Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering Research

Computer Science and Engineering

Report Number: WUCSE-2005-34

2005-03-26

Localized and Configurable Topology Control in Lossy Wireless Sensor Networks

Guoliang Xing, Chenyang Lu, and Robert Pless

Recent empirical studies revealed that multi-hop wireless networks like wireless sensor networks and 802.11 mesh networks are inherently lossy. This finding introduces important new challenges for topology control. Existing topology control schemes often aim at maintaining network connectivity that cannot guarantee satisfactory path quality and communication performance when underlying links are lossy. In this paper, we present a localized algorithm, called Configurable Topology Control (CTC), that can configure a network topology to different provable quality levels (quantified by worst-case dilation bounds in terms of expected total number of transmisssions) required by applications. Each node running CTC computes its transmission... Read complete abstract on page 2.

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research

Recommended Citation

Xing, Guoliang; Lu, Chenyang; and Pless, Robert, "Localized and Configurable Topology Control in Lossy Wireless Sensor Networks" Report Number: WUCSE-2005-34 (2005). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/953

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

This technical report is available at Washington University Open Scholarship: https://openscholarship.wustl.edu/cse_research/953

Localized and Configurable Topology Control in Lossy Wireless Sensor Networks

Guoliang Xing, Chenyang Lu, and Robert Pless

Complete Abstract:

Recent empirical studies revealed that multi-hop wireless networks like wireless sensor networks and 802.11 mesh networks are inherently lossy. This finding introduces important new challenges for topology control. Existing topology control schemes often aim at maintaining network connectivity that cannot guarantee satisfactory path quality and communication performance when underlying links are lossy. In this paper, we present a localized algorithm, called Configurable Topology Control (CTC), that can configure a network topology to different provable quality levels (quantified by worst-case dilation bounds in terms of expected total number of transmisssions) required by applications. Each node running CTC computes its transmission power solely based on the link quality information collected within its local neighborhood and does not assume that the neighbor locations or communication ranges are known. Our simulations based on a realistic radio model of Mica2 motes show that CTC yields configurable communication performance and outperforms existing topology control algorithms that do not account for lossy links.

Department of Computer Science & Engineering

Washington University in St.Louis
School of Engineering & Applied Science

2005-34

Localized and Configurable Topology Control in Lossy Wireless Sensor Networks

Authors: Xing, Guoliang; Lu, Chenyang; Pless, Robert

March 26, 2007

Abstract: Recent empirical studies revealed that multi-hop wireless networks like wireless sensor networks and 802.11 mesh networks are inherently lossy. This finding introduces important new challenges for topology control. Existing topology control schemes often aim at maintaining network connectivity that cannot guarantee satisfactory path quality and communication performance when underlying links are lossy. In this paper, we present a localized algorithm, called Configurable Topology Control (CTC), that can configure a network topology to different provable quality levels (quantified by worst-case dilation bounds in terms of expected total number of transmisssions) required by applications. Each node running CTC computes its transmission power solely based on the link quality information collected within its local neighborhood and does not assume that the neighbor locations or communication ranges are known. Our simulations based on a realistic radio model of Mica2 motes show that CTC yields configurable communication performance and outperforms existing topology control algorithms that do not account for lossy links.

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160

Localized and Configurable Topology Control in Lossy Wireless Sensor Networks

Guoliang Xing; Chenyang Lu; Robert Pless

Abstract

Wireless sensor networks (WSNs) introduce new challenges to topology control due to the prevalence of lossy links. We propose a new topology control formulation for lossy WSNs. In contrast to previous deterministic models, our formulation captures the stochastic nature of lossy links and quantifies the worst-case path quality in a network. We develop a novel localized scheme called Configurable Topology Control (CTC). The key feature of CTC is its capability of flexibly configuring the topology of a lossy WSN to achieve desired path quality bounds in a localized fashion. Furthermore, CTC can incorporate different control strategies (per-node/per-link) and optimization criteria. Simulations using a realistic radio model of Mica2 motes show that CTC significantly outperforms an representative traditional topology control algorithm called LMST in terms of both communication performance and energy efficiency. Our results demonstrate the importance of incorporating lossy links of WSNs in the design of topology control algorithms.

keywords: Lossy Sensor Networks, Topology Control, Link Quality; Dilation of Transmission Count; Localized Algorithms

Technical area: Sensor Networks

I. Introduction

Recent years have seen the deployment of wireless sensor networks (WSNs) for a variety of applications such as environmental monitoring, precision agriculture, and perimeter security. The key to the success of these applications lies in the ability of the WSNs to support reliable communication over long periods of time without wired power supplies. Recent empirical studies [1], [2], [3] revealed that the quality of wireless links in WSNs suffer from significant variations with time and environments, which has introduced a major challenge to achieving reliable and power-efficient multi-hop communication. Lossy links can result in severe degradation in communication performance and excessive energy wastage. Zhao et al. [1] reported that a third of the links in a test-bed composed of 60 Mica motes experienced more than 30% packet loss even under light workloads. Consequently, up to 80% of the total energy consumption of the radio was attributed to packet loss [1].

Guoliang Xing is with City University of Hong Kong, E-mail: glxing@cityu.edu.hk; Chenyang Lu and Robert Pless are with Washington University in St. Louis, E-mail: {lu,pless}@cse.wustl.edu. A short version of this paper will appear at the 16th International Conference on Computer Communications and Networks (ICCCN 2007).

Topology control is a key technique to reducing network transmission power while maintaining desired network properties. A multitude of topology control algorithms [4] have been proposed for wireless ad hoc networks. However, WSNs introduce important new challenges that have not been adequately addressed by existing solutions.

Firstly, recent empirical studies [1], [5] revealed the prevalence of lossy and asymmetric links in WSNs. Moreover, receivers with a same distance to a sender experience highly variable reception performance. These findings contradict the widely adopted deterministic link models. Hence, topology control needs to adopt more realistic network models that capture the lossy nature of WSNs.

Secondly, most topology control schemes aim at maintaining connectivity based network properties. However, connectivity alone does *not* suffice to provide satisfactory communication performance when the network is lossy. Communication along a lossy network path may result in excessive packet loss and energy waste. To address the issue of link unreliability, new topology control metrics need to be devised.

Thirdly, different WSN applications require different levels of topology quality in a network. For example, code dissemination requires highly reliable packet delivery in order to ensure consistency among all nodes [6], while sporadic data loss is tolerable for data collection in dense WSNs since sensor data usually has high redundancy [7]. Therefore, topology control must minimize the power consumption of the network while achieving the desired path quality required by the application.

This paper makes the following main contributions. (1) We propose a new formulation of topology control problem for lossy WSNs based on a new metric called *dilation of transmission count (DTC)*. In sharp contrast to earlier metrics based on deterministic link models, DTC captures the stochastic nature of lossy links and quantifies the worst-case path quality of a network topology. (2) We propose a set of novel, localized *configurable topology control (CTC)* algorithms that can achieve different DTC bounds. CTC has three salient features. First, it can provide path quality assurance over lossy and asymmetric links in WSNs. Furthermore, it enables applications to achieve desired tradeoff between transmission power and path quality based on their specific requirements. Finally, it can handle network dynamics efficiently. (3) We conducted extensive simulations based on a realistic link model [8] that captures lossy link characteristics of Mica2 motes. Our results show that CTC significantly outperforms a representative topology control scheme called LMST [9] in terms of delivery rate, data latency and energy consumption.

The rest of the paper is organized as follows. Section II reviews related work. Section III provides a new formulation for the topology control problem in lossy networks. Section IV presents the design and theoretical analysis of the CTC algorithms. Section V discusses how our approach can be extended when the assumption on the monotone link property is relaxed. Section VI presents the simulation results. Section VII concludes the paper.

