by (1) is similar to DSDV. We now discuss in detail the route updates caused by (2) and (3).

When a source node changes its data rate to a value that differs 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. To reduce the overhead of route updates caused by case 1), the sink can include several default data rates in its initial route updates. Then only the data rate significantly different from the default ones will cause a round of route updates.

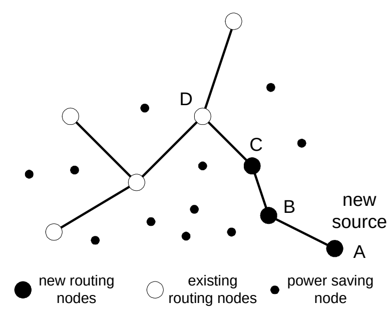

Figure 9: Node A is a new source. The junction node C will initiate a round of route update

Route updates may also be triggered when a new data flow appears. If the new flow has a data rate significantly different from the ones 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 (13)). As shown in Fig. 9, 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 see the reduction of routing costs. In such a case, to reduce the number of route updates, only the 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 Fig. 9, node C will broadcast a route update with a new sequence number and reduced routing costs to initiate a round of route updates. Nodes B, A and other nodes with reduced routing costs to the sink participate in the route update process initiated by C. Note that the 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 no reduction on their routing costs.

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

6.3 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, *e.g.*, those with high packet perception ratios (PRR), in neighborhood tables [24]. A simple way of obtaining the PRR of a link is profiling the link characteristics off-line. Alternatively, the PRR can be obtained from on-line link estimators [24]. For example, nodes can broadcast periodic beacon messages and the PRR of a link to a neighbor can be estimated by counting the number of messages received from that neighbor. Further discussion on this issue is beyond the scope of this paper.

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 range and probabilistic link characterization [25]. The simplifying assumptions on wireless radio propagation made by a network simulator may cause simulation results to differ significantly from real-world experimental results [34]. Accurate simulation to the characterization of real wireless radios with different transmission powers is key for evaluating the realistic performance of MPCP. For this purpose, we implement the link layer model from USC [26] in the Prowler network simulator [35] with improved routing support developed in the Rmase project [36]. Experimental data showed that the USC model can simulate highly unreliable links in the Mica2 motes [26]. 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 [37] in TinyOS. To improve the communication reliability in the lossy simulation environment, we implemented a ARQ (Automatic Repeat Request) scheme that retransmits a packet if an acknowledgment is not received after a 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, which allows us to easily implement new network modules (such as the link model from USC) and to port MPCP to Berkeley motes in future.

7.2 Simulation Settings

For performance comparison, in addition to MPCP, we have implemented two baseline protocols, minimum transmission (MT) routing [24] and minimum transmission power (MTP) routing. They have similar components with MPCP except the cost metrics. MT is shown to be more reliable than hop count based routing scheme in lossy networks [24]. A node in MT chooses the next hop node with minimum expected number of transmissions to the sink. All communication links in the original MT protocol use the same transmission power. A link between node 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 (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 $\frac{P_{tx}(u,v)}{PRR(u,v,P_{tx}(u,v))}$. Besides the consideration for unreliable links, MTP is similar to the minimum power routing schemes studied in [9, 10].

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 improve the hop count from sources. The node bandwidth is 40 Kbps. Power parameters of the radio are set according to the empirical measurements of Mica2 motes [38] as follows. Each node is capable of transmitting data at 10 power levels ranging from -20 dBm to 10 dBm. The corresponding current consumption ranges from 3.7 mA to 21.5 mA. The receiving and idle current is 7 mA. Each simulation lasts for 300 seconds. There is a 60-second initialization phase at the beginning of each simulation, during which all nodes remain active. Every source node starts sending data at a random time instance during the initialization phase. After the 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 active window of one second. The data packet size is 120 bytes. Each source sends a packet every $10 \sim 14$ seconds and the number of sources varies from 5 to 30, which results in a total data rate of 0.3 to 3 Kbps at the sink. Real-world experiments show that the maximum effective bandwidth of Mica2 motes can barely reach 6 Kbps due to channel contention and lossy wireless links [39], which conforms to our observation in simulations. The results in this section are the average of 5 different network topologies.