II. RELATED WORK

Topology control aims at maintaining desirable properties of wireless ad hoc networks (e.g., connectivity and power efficiency). We refer to [4] for a comprehensive survey on the existing topology control algorithms. They fall into two basic classes: per-link control [10], [11], [12], [13], [14] and per-node control [15], [9], [16], [17], [18],

[19]. In per-link control, a node can use different transmission power for different receivers. In contrast, a node in per-node control uses the same transmission power for different receivers. Per-node control simplifies the design of neighbor management and the underlying MAC protocol while per-link control may result in more energy saving.

Compared to earlier algorithms, localized and fault-tolerant topology control schemes are more suitable for lossy WSNs because they are more robust against network dynamics. Several algorithms [20], [21], [22] can mitigate the impact of lossy links by maintaining K-connectivity of the network. While K-connectivity may improve the reliability of a network topology to some extent, it does not provide assurance of path quality because lossy links may exist on multiple paths.

XTC [23] preserves links based on certain ordering of the neighbors. Link quality is one of the ordering metrics. Although XTC assumes a general graph model and constructs topologies with good average spanner property, it does not provide path quality assurance. Moreover, XTC cannot configure a topology to different quality levels required by applications. Recently, a lightweight algorithm called ATPC [24] is proposed to achieve reliable topologies in lossy WSNs. ATPC is designed to maintain per-hop link quality only. It cannot achieve desired path quality over multiple hops, nor can it flexibly configure a network to different quality levels.

Moscibroda et al. [25] studied the limitations of traditional network models and analyzed the impact of topology control on link scheduling based on a physical Signal-to-Interference-plus-Noise-Ratio (SINR) model. In contrast to the previous deterministic graph models, we adopt a network model suitable for lossy WSNs, and propose solutions to handle the impact of network dynamics on topology control.

The metric of dilation of transmission count in this paper is related to the *stretch factor* in graph spanner problems. We refer to [26] for a review of the existing centralized algorithms for constructing graph spanners. Recently, localized algorithms have also been proposed [27], [28], [29]. However, they are only applicable to geometric network models based on circular radio ranges. In contrast, our algorithms are based on a general network model that accounts for lossy and asymmetric links.

III. Problem Formulation

In this section, we first introduce a network model that captures the lossy nature of WSNs. We then provide new formulation of the topology control problem for lossy WSNs.

A. Network Model

Each node can transmit at any power from a discrete set $S = \{P_i | 1 \leq i \leq n\}$. $P_i > P_j \Leftrightarrow i > j$. For example, the CC1000 radio on Mica2 motes [30] can transmit at 32 different power levels. We note that our algorithms in Section IV do not require that all nodes have the same set of tunable power. The *transmission count*, $R_{u,v,i}$, is defined as the expected number of transmissions needed for node u to successfully send a packet to v at power P_i. Note that $R_{u,v,i}$ may not equal $R_{v,u,i}$ due to link asymmetry. The transmission count of a link can be estimated based on the physical or empirical model of the radio [8], [3], [2], or using a link estimator [5], [31] that collects the transmission statistics online. We assume the use of a simple automatic repeat request (ARQ) mechanism at the MAC layer as follows. A sender drops a data packet after T transmissions if no acknowledgement is received.

A *power assignment* $\Omega = \{P_i | P_i \in S\}$ assigns a transmission power for every node in the network if the per-node topology control is used, or for every link if the per-link topology control is used. The network induced by Ω is denoted by a directed graph $G_\Omega(V, E)$. V includes all nodes in the network. $E = \{(u, v, i) \mid R_{u,v,i} \leq T;\ u, v \in V;\ P_i \in \Omega\}$. Note that there exist multiple links from u to v at different power levels. We ignore the links with a transmission count greater than T. The *transmission count of a path* is the sum of the transmission counts of all the links on the path.

We note that the above network model is very general. First, it does not assume deterministic transmission ranges or homogeneous radios. Second, it can capture realistic network properties such as lossy and asymmetric links. Third, it can incorporate empirical measurements (e.g., the transmission count of a link) that reflect dynamic nature of wireless links.

In this paper, we mainly focus on the WSNs that experience little interference or contention caused by concurrent transmissions. Accordingly, we assume that higher transmission power leads to better link quality (and hence a lower transmission count), *i.e.*, $P_i > P_j \Rightarrow R_{u,v,i} < R_{u,v,j}$. This assumption is referred to as *monotone link quality*. This assumption is justified by the fact that higher transmission power alleviates the impact of path fading and noise, which always results in better link quality when the interference is low. This property has been observed in several recent empirical studies on WSNs [32], [31]. Many sensor networks in practice only impose light workload and hence the interference among neighboring nodes is low. For instance, in the WSN deployed at Great Duck Island for habitat monitoring [7], each of the 98 motes wakes up every 20 minutes to send its data to the base station. Many other representative applications (*e.g.*, precision agriculture and cargo tracking) also have low data rates. Furthermore, interference can be eliminated or significantly reduced by scheduling interfering nodes to communicate at different times. For example, TDMA MAC protocols [33], [34], [35] can schedule the channel access of neighboring nodes to avoid contention. Recent interference-aware scheduling algorithms such as [36] and DCQS [37] can schedule transmissions while avoiding both contention and interference in WSNs. Our topology control algorithms will work particularly well for WSNs which has light load and/or uses interference-aware scheduling to minimize network contention and interference. Nevertheless, the monotone link property may not hold temporarily due to dynamics in such networks (e.g., occasional contention and interference may occur when a TDMA scheduling algorithm is in transient sate caused by node failures). We discuss in Section V how to extend our approach when the assumption of monotone link property is relaxed.

Finally, we assume nodes are stationary. Note that most existing WSNs are composed of stationary nodes. We note that the quality of a link may still fluctuate even when nodes are not mobile due to the environmental noise.

B. Topology Control Problems

The problem of topology control has different formulations corresponding to different control strategies and optimization metrics. In this paper, we consider both *per-node* and *per-link* power control strategies. While per-node control assigns each node a single power, per-link control may assign a node different power for different links originating from it. We consider two optimization metrics: min_sum that minimizes the total power of all nodes or links in the network, and min_max that minimizes the maximum power among all nodes or links. The min_max metric may lead to a longer network lifetime by balancing the power consumption of different nodes. We first formulate the problem with per-node control and the min_sum metric, and then extend the formulations to the other cases.

G_M denotes the topology where each node is assigned the maximum power. G_M achieves the best path quality among all topologies under any possible power assignment when the network workload is light. G_Ω represents the topology induced by the power assignment Ω. We define the *dilation of transmission count (DTC)* of G_Ω as the *maximum* ratio of the minimum transmission count between any two nodes in G_Ω to that between the same nodes in G_M. DTC quantifies the worst-case degradation in network's path quality under a power assignment relative to the maximum-power case. This metric closely relates to communication performance like reliability, throughput, and delay. Recent empirical studies [38], [5] showed that transmission count significantly outperforms the hop count in multi-hop routing in lossy wireless networks.

The problem can be formulated as follows when the *min_sum* metric is used. Given a DTC bound $t \geq 1$ specified by the application, the objective is to choose a power assignment Ω with the minimum sum while the DTC bound of the induced topology under Ω is no greater than t:

$$\begin{aligned} \Omega = argmin \sum_{P_i \in \Omega} P_i,\ subject\ \ to \\ \max_{u,v \in V} \frac{\Gamma_{G_\Omega}(u,v)}{\Gamma_{G_M}(u,v)} \leq t \end{aligned} \tag{1}$$

Note that Ω may include a transmission power for each node or link depending whether the the per-node control or per-link control strategy is used. $\Gamma_{G_X}(u,v)$ denotes the minimum transmission count from u to v in the network under power assignment X. When the metric is *min_max*, the minimization objective in the above formulation needs to be replaced by $\max_{P_i \in \Omega} P_i$.

As discussed in Section II, per-node control simplifies the design of neighbor management and the underlying MAC protocol while per-link control may result in more energy saving. The *min_max* metric can minimize the total network power consumption while the *min_max* balance the power consumption of different nodes and may lead to better network lifetime. Our algorithm can be easily configured to perform different control strategies and minimization metrics. This feature allows the application flexibly configure the behavior of topology control to meet its needs.

The network topologies under the above formulations can improve the performance of several representative routing approaches. It has been shown in [5], [38], [39] that expected transmission count outperforms hop-count-based routing metrics in terms of reliability, delay and throughput when links are lossy. Therefore, the network topology with bounded DTC allows the transmission-count-based routing protocols [5], [38], [39] to achieve desired performance in lossy WSNs. In addition, our formulations preserve the power-efficient routes in a lossy network, which allows power-aware routing protocols to minimize the energy wasted by packet retransmissions. Finally, the network topologies under the *min_max* formulations can enhance the capability of power-aware routing protocols to extend network lifetime via load balancing.

C. Centralized Solutions

We now discuss possible centralized solutions for our problems and their hardness.

*a) **Solutions for the*** **min_max** ***Formulations.**:* When the metric is *min_max*, both the per-link and per-node topology control can be solved optimally in polynomial time. According to the monotone link property, power increase of a link or a node does not result in the increase of the network DTC, which implies that there must exist an optimal solution in which all nodes or links are assigned the same power when the maximum power among all nodes (per-node control) or links (per-link control) is minimized. Hence an optimal power assignment can be found through a binary search on all possible power levels, which can be done in $O(log|S|)$ time. For each power level, the DTC of the network can be computed in $O(|V| \cdot |E| \cdot log|V|)$ time using the Dijkstra's algorithm [40]. Hence the minimum power level that yields the required DTC bound can be found in $O(|V| \cdot |E| \cdot log|V| \cdot log|S|)$. However, the resultant topology may unnecessarily waste energy as all nodes are forced to have the same transmission power. Minimizing the number of the maximum-power nodes in the optimal power assignment is NP-hard [19]. Moreover, such a strategy can not be implemented in a localized fashion because excessive synchronization and information exchange would be needed among nodes in order to find a uniform power for all nodes under the DTC constraint.

*b) **Hardness of the*** **min_sum** ***Formulations**:* When the metric is *min_sum*, both the per-node and per-link topology control problems are NP-hard. Specifically, the special case of the *min_sum* per-node control problem where the transmission count of all links is one and the required DTC bound is large is equivalent to minimizing the total node power while achieving strong network connectivity. This problem has been shown to be NP-hard [41].

An NP-hard special case of the *min_sum* per-link control problem can be constructed as follows. For every two nodes u and v in the network, add an edge from u to v (with transmission count one) if there exists a power level $P_{u,v}$ at which u can reach v. Let $P_{u,v}$ be the weight of the edge. Then our problem becomes, for a given graph and a constant $t > 1$, find the subgraph with the minimum total edge weights under that constraint that the shortest path between any two nodes is no longer than t times of that in the original graph. This problem has been studied as the minimum weight t-spanner and been proven to be NP-hard. It was shown in [42] that it is hard to find $O(log|V|)$ approximations for this problem.

In this paper, we propose a set of localized solutions which only require the information of each node's local neighborhood. Localized algorithms are highly desirable in WSNs whose topologies may change dynamically due to node/link failures and fluctuations of link quality [2]. Centralized solutions can be prohibitively expensive in such WSNs due to the need for gathering the topology and link quality information of the whole network.

IV. The Localized CTC Algorithms

In this section, we present a set of localized Configurable Topology Control (CTC) algorithms. The key challenge for the design of CTC is to achieve the required DTC bound on the *global* network topology quality in a localized fashion. We first introduce the following theorem.

Theorem 1: The network topology G_Ω induced by power assignment Ω has a DTC bound t if for each link (x, y, i) in topology G_M where each node is assigned the maximum power, G_Ω contains a path from x to y whose total transmission counts is no higher than t times of the transmission count of the link. That is:

$$\forall (x, y, i) \in E(G_M),\ \Gamma_{G_\Omega}(x, y) \leq t \cdot R_{x,y,i} \Longrightarrow \max_{u,v \in V} \frac{\Gamma_{G_\Omega}(u, v)}{\Gamma_{G_M}(u, v)} \leq t \tag{2}$$

where $\Gamma_{G_X}(u, v)$ denotes the minimum total transmission counts from u to v in the network topology induced by power assignment X.

Proof: Suppose Υ represents the shortest path (in term of transmission count) from node u to v in G_M: $u = u_0, u_1, u_2, \cdots, u_{n-1}, u_n = v$. For each link (u_l, u_j, k) (where k is the transmission power level of u_l) on Υ, the total transmission counts of the shortest path from u_l to u_j in G_Ω must be lower than $t \cdot R_{x,y,i}$. Hence, concatenating such path in G_Ω for each link on Υ results in a path no longer than t times of the total transmission counts of Υ. Since this holds for every pair of two nodes in the network, the DTC bound of G_Ω is no more than t. ■

According to Theorem 1, CTC achieves the DTC bound by replacing each max-power link with a low-power path that has a bounded transmission count relative to the replaced link. This strategy can be implemented in a localized fashion since a replacement path is likely located within the neighborhood of the replaced link in a dense network. However, the challenge is to ensure the replacement paths found by different nodes are consistent. The *key feature* of CTC is that it ensures this consistency in a localized fashion without any decision exchange among neighboring nodes.

We first describe the concept of neighborhood used by CTC. We then illustrate the basic idea of CTC using an example, followed by the detailed description of CTC. Finally we present the theoretical analysis of CTC and describe extensions to CTC for handling several practical issues in WSNs.

A. Neighborhood

CTC uses a two-hop neighborhood graph that is constructed from link quality information. Node v is node u's one-hop neighbor if there exists at least one link, (u, v, i) where $P_i \in S,\ R_{u,v,i} \leq T$, from u to v. The one-hop neighborhood graph of u includes u and all the one-hop neighbors of u, and all the links from u to its neighbors. The two-hop neighborhood graph of node u is the union of the one-hop neighborhood graphs of u and u's neighbors. We use $N_i(u) = (V_i(u), E_i(u))$ $(i = 1, 2)$ to denote the one-hop and two-hop neighborhood graphs at u.

Although links may be asymmetric, we require the neighborhood relation to be symmetric, *i.e.*, $(u, v, i) \in E_1(u) \Leftrightarrow (v, u, j) \in E_1(v)$. Each node u can enforce this requirement by pruning the links to the neighbors who do not include u within their one-hop neighborhood.

In order to construct $N_2(u)$, node u needs to collect the transmission counts of the links within its two-hop neighborhood at different power levels. Each node can measure transmission counts of its one-hop links based on data or hello messages, and exchange them with its one-hop neighbors. Efficient algorithms for neighborhood discovery and link quality estimation have been proposed in earlier work [5], [31] and is not the focus of this paper.

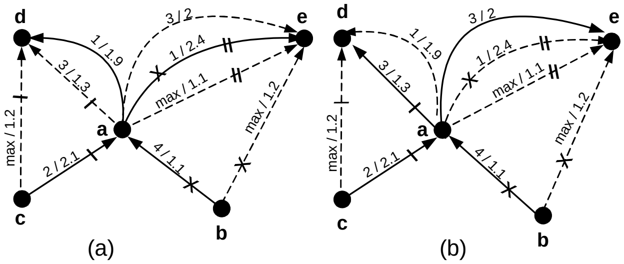

Fig. 1. The execution of two algorithms with a required DTC bound of 3. (a) illustrates a naive algorithm in which each node only replaces its own max-power links. (b) illustrates the CTC algorithm with the min_sum metric. Each link is labeled by *power / transmission count*. max represents the maximum transmission power. Solid links represent the actual links after the execution of the algorithm. The max-power links and their corresponding replacement paths are labeled by the same symbols.