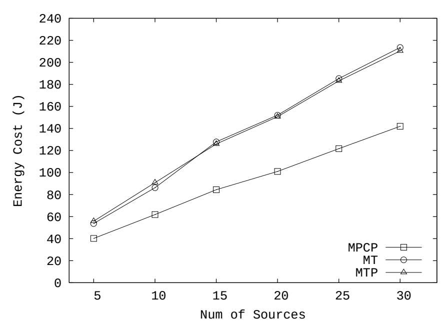

Figure 10: Total network energy cost

7.3 Results

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 the power saving mode. This metric indicates the net energy consumed by a protocol due to the communication activities of the network. As shown in Fig. 10, MPCP consumes the least energy among all protocols. As the number of sources increases, routing paths from different sources share more nodes under MPCP, resulting in more energy reduction in idle state and better energy efficiency. The overall energy reduction of MPCP can be as high as 33%. Interestingly, although MTP optimizes the transmission energy, it has the similar total energy cost to MT that makes simpler routing decisions based on the number of transmissions. As transmission power grows quickly with transmission distance, the routing paths found by MTP likely consist of more hops. Consequently, more nodes have to remain active on the routing paths, resulting in more energy waste due to idle listening. On the other hand, although MT does not optimize transmission power, its routings paths contain fewer hops and hence more nodes can run in power saving mode. In contrast to MTP or MT that only reduces the radio energy costs under partial working modes, MPCP effectively minimizes the total energy cost of radios based on data rates.

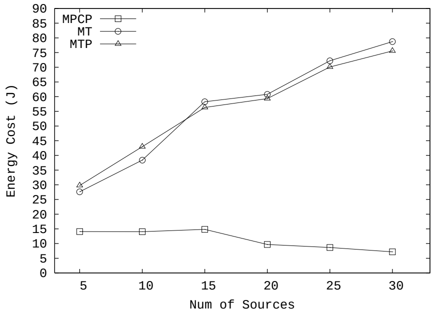

Figure 11: Total energy cost of the communication activities excluding the idle listening of source nodes

We observe that, when the number of source nodes is large, most of the energy consumption is due to the idle listening of the sources. This phenomenon reduces the difference in total energy consumption between different protocols. To focus our analysis on the energy consumption of non-source nodes, we measure the difference between the total energy consumption of the network and that of a network where there is no communication activity, all non-source nodes remain in the power saving mode but all source nodes remain in the idle state. This metric indicates the net energy consumption of the communication activities *excluding* the idle listening of source nodes. Fig. 11 shows that MPCP consumes $50\% \sim 90\%$ less energy than MT or MTP. This result shows that different sources can effectively share intermediate communicating nodes under MPCP. Another interesting result in Fig. 11 is that MPCP may not consume more 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 since they must remain active and have lower costs to the sink. Hence more intermediate nodes may run in the 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.

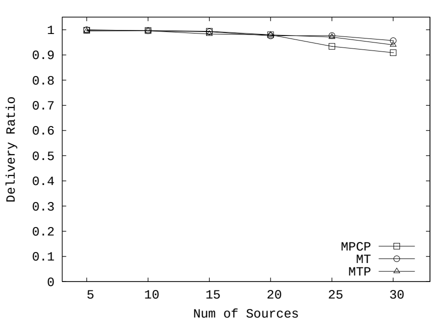

Figure 12: Data delivery ratio at the sink

Fig. 12 shows the data delivery ratio at the sink under different protocols. We can see that the delivery ratio of all protocols decrease slowly when there are more sources in the network. MPCP delivers slightly less data than the other protocols when the number of sources is 30. This is because, when the network workload is high, MPCP causes slightly higher network contention due to path sharing between different sources. However, MPCP can still successfully deliver more than 90% data in all settings.