B. An Illustrative Example

We now illustrate the basic idea of CTC using a example depicted in Fig. 1. We will describe how CTC is executed at three nodes a, b, and c when per-node control and the min_sum metric are used. For clarity, Fig. 1 only shows a subset of the links that exist between nodes $a \sim e$. The DTC bound required by the application is 3.

We first describe a naive localized algorithm that may result in conflicting power assignments. Each node in this algorithm independently replaces each of the max-power links that originate from it with a low-power path whose transmission count satisfies the DTC bound. Fig. 1(a) depicts a possible output after the executions at a, b, and c. Node b replaces the max-power link (b, e, max) with path $(b, a, 4) \rightarrow (a, e, 1)$. The transmission count of the new path is $1.1 + 2.4 = 3.5$, which is lower than triple of that of (b, e, max). Similarly, nodes a and c replace (a, e, max) with $(a, e, 1)$, and (c, d, max) with $(c, a, 2) \rightarrow (a, d, 3)$, respectively. Notice that a is assigned two different power, 3 and 1, on the three replacement paths. If each node sets its power independently according to the replacement paths it finds, a will choose a power of 1 as it is not aware of the existence of the other replacement paths. As a result, the actual quality of the link from a to e is lower than required by the path found by c. Consequently, the path from c to d has a dilation of $(2.1 + 1.9)/1.2 = 3.3$ that violates the required DTC bound of 3. This problem is caused by the inconsistency of the local paths found by different nodes. An simple solution is to have nodes exchange their local solutions with their neighbors. However, such solution is not desirable due to the communication overhead and convergence latency.

We now discuss how CTC solves this problem. The basic idea is that, in addition to replacing its own max-power links, each node also computes its power assigned by its neighbors on their local paths. As a result, it always chooses a power no lower than any power assigned by itself and its neighbors, which preserves the dilation of all replacement paths.

Specifically, a node finds a replacement path for each max-power link in its two-hop neighborhood. The replacement path must yield the minimum total power among all possible paths that satisfy the dilation constraint. For instance, the replacement path of (b, e, max) is $(b, a, 4) \rightarrow (a, e, 1)$, which has the minimum total power among all paths from b to e with a dilation no greater than 3. Node a starts with the lowest power, and once finds a new replacement path that includes itself, it increases its power to match its power assigned on the path if necessary. As shown in Fig. 1(b), node a first assigns itself a power of 1 after replacing (a, e, max) and (b, e, max), and then increases its power to 3 after finding the replacement path for (c, d, max). As a result, all replacement paths are preserved after a executes the algorithm.

We can see from Fig. 1(b) that all the nodes on a replacement path find the same path when they replace the same max-power link. For example, the path $(c, a, 2) \rightarrow (a, d, 3)$ is found by both c and a to replace (c, d, max) in their local executions. As a result, the dilation of the path is preserved as a and c will assign their power no lower than the values on the path. We offer a rigorous proof of the correctness of a generalized algorithm in Section IV-E.

C. Per-node Power Control

We now present CTC with per-node control. We first describe the algorithm with the min_sum metric, and then discuss how it can be modified to adopt the min_max metric. For each max-power link, CTC finds a replacement path composed of up to d low power links in the node's two-hop neighborhood. d is referred to as *search depth* hereafter. A larger search depth increases the opportunity for CTC to find lower power assignments at the cost of higher computation complexity.

CTC executed at node u with the min_sum metric is depicted in Fig. 2. To enforce consistent power assignments on the replacement paths found by different nodes, u invokes the function *LabelSet(v)* for each node $v \in V_1(u)$ including itself. In doing so, u essentially "simulates" the execution of the algorithm at all nodes within its one-hop neighborhood. Function *LabelSet(v)* finds the replacement paths with DTC bound t for all the max-power links that originate from v. Special care needs to be taken at this step since a node has different neighborhood view from its neighbors. The key is that if a node lies on a replacement path found by its neighbors, it should also find the same path in its own execution of CTC. Once u finds a replacement path that includes itself, it increases its power to match its power assigned on the path if necessary.

The function *LabelSet* extends the Generalized Permanent Labeling Algorithm (GPLA) [43] for the shortest path problem with time window (SPPTW). A special case of SPPTW, the weight-constrained shortest path (WCSP) problem, resembles our problem. Each link in a WCSP problem has two weights in different metrics. The goal is to find the shortest path between two nodes in terms of one weight metric under the constraint that the total weights of the other metric is bounded. The power and transmission count of a local path correspond to the two different weight metrics in a WCSP problem.

LabelSet(v) extends GPLA in several important aspects. First, while GPLA finds a single best path between two nodes, *LabelSet(v)* finds the best replacement paths from v to all its neighbors. Second, a set of constraints are added in the search process to ensure that different nodes will find consistent replacement paths for the same max-power link. As shown in Section IV-E, this property is important for ensuring the correctness of CTC. Finally, in addition to minimizing the total power of a replacement path, we also extend GPLA to incorporate other optimization metrics like min_max.

```
Input: $t$, $d$, $N_1(u)$, $N_2(u)$
Output: $power(u)$

$power(u) = min$;
for $v \in V_1(u)$   call $LabelSet(v)$;   end

function LabelSet(v)
  1) $W = t \cdot \max\{R_{v,w,max} | (v,w,max) \in E_1(v)\}$. Set $L_v = \{(0,0)\}$ and $L_i = \emptyset$ for all $i \in V_1(v) - \{v\}$.
  2) If all labels have been marked, go to 5); else choose $i \in V_1(v)$ that has an unmarked label $(R_i^q, P_i^q)$ with minimal $R_i^q$.
  3) For each link $(i,j,k) \in E_2(u)$ do
     $L_j = L_j \cup \{(R_i^q + R_{i,j,k}, P_i^q + P_k)\}$, if the following conditions are met:
        $R_i^q + R_{i,j,k} \leq W$                                   (3)
        $|q| < d$                                                    (4)
        $j \in \bigcap_{k \in V_1(v)} V_1(k)$                        (5)
        $\nexists (R_j^q, P_j^q) \in L_j,$
        $(R_j^q \leq R_i^q + R_{i,j,k}) \wedge (P_j^q \leq P_i^q + P_k)$  (6)
  4) Mark label $(R_i^q, P_i^q)$. Go to step 2.
  5) For each link $(v,w,max)$ in $E_1(v)$, do:
     a) Find the label $(R_w^q, P_w^q)$ in $L_w$ such that $R_w^q \leq t \cdot R_{v,w,max}$ and has the minimal $P_w^q$.
     b) If there exists a $u$'s link $(u,z,k) \in q$ and $power(u) < P_k$, $power(u) = P_k$.
```

Fig. 2. The per-node CTC executed at u with the min_sum metric.

LabelSet(v) is a dynamic programming procedure in which the partial paths found are stored by *labels* on nodes. Specifically, a label on node i is a tuple (R_i^q, P_i^q) where q corresponds to a path from v to i, and R_i^q and P_i^q are the transmission count and total power of the path respectively. Such a path is a candidate replacement path for the max-power link from v to i, and can also be a partial path on the replacement paths for the links from v to other neighbors. L_i represents the set of labels on i that corresponds to all such partial paths.

The procedure starts by initializing v's label set to $\{(0,0)\}$ and all the label sets on other nodes to be empty. Then the algorithm executes in iterations. In each iteration (composed of step 2 to 4), an existing label (R_i^q, P_i^q) with minimum transmission count is extended along all outgoing links of node i, which corresponds to extending the partial path q to all possible next-hop nodes (step 3). The label is *marked* after all next-hop nodes are examined (step 4). The search process initiated from v terminates if all labels on the nodes within $V_1(v)$ have been marked. Step 3 extends label (R_i^q, P_i^q) along a link (i,j,k) by adding the transmission count and power of (i,j,k) to R_i^q and P_i^q respectively. The link will be added to the label set of j, if the constraints (3)-(6) are met.

Constraint (3) requires that the total transmission count of the expanded path must be smaller than W which is t times the maximum transmission count of all the max-power links originated from v. This constraint reduces the search space by eliminating the paths that would have a dilation higher than t. Constraint (4) limits the maximum hop count of a path to d. Constraint (5) enforces that all nodes on a path must be located within one hop of each other. As shown in Section IV-E, this constraint is critical for ensuring the consistency in the power assignments computed by different nodes.

Constraint (6) ensures that there does not exist a label on the next-hop node that represents a better path than the extended path. A path X is better than path Y if and only if X has a lower transmission count *and* lower power than Y. If (6) does not hold, we keep the paths with higher power but lower transmission count, or the paths with higher transmission count but lower power, since both types of paths may satisfy constraint (3) and evolve into valid replacement paths in following iterations. It can be seen that this property allows *LabelSet* to find the *optimal* replacement path (e.g., with the minimum total power) under constraints (3)-(5).

At the end of the procedure, for each max-power link (v, w, max), the replacement path is the path that has the minimum total power among all paths that satisfy the dilation constraint (see step 5.a). Note that such a path must exist since in the worst case the max-power link (v, w, max) will be found. Finally, if node u (that executes the algorithm) lies on the replacement path, it sets the power to the max of its current power and the power on the path.

Minimizing the maximum power on a replacement path may lead to more balanced power on different nodes. We modify CTC depicted in Fig. 2 as follows to adopt the min_max metric. In a label (R_i^q, P_i^q), instead of storing the total power of path q in P_i^q, we redefine P_i^q as the maximum power of the links on q. Accordingly, constraint (6) needs to be changed to $\nexists (R_j^q, P_j^q) \in L_j,\ (R_j^q \leq R_i^q + R_{i,j,k}) \wedge (P_j^q \leq \max(P_i^q, P_k))$.

D. Per-link Power Control

Different from per-node control that restricts a node to a fixed power, per-link control allows a node to use different power to transmit to different neighbors. As a result, per-link control may lead to more energy saving. An advantage of the algorithm depicted in Fig. 2 is that it can be easily modified to use per-link control. Specifically, node u stores a power value $power(u, v)$ with an initial value of minimum power for each of its one-hop neighbors, $v \in V_1(u)$. In addition, step 5.b needs to be modified as follows: If there exists u's link $(u, z, k) \in q$ and $power(u, z) < P_k$, $power(u, z) = P_k$. Notice that both per-node and per-link control share the same procedure for searching replacement paths (step 1 to 4 of function *LabelSet* in Fig. 2). Hence, the same modification introduced in Section IV-C can also be used to adopt different optimization metrics, including min_sum and min_max, in per-link control.

E. Correctness of CTC

We now prove the correctness of CTC. We first show that CTC with per-node control and the min_sum metric achieves the required dilation bound. We then extend this result to per-link control and the min_max metric.

Theorem 2: Suppose M is the power assignment where each link is assigned the maximum power, Ω is the power assignment produced by the CTC algorithm with a DTC bound $t \geq 1$. Then the network G_Ω satisfies the DTC bound t: $\max_{u,v\in V} \frac{\Gamma_{G_\Omega}(u,v)}{\Gamma_{G_M}(u,v)} \leq t$.

Proof: To prove the theorem, it suffices to show that any link in G_M, say (v, w, max), is replaced by a path in G_Ω with a dilation no greater than t. We prove that this holds after the execution of CTC at each node.

Suppose v finds a replacement path $F^v_{v,w}$ for (v, w, max). Note that $F^v_{v,w}$ corresponds to the label (R^q_w, P^q_w) found by v at step 5.a. According to step 5.a, $F^v_{v,w}$ must have a dilation no greater than t. Hence, it remains to be shown that this path is preserved by the power choices made by other nodes on the path in their executions of CTC. Suppose (x, y, i) is an arbitrary link on path $F^v_{v,w}$. That is, v assigns power P_i to x. In the following we will show that the replacement path for (v, w, max) found in node x's execution of CTC, $F^x_{v,w}$, is exactly $F^v_{v,w}$, and hence the power of x is no lower than the power assigned on $F^v_{v,w}$.

We define graph $G^x(v, w) = (V^x(v, w), E^x(v, w))$ as follows.

$$\begin{aligned} V^x(v,w) &= \bigcap_{k\in F^x_{v,w}} V_1(k) \\ E^x(v,w) &= \bigcup_{(a,b\ \in V^x(v,w))\wedge((a,b,i)\in E_1(a))} (a,b,i) \end{aligned}$$

$G^x(v, w)$ includes all the nodes shared by the one-hop neighborhoods of the nodes on path $F^x_{v,w}$. Hence $G^x(v, w) \subseteq N_2(v)$. In other words, all the replacement paths found by x for (v, w, max) are included in the two-hop neighborhood of v. On the other hand, according to step 5.a of CTC in Fig. 2, $F^x_{v,w}$ is the *optimal* replacement path (in terms of total power) among all possible paths within $N_2(v)$ that satisfy the dilation bound and constraints (3) and (4). As a result, node x chooses $F^v_{v,w}$ as the replacement path for (v, w, max) within its execution of CTC. That is, $F^x_{v,w} = F^v_{v,w}$ and hence the power of x decided by x itself is the same as assigned by v on path $F^x_{v,w}$.

We have shown that each replacement path found by v for (v, w, max) is preserved after all the nodes on the path compute their power assignments in their local executions of CTC. That is, each max-power link is replaced by a path with a dilation no greater than t after the execution of CTC at each node. Therefore, the resultant network has a DTC no greater than t. ■

We note that similar arguments can prove the correctness of CTC with per-link control or the min_max metric. This is because, the nodes on a replacement path will find the same path as long as the the path is optimal (in terms of the min_sum or min_max metric) within the two-hop neighborhood of the originator of the link.

F. Time Complexity of CTC

We now analyze the time complexity of CTC. Suppose the number of links in each node's two-hop neighborhood is bounded by $|E_2|$. Procedure $LabelSet(v)$ without constraints (4) and (5) is similar to the original GPLA algorithm that has a complexity of $O(|E_2|W)$ where W is t times the maximum transmission count from v to its one-hop neighbors. Since we only keep the labels that satisfy constraint (6), there is at most one label kept for each value of transmission counts. That is, a node has at most W labels. Hence, in step 2, a link is processed at most W times. Summing the number of times an link is processed over all links gives a time complexity of $O(|E_2|W)$. We note that this complexity is pseudo-polynomial as it depends on W.

On the other hand, the actual time complexity of $LabelSet(v)$ is lower due to the constraints (4) and (5) in Fig. 2. Specifically, (4) requires the number of hops of a path to be smaller than d. Suppose the number of nodes within a one-hop neighborhood is bounded by $|V_1|$, the total number of link processing in $LabelSet$ is bounded by $O(|V_1|^{d-1})$. Hence the time complexity of *LabelSet* is bounded by $O(min(|V_1|^{d-1}, |E_2|W))$. Since *LabelSet* is invoked for each one-hop neighbor, the overall time complexity of the generalized CTC algorithm is $O(|V_1| \cdot min(|V_1|^{d-1}, |E_2|W))$. We note that this complexity result is an upper bound, which does not consider constraint (5). Although this bound is exponential in d, we show experimentally that small search depth, (*e.g.*, choosing $d = 2$ or 3) gives a very good performance in Section VI.

G. Extensions

We now discuss extensions to CTC that can deal with several practical issues in WSNs.

1) Handling node and link dynamics: In a real-world WSN, nodes and links often exhibit various dynamics that may cause the network topology to violate the dilation bound. We now discuss how CTC can handle three important types of dynamics: node failure, link failure, and link quality variation. Thanks to its localized nature, a key advantage of CTC is that it can maintain required DTC bound via *local* repair in face of network dynamics.