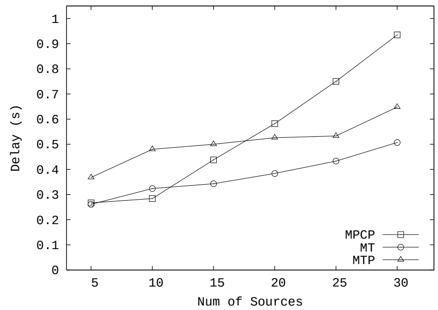

Figure 13: End-to-end data delay

We plot the average end-to-end delay of data packets in Fig. 13. Not surprisingly, MT yields the shortest latency since it finds the routing paths with fewer retransmissions. MPCP performs similarly with MTP when network workload is low but yields higher latency when network workload becomes higher due to network contention caused by path sharing between different sources.

Finally, Fig. 14 shows the number of route update messages of different protocols. The number of route updates of MT and MTP is similar and remains roughly constant as more sources appear. MPCP yields more route updates than the other protocols because the appearance of a new source node changes the node states and routing costs (see (13)), which triggers new route updates. However, consistent to the discussion in Section 6, most route updates are triggered by 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 packets compared with MT and MTP, MPCP still achieves significantly less energy consumption, as shown in Fig. 10 and Fig. 11.

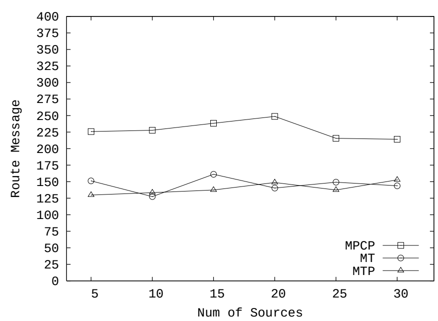

Figure 14: Num of route update messages

The overall simulation results in this section show that MPCP can achieve significant energy reduction and comparable data delivery ratio and latency with MT and MTP. Although MPCP yields more route updates than other protocols, the overhead caused by a new source that joins the network after several initial sources is low.

8. DISCUSSION

In this section, we discuss several limitations of this paper 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 Fig. 10 and Fig. 11 show that further energy savings can be achieved by reducing the idle time of active nodes (*e.g.*, through sleep management). Moreover, MPC problem would be solved optimally if there exists an *ideal* sleep management scheme that schedules an active node to sleep whenever it becomes idle and wakes up an node whenever data arrives. However, the data arrival times can be highly unpredictable in a multi-hop 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 [40], on-demand power management [41], T-MAC [42]) that are adaptive to the network traffic are suitable to work with MPCP to further reduce the idle energy consumption of active nodes.

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 [8, 20]. Finally, while we focus on "many-to-one" workloads, MPCP can be extended to more general workload models with multiple sinks.

9. CONCLUSION

In this paper we propose the minimum power configuration approach to minimize the total power consumption of WSNs. We first formulate the energy conservation problem as a joint optimization problem in which the power configuration of a network consists of a set of active nodes and the transmission ranges of the 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. Simulations based on realistic radio models of the Mica2 motes show that our protocols can conserve significantly more energy than existing minimum power routing and topology control protocols.

Acknowledgement

This work is funded in part by the NSF under an ITR grant # CCR-0325529 and by the DARPA under a contract # F33615-01-C-1904. We thank Prof. R. Ravi at Carnegie Mellon University for his insightful discussion on this paper. We also thank the reviewers for their valuable feedback.

10. REFERENCES

[1] Volkan Rodoplu and Teresa H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Selected Areas in Communications*, vol. 17(8), 1999.

[2] Ram Ramanathan and Regina Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *INFOCOM*, 2000.

[3] Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol," in *European Wireless Conference*, 2002.

[4] Vikas Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in *INFOCOM*, 2003.

[5] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, 2001.

[6] Khaled Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder, "Geometric spanners for wireless ad hoc networks," *IEEE Transactions On Parallel And Distributed System*, vol. 14, May 2003.

[7] Ning Li, Jennifer C. Hou, and Lui Sha, "Design and analysis of an mst-based topology control algorithm," in *INFOCOM*, 2003.

[8] Suresh Singh, Mike Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998.

[9] Sheetalkumar Doshi, Shweta Bhandare, and Timothy X Brown, "An on-demand minimum energy routing protocol for a wireless ad hoc network," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 50–66, 2002.