CTC can detect node failure and link changes based on hello messages used for neighborhood maintenance and link quality estimation. Alternatively, CTC may be notified on demand by the feedback from the MAC layer (e.g., successive transmission failures on a link). In order to preserve the DTC bound for the network when a node fails, only the nodes within one hop from the failed node need to execute CTC again with the updated neighborhood information. This is because, as discussed in Section IV-E, all the nodes on a replacement path are one hop from each other. Therefore, only one-hop neighbors of the failed node need to recompute their replacement paths. That is, a node failure only requires local repair to the network topology. This feature allows CTC to scale effectively for large-scale WSNs. Similarly, when the link from u to v fails or experiences quality degradation, only the one-hop neighbors of u need to rerun CTC to maintain the DTC bound.

The link from u to v may also experience quality increase due to reduced environmental interference, or a higher power assignment of u after rerunning CTC for a local repair. In such a case, the neighbors of u rerun CTC to lower their power assignments only if the link quality increase exceeds a threshold. The threshold reduces the propagation of power reassignments and should be determined based on the desirable trade-off between topology stability and power saving. We note that such propagation of power reassignments is needed only for power optimizations. It is not needed for preserving the DTC bound, which can be achieved via local repair.

A more efficient mechanism to handle node and link dynamics is possible at higher storage cost. Each node can store the replacement paths for each max-power link in its two-hop neighborhood, and only update the affected replacement paths in presence of link or node failures. The storage cost is $O(|E_1| \cdot d)$ where E_1 are d are one-hop neighborhood and the search depth of CTC, respectively.

2) Integration with sleep management: CTC aims at reducing transmission power consumption of a network. Another significant source of power consumption is idle listening. CTC can be combined with a sleep management protocol to minimize the energy consumed by both transmission and idle listening. Existing sleep management schemes fall into two basic classes: backbone maintenance and sleep scheduling. A backbone maintenance protocol constructs a backbone composed of a small number of active nodes and schedules the other nodes to sleep. The active nodes on the backbone can run CTC to reduce the transmission power consumption and achieve bounded dilation on the backbone topology while other nodes can reduce the idle listening power consumption through sleeping. In a sleep scheduling protocol, each node operates in a schedule composed of active and asleep intervals. In such a case, each node can run CTC to reduce the power consumed for packet transmissions during the active intervals.

```
Input: $t$, $d$, $N_1(u)$, $N_2(u)$
Output: $power(u,v)$ $(v \in E_1(u))$

$power(u,v) = min$;
for $v \in V_1(u)$
    $T(v) = \{(v,w,i) | i = \max_{(v,x,j) \in E_1(v)} j\}$
    call $LabelSet(v)$
end

function LabelSet(v)
  1) $W = t \cdot \max\{R_{v,w,i} | (v,w,i) \in T(v)\}$. Set $L_v = \{(0,0)\}$ and $L_i = \emptyset$ for all $i \in V_1(v) - \{v\}$.
  2) If all labels have been marked, go to 5); else choose $i \in V_1(v)$ that has an unmarked label $(R_i^q, P_i^q)$ with minimal $R_i^q$.
  3) For each link $(i,j,k) \in E_2(u)$ do
     $L_j = L_j \cup \{(R_i^q + R_{i,j,k}, P_i^q + P_k)\}$, if the following conditions are met:
         $R_i^q + R_{i,j,k} \leq W$
         $|q| < d$
         $j \in \bigcap_{k \in V_1(v)} V_1(k)$
         $\nexists (R_j^q, P_j^q) \in L_j,$
         $(R_j^q \leq R_i^q + R_{i,j,k}) \wedge (P_j^q \leq P_i^q + P_k)$
  4) Mark label $(R_i^q, P_i^q)$. Go to step 2.
  5) For each link $(v,w,i)$ in $T(v)$, do:
     a) Find the label $(R_w^q, P_w^q)$ in $L_w$ such that $R_w^q \leq t \cdot R_{v,w,i}$ and has the minimal $P_w^q$.
     b) If there exists a $u$'s link $(u,z,k) \in q$ and $R_{u,z,i} > R_{u,z,k}$, $power(u,z) = P_k$.
```

Fig. 3. The extended per-link CTC (with the min_sum metric) for non-monotone link property.

V. Relaxing the Assumption of Monotone Link Property

In this work we mainly focus on WSNs that do not experience significant interference as a result of light workload and/or TDMA scheduling techniques. Accordingly, we assume the monotone link property, i.e., the transmission count of a link decreases with the transmission power. However, such a property may not hold due to network dynamics such as the occasional network contention in the transient state of a TDMA scheduling algorithm. We now discuss how per-link CTC can be extended when the assumption of monotone link property is relaxed.

When the monotone link property does not hold, the definition of DTC bound relative to the maximum power topology needs to modified. Suppose G represents the network graph in which there exist multiple links between two nodes corresponding to the communication links using different transmission power. The weight of each link from node u to node v at transmission power P_i is the transmission count $R_{u,v,i}$. Suppose $G_{opt} \subseteq G$ represents the shortest-path spanning tree of G. Obviously G_{opt} is the *optimal* network topology in term of transmission counts. G_Ω represents the network graph under power assignment Ω where each link is assigned a power. We redefine the DTC of G_Ω as the *maximum* ratio of the minimum total transmission counts between any two nodes in G_Ω to that between the same nodes in G_{opt}. Then the per-link topology control problem can be formulated as follows:

$$\begin{aligned} \Omega = argmin \sum_{P_i \in \Omega} P_i,\ subject\ \ to \\ \max_{u,v \in V} \frac{\Gamma_{G_\Omega}(u,v)}{\Gamma_{G_{opt}}(u,v)} \leq t \end{aligned} \tag{7}$$

The above formulation assumes per-link topology control and cannot be easily extended to the case of per-node control. This is because, all the links originated from a node are assigned the same power under per-node control, hence the change of a node's power may increase the transmission count of one network path and decrease another at the same time. In other words, when the monotone link property does not hold, the optimal topology that contains the shortest paths among all nodes in the network may not exist. Seeking appropriate formulation for the per-node control in such a case is left for future work.

Per-link CTC can be extended as follows to accommodate the new problem formulation. Node u creates a set $T(v)$ for each of its one-hop neighbors v including itself. $T(v)$ includes a link from v to each one-hop neighbor w that has the minimum transmission count among all links from v to w. Then node u invokes function $LabelSet(v)$ to find a low-power replacement path for each link in $T(v)$. The modified CTC algorithm is shown in Fig. 3. The function $LabelSet(v)$ is similar to the one shown in Fig. 2 because the dynamic programming procedure used to find the replacement paths does not assume any relationship between the transmission power and the corresponding transmission count of a link. The major difference is in step 5 where the transmission power of a link is changed to the one used on the replacement path if the new transmission count is lower.

The correctness of the extended CTC can be shown as follows. Suppose $G_r = (V, \bigcup_{v \in V} T(v))$. Apparently, $G_r \subseteq G$. According to the definition of $T(v)$, each edge (u, v, i) in G_r has the lowest transmission count among all edges from u to v in G. Therefore, the shortest-path spanning tree of G, G_{opt}, is a subgraph of G_r because any edge (u, v, j) of G_{opt} must also be an edge of G_r. Otherwise, the edge from u to v in G_r represents a better edge, which contradicts the definition of shortest-path spanning tree. The rest of the proof requires to show that each edge in G_r has a replacement path with a dilation bound of at most t after the execution of CTC. This can be shown by the same argument in the proof of Theorem 2 in Section IV-E, because the dynamic programming procedure (steps 2-4) in CTC used to find replacement paths remain unchanged.

VI. Evaluation

We have evaluated CTC through two sets of simulations. We first study the network topology produced by CTC using a simple simulator, and then evaluate CTC through realistic packet-level simulations using an open-source WSN simulator called Prowler [44]. To create a realistic simulation environment, we implemented the probabilistic link model from USC [8] in both simulators. The USC model characterizes the transitional region in the reception performance of low-power radios on Mica2 motes based on a log-normal propagation model. Previous experiments showed that the USC model produces lossy and asymmetric links that approximate those in the networks of Mica2 motes [8].

A. Quality of Network Topology

In this section, we evaluate the topologies produced by CTC using a simple simulator. The transmission count of each link is computed according to the link model from USC [8].

In each simulation, nodes are uniformly deployed in a $150 \times 150\ m^2$ region. The number of nodes is 100 unless indicated otherwise. Each data point presented is the average of five different networks. Its 90% confidence interval is also shown. Each node can transmit at 11 different power levels from -20 dbm to 10 dbm, at an increment of 2 dbm[1].

We compare CTC against an existing topology control algorithm called LMST [9]. Each node running LMST builds a minimum spanning tree (in term of Euclidean distance) within its neighborhood and reduces its transmission power to reach only the neighbors on the tree. LMST is a representative localized topology control algorithm that is shown in [9] to outperform several earlier algorithms such as CBTC [15] and R&M [10].

The original design of LMST relies on a common maximum communication range of nodes and does not consider link quality. The notion of communication range is not applicable to lossy WSNs. We extend LMST to handle lossy networks as follows. A node includes another node in its one-hop neighborhood only when there exists a transmission power level at which the link yields a transmission count lower than the preset threshold. In other words, all links with a transmission count higher than the the threshold are blacklisted. Then each node builds a MST based on the transmission power of links within its one-hop neighborhood. Although a low threshold allows a node to find more neighbors and construct a MST with lower power, the resulting DTC can be very high due to

[1] The Chipcon CC1000 radio on Mica2 motes supports 32 power levels. While we only use 11 power levels in our simulations, using more power levels may further improve the performance and configurability of the network at the cost of higher overhead.

the low quality links on the MST. On the other hand, although a higher threshold achieves a lower dilation bound by only including good links on the MST, it can potentially eliminate too many links and cause network partitions. Our extensive simulations showed that a threshold of 1.67 in LMST yields the best communication performance without causing network partitions in our settings.

We first vary the search depth of CTC from 2 to 5 to evaluate its impact on the topology quality. For each combination of optimization metric and search depth, we measure the DTC of the network topology configured by each algorithm. Each setting is denoted as *CTC-control-metric-depth*. For example, *CTC-node-mm-3hop* represents the per-node control algorithm with the min_max metric with a search depth of 3 hops.

Fig. 4 shows the measured DTC under CTC-node with different search depths when the required dilation ranges from 1.5 to 5.5. CTC-node-ms yields the same DTC 1.5 irrespective the search depth. This is because the min_sum metric can lead to unbalanced node power on replacement paths. As a result, a node is often assigned high power, because it lies on many replacement paths. When the search depth increases, CTC-node-mm achieves a better configurability as it can find replacement paths with lower power. Fig. 4 shows that CTC-node can produce highly configurable network topologies with the min_max metric even when the search depth is as low as 3. Note that a small search depth is desirable as the time complexity of CTC increases with the search depth.

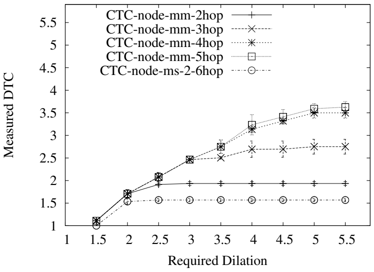

Fig. 4. Measured DTC of per-node CTC algorithms

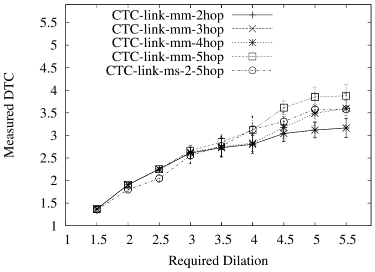

Fig. 5. Measured DTC of per-link CTC algorithms

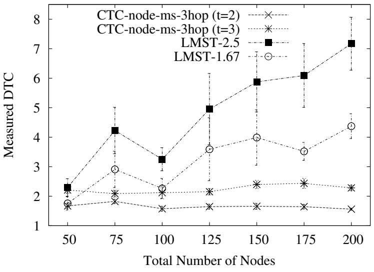

Fig. 6. Measured DTC of CTC and LMST

Fig. 5 shows the measured DTC under the CTC-link algorithms. Similar to CTC-node-ms, CTC-link-ms yields the same DTC irrespective of the search depth. We can see that CTC-link demonstrates a higher degree of configurability than CTC-node. This is because per-link control allows a node to use different transmission power when it lies on multiple replacement paths. Furthermore, a search depth of only 2 enables CTC-link to achieve a high degree of configurability at low computation cost. Overall our results show that the CTC-link algorithms can provide more efficient and flexible topology control than the CTC-node algorithms.

Fig. 6 compares the DTC of CTC and LMST algorithms under different node densities. LMST-2.5 and LMST-1.67 represent the LMST algorithm with a transmission count threshold of 2.5 and 1.67, respectively. Under all node densities, CTC consistently produces topologies that satisfy the required quality bounds. In contrast, the DTC of LMST has a high variation for different networks with the same density, and is heavily affected by node densities. This is because LMST tends to connect nodes with short and low-power links that are more likely to be lossy. This result shows that connectivity-based topology control algorithms cannot provide guaranteed path quality in lossy WSNs as they do not account for link quality. The DTC of LMST decreases with a lower transmission count threshold, because the links retained by each node become more reliable. However, a lower transmission count threshold may cause a node to blacklist too many links resulting network partition. It is therefore difficult to choose a transmission count threshold for LMST that achieves both low DTC and network connectivity under different network settings. We set the minimum transmission count threshold to 1.67 in the following simulations as it results in the lowest DTC without partitioning the network under our settings.

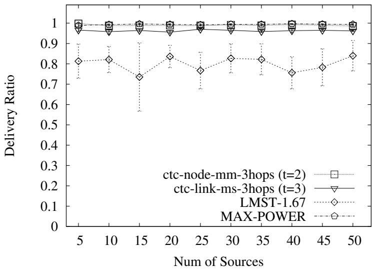

Fig. 7. Packet delivery ratio

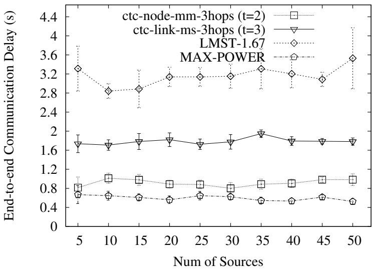

Fig. 8. Average delay of the received packets at the sink

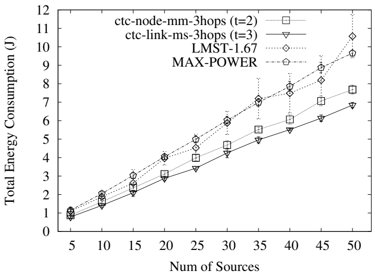

Fig. 9. Transmission energy consumption

B. Simulation Settings on Prowler

Prowler [44] is an open-source WSN simulator that has a layered event-driven structure similar to TinyOS. The MAC layer employs a CSMA/CA scheme similar to B-MAC [45]. The maximum number of retransmissions before dropping a packet is 3. DSDV [46] is used as the routing layer. We modified DSDV [46] to use transmission count as the routing metric, which is more suitable than hop count in lossy wireless networks [5], [39], [38].

The node distributions are the same as in the first set of simulations. The node bandwidth is 40 Kbps. The data packet size is 120 bytes. Each node runs an online link estimator similar to the one described in [5] to estimate the link quality in its two-hop neighborhood. The network follows a traffic pattern common in data collection applications [7]. Every source sends a packet to a base station every 5 minutes. The base station is located in the right border of the region. Sources are randomly chosen from the left 60% of the region to increase the distance to the base station. We vary the number of sources from 5 to 50. Each result is the average of 10 different network topologies with a 90% confidence interval. Each run lasts 80 minutes of simulated time.