[10] Sheetalkumar Doshi and Timothy X Brown, "Minimum energy routing schemes for a wireless ad hoc network," in *INFOCOM*, 2002.

[11] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *INFOCOM*, 2000.

[12] Arvind Sankar and Zhen Liu, "Maximum lifetime routing in wireless ad-hoc networks," in *INFOCOM*, 2004.

[13] Crossbow, "Mica2 wireless measurement system datasheet," 2003.

[14] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *MobiCom*, 2001.

[15] Alberto Cerpa and Deborah Estrin., "Ascent: Adaptive self-configuring sensor networks topologies," in *INFOCOM*, 2002.

[16] Ya Xu, John Heidemann, and Deborah Estrin, "Adaptive energy-conserving routing for multihop ad hoc networks," Research Report 527, USC, October 2000.

[17] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *MobiCom*, 2001.

[18] John A. Stankovic, Tarek Abdelzaher, Chenyang Lu, Lui Sha, and Jennifer Hou, "Real-time communication and coordination in embedded sensor networks," *Proceedings of the IEEE*, vol. 91, no. 7, 2003.

[19] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky, "Network lifetime and power assignment in ad-hoc wireless networks," in *ESA*, 2003.

[20] Qun Li, Javed Aslam, and Daniela Rus, "Online power-aware routing in wireless ad-hoc networks," in *MobiCom*, 2001.

[21] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher D. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Sensys*, 2003.

[22] T. Moscibroda and R. Wattenhofer, "Maximizing the lifetime of dominating sets," in *WMAN*, 2005.

[23] Michael R. Garey and David S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1990.

[24] Alec Woo, Terence Tong, and David Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *SenSys*, 2003.

[25] Jerry Zhao and Ramesh Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Sensys)*, Los Angeles, CA, November 2003.

[26] Marco Zuniga and Bhaskar Krishnamachari, “Analyzing the transitional region in low power wireless links,” in *First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*, October 2004.

[27] A. Meyerson, K. Munagala, and S. Plotkin, “Cost-distance: two metric network design,” in *FOCS ’00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[28] Mirjam Wattenhofer and Roger Wattenhofer, “Distributed weighted matching,” in *18th Annual Conference on Distributed Computing (DISC)*, 2004.

[29] H. Takahashi and A. Matsuyama, “An approximate solution for the steiner problem in graphs,” *Math. Japonica*, vol. 24(6), 1980.

[30] Gabriel Robins and Alexander Zelikovsky, “Improved steiner tree approximation in graphs,” in *SODA*, 2000.

[31] Falchi Alessio, “Sensor networks: performance measurements with motes technology,” Tech. Rep., University of Pisa, Italy, 2004.

[32] Wei Ye, John Heidemann, and Deborah Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *INFOCOM*, 2002.

[33] Charles E. Perkins and Pravin Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, 1994.

[34] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott, “Experimental evaluation of wireless simulation assumptions,” in *MSWiM*, October 2004.

[35] G. Simon, “Probabilistic wireless network simulator,” http://www.isis.vanderbilt.edu/projects/nest/prowler/.

[36] Ying Zhang, “Routing modeling application simulation environment,” http://www2.parc.com/spl/projects/era/nest/Rmase/.

[37] Joseph Polastre, Jason Hill, and David Culler, “Versatile low power media access for wireless sensor networks,” in *SenSys*, 2004.

[38] Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoff Werner Allen, and Matt Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *SenSys*, 2004.

[39] Tian He, Sudha Krishnamurthy, John A. Stankovic, Tarek Abdelzaher, Liqian Luo, Radu Stoleru, Ting Yan, Lin Gu, Jonathan Hui, and Bruce Krogh, “Energy-efficient surveillance system using wireless sensor networks,” in *Mobisys*, 2004.

[40] Octav Chipara, Chenyang Lu, and Gruia-Catalin Roman, “Efficient power management based on application timing semantics for wireless sensor networks,” in *International Conference on Distributed Computing Systems (ICDCS)*, 2000.

[41] Rong Zheng and Robin Kravets, “On-demand power management for ad hoc networks,” in *INFOCOM*, 2003.

[42] Tijs van Dam and Koen Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in *Sensys*, 2003.