C. Performance Results

We evaluate both communication performance and energy consumption of different algorithms. We evaluate two CTC algorithms: ctc-node-mm with a required DTC bound of 2, and ctc-link-ms with a required DTC bound of 3. The search depth is set to 3. Besides LMST, we also use the network topology where each node transmits at the maximum power as a baseline, which is denoted *MAX-POWER*. As light load is used in our simulations, MAX-POWER yields the best performance in terms of delay and delivery ratio.

Fig. 7 shows the data delivery ratio under each algorithm. Similar to MAX-POWER, all CTC algorithms delivered over 95% of the total packets to the base station. LMST yields the lowest delivery ratio due to the lossy links on its topology.

Fig. 8 shows the average delay of the packets received by the base station. LMST yields the highest delay because a packet often experiences retransmissions over lossy links. Both CTC algorithms achieve lower delay than LMST. Furthermore, the delay under CTC increases with a higher DTC bound. This result shows that CTC enables applications to effectively control the network performance by adjusting the DTC bound.

Fig. 9 shows the transmission energy consumed by different algorithms. CTC-link performs slightly better than CTC-node. Interestingly, although LMST assigns *lower* power than the other algorithms, the network consumes almost the same amount of *energy* under LMST as under MAX-POWER. This is because, the links on LMST's topology are less reliable resulting in more energy wasted for packet retransmissions. Therefore, the benefit of lower power is offset by the increase in the number of transmissions in lossy networks. In contrast, CTC-link-ms reduces the energy consumption by $27\% \sim 36\%$ compared with MAX-POWER. This result demonstrates the importance of considering lossy link models in both the design and evaluation of topology control algorithms.

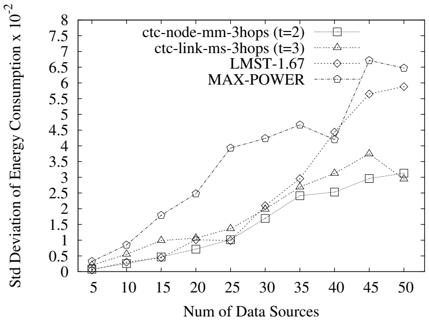

Fig. 10. The standard deviation of transmission energy of all nodes.

Fig. 10 shows the standard deviation of nodes' transmission energy consumption in a typical run. The variation of the energy consumption affects the lifetime of the network before partition. Both CTC-node and CTC-link achieve significantly lower variation in nodes' energy consumption than LMST when source density is high. They also achieve much more balanced energy consumption in the network than MAX-POWER under all source densities. This result indicates that CTC can effectively prolong the lifetime of the network.

VII. Conclusion

In this paper, we first provide a new formulation of the topology control problem that captures the stochastic nature of WSNs. We then propose the Configurable Topology Control (CTC) approach for lossy WSNs. The key novelty of CTC lies in its capability of configuring a network topology to achieve desired path quality bounds in a lossy network through localized algorithms. We present four CTC algorithms that combine per-node/per-link power control with two metrics for power assignment. Realistic simulations based on the characteristics of Mica2 motes show that CTC can provide desired tradeoff between power consumption and network performance according to application requirements. Furthermore, CTC outperforms LMST in terms of both communication performance and energy consumption. Our results demonstrate the importance of incorporating lossy link models in the design of topology control algorithms for WSNs.

REFERENCES

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

[2] Alberto Cerpa, Jennifer L. Wong, Louane Kuang, Miodrag Potkonjak, and Deborah Estrin, "Statistical model of lossy links in wireless sensor networks," in *IPSN*, 2005.

[3] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *MobiSys*, 2004.

[4] Paolo Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Comput. Surv.*, vol. 37, no. 2, 2005.

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

[6] Philip Levis, Neil Patel, David Culler, and Scott Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks.," in *NSDI*, 2004.

[7] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler, "An analysis of a large scale habitat monitoring application," in *SenSys*, 2004.

[8] 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.

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

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

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

[12] 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.

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

[14] G. Calinescu, S. Kapoor, and M. Sarwat, "Bounded-hops power assignment in ad-hoc wireless networks," in *WCNC*, 2004.

[15] 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.

[16] Douglas M. Blough, Mauro Leoncini, Giovanni Resta, and Paolo Santi, "The lit k-neigh protocol for symmetric topology control in ad hoc networks," in *MobiHoc*, 2003.

[17] Xiang-Yang Li, Wen-Zhan Song, and Yu Wang, "Localized topology control for heterogeneous wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, 2006.

[18] Ning Li and Jennifer C. Hou, "Localized topology control algorithms for heterogeneous wireless networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 6, 2005.

[19] Errol L. Lloyd, Rui Liu, Madhav V. Marathe, Ram Ramanathan, and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *Mob. Netw. Appl.*, vol. 10, no. 1-2, 2005.

[20] Ning Li and Jennifer C. Hou, "Flss: a fault-tolerant topology control algorithm for wireless networks," in *MobiCom*, 2004.

[21] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Chih-Wei Yi, "Fault tolerant deployment and topology control in wireless networks," in *MobiHoc*, 2003.

[22] MohammadTaghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni, "Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks," in *MobiCom*, 2003.

[23] Roger Wattenhofer and Aaron Zollinger, “Xtc: A practical topology control algorithm for ad-hoc networks,” in *4th International IEEE Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2004.

[24] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A. Stankovic, and Tian He, “Atpc: adaptive transmission power control for wireless sensor networks,” in *SenSys*, 2006.

[25] Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger, “Topology control meets sinr:: the scheduling complexity of arbitrary topologies,” in *MobiHoc*, 2006.

[26] David Eppstein, “Spanning trees and spanners,” Tech. Rep. ICS-TR-96-16, 1996.

[27] 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.

[28] Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan, “Distributed construction of a planar spanner and routing for ad hoc wireless networks,” in *INFOCOM*, 2002.

[29] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu, “Geometric spanner for routing in mobile networks,” in *Proc. 2nd ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Oct. 2001, pp. 45–55.

[30] Crossbow, “Mica and mica2 wireless measurement system datasheets,” 2003.

[31] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J.A. Stankovic, and T.F. Abdelzaher, “Real-time power-aware routing in sensor networks,” in *IEEE International Workshop on Quality of Service (IWQoS’06)*, 2006.

[32] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann, “Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks,” in *SECON*, 2004.

[33] Lichun Bao and J. J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *MobiCom*, 2001.

[34] Injong Rhee, Ajit Warrier, Jeongki Min, and Lisong Xu, “Drand:: distributed randomized tdma scheduling for wireless ad-hoc networks,” in *MobiHoc*, 2006.

[35] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” *Wirel. Netw.*, vol. 12, no. 1, 2006.

[36] Weizhao Wang, Xiang-Yang Li, Ophir Frieder, Yu Wang, and Wen-Zhan Song, “Efficient interference-aware tdma link scheduling for static wireless networks,” in *MobiCom*, 2006.

[37] Octav Chipara, Chenyang Lu, and John Stankovic, “Dynamic conflict-free query scheduling for wireless sensor networks,” in *ICNP*, 2006.

[38] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom*, 2003.

[39] Richard Draves, Jitendra Padhye, and Brian Zill, “Comparison of routing metrics for static multi-hop wireless networks.,” in *SIGCOMM*, 2004, pp. 133–144.

[40] Robert E. Tarjan, *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, 1996.

[41] W.-T. Chen and N.-F. Huang, “The strongly connecting problem on multihop packet radio networks,” *IEEE Transactions on Communications*, vol. 37, no. 3, 1989.

[42] Michael Elkin and David Peleg, “The hardness of approximating spanner problems,” in *17th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, 2000.

[43] M Desrochers and F. Soumis, “A generalized permanent labeling algorithm for the shortest path problem with time windows.,” *INFOR.*, vol. 26:191-212, 1988.

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

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

[46] Charles E. Perkins and Pravin Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” in *SIGCOMM: Proceedings of the conference on Communications architectures, protocols and applications*, 1994.