Localized and Configurable Topology Control in Lossy Wireless Sensor Networks

Guoliang Xing; Chenyang Lu; Robert Pless
Washington University in St. Louis

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 existing topology control algorithm called LMST. Our results demonstrate the importance of incorporating lossy links of WSNs in the design of topology control algorithms.

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 [35], [4], [36] revealed that the lossy nature of wireless links in WSNs has introduced a major challenge to achieving reliable and power-efficient multi-hop communication. Lossy links can result in excessive energy wastage and severe degradation in communication performance. For instance, a recent study on Mica motes [7] showed that up to 80% of the total energy consumption of the radio was attributed to packet loss [35].

Topology control is a key technique to reducing network transmission power while maintaining desired network properties. A multitude of topology control algorithms [27] 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 [35], [33] 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 [14], while sporadic data loss is tolerable for data collection in dense WSNs since sensor data usually has high redundancy [30]. Therefore, topology control must minimize the power consumption of the network while achieving the desired path quality required by the application.

Finally, unlike traditional ad hoc networks, many WSNs only impose light load. For instance, in the WSN deployed at Great Duck Island for habitat monitoring [30], 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 rate. Topology control algorithms may be designed to exploit the light load of this common class of applications.

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 [37] that captures lossy link characteristics of Mica2 motes. Our results show that CTC significantly outperforms a traditional topology control scheme called LMST [17] 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 presents the simulation results. Section VI 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 [27] for a comprehensive survey on the existing topology control algorithms. They fall into two basic classes: per-link control [26], [25], [22], [13], [3] and per-node control [15], [17], [2], [34], [20]. 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 [16], [19], [12] 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 [31] 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. More recently, Mocibroda et al. [21] 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 with light workload, 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 [10] for a review of the existing centralized algorithms for constructing graph spanners. Recently, localized algorithms have also been proposed [1], [18], [11]. 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 \( S = \{ P_i | 1 \leq i \leq n \} \), \( P_i > P_j \iff i > j \). For example, the CC1000 radio on Mica2 motes [7] can transmit at 32 different power levels. 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 [37], [36], [4], or using a link estimator [33], [5] 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. Acknowledgements are sent at the maximum power level by receivers. We note that such an acknowledgement scheme does not incur high overhead as the length of acknowledgements is often much shorter than that of data packets in WSNs. For instance, in the MAC protocol of TinyOS, B-MAC [24], the length of an acknowledgement is only five bytes while the length of a data packet is 46 bytes by default.

The network is denoted by a directed graph \( G(V,E) \). \( V \) includes all nodes in the network. \( E = \{(u,v,i) | R_{u,v,i} \leq T; u,v \in V; P_i \in S \} \). We ignore the links with a transmission count greater than \( T \). \( G(V,E) \) is a multi-graph where there may exist multiple links between two nodes at different power.

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 focus on the WSNs that impose light workload and hence 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 justified by the fact that higher transmission power will result in higher signal strength, which alleviates the impact of path fading and noise. This assumption has been confirmed by several recent empirical studies on WSNs [29], [5]. Finally, we assume nodes are stationary. Note that most existing WSNs are composed of stationary nodes.

B. Topology Control Problem

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. When per-node control is used, a power assignment assigns a transmission power to every node in the network. $G_Ω(V, E_Ω)$ represents the network graph under 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.

Given a DTC bound $t \geq 1$ specified by the application, our objective is to choose a power assignment $Ω$ that minimizes the total power of all nodes under the constraint that the DTC is no greater than $t$:

$$Ω = \arg\min_{P_1 \in Ω} \sum_{(u,v) \in Ω} P_1, \text{ subject to } \max_{u,v \in V} \frac{Γ_{G_u}(u,v)}{Γ_{G_M}(u,v)} \leq t$$

where $Γ_{G_u}(u,v)$ denotes the minimum transmission count from $u$ to $v$ in the network under power assignment $X$.

The other variations of the problem can be formulated similarly. When the metric is $\min_{\text{max}}$, the minimization objective needs to be replaced by $\max_{\text{max}} P_1 \in Ω P_1$. When the per-link control is used, power assignment $Ω$ assigns a transmission power to a node for each of its outgoing links.

IV. THE CTC ALGORITHMS

In this section, we present a set of Configurable Topology Control (CTC) algorithms. CTC enforces 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 [33], [5] and is not the focus of this paper.

![Diagram](image)

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 $\min_{\text{sum}}$ 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 $\min_{\text{sum}}$ 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 composed of $(b, e, max)$ 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 will 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) [8] for the shortest path problem with time window (SPPTW). A special case of SPPTW, the weight-constrained shortest path (WSCP) problem, resembles our problem. Each link in a WSCP 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 WSCP 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.

\begin{align}
\text{Input:} & \quad t, d, N_1(u), N_2(u) \\
\text{Output:} & \quad \text{power}(u) \\
\text{power}(u) = \min; \\
\text{for} \quad v \in V_1(u) & \quad \text{call} \; LabelSet(v); \quad \text{end}
\end{align}

\text{function} \; LabelSet(v) 
\begin{align}
1) & \quad W = t \cdot \max\{R_{w,w,\max}\} \cap (v, w, \max) \in E_1(v). & \text{Set} \; L_v = \{0, 0\} \text{ and } L_i = \emptyset \text{ for all } i \in V_1(v) \setminus \{v\}. \\
2) & \quad \text{If all labels have been marked, go to 5); else choose } i \in V_1(v) \text{ that has an unmarked label } (R_i^q, P_i^q) \text{ with minimal } R_i^q. \\
3) & \quad \text{For each link } (i, j, k) \in E_2(u) \text{ do:} \\
4) & \quad \text{Mark label } (R_i^q, P_i^q). \text{ Go to step 2.} \\
5) & \quad \text{For each link } (v, w, \max) \in E_1(v), \text{ do:} \\
& \quad a) \text{ Find the label } (R_{w,w,\max}) \in E_{w} \text{ such that } R_{w}^q \leq t \cdot R_{w,w,\max} \text{ and has the minimal } P_{w}^q. \\
& \quad b) \text{ If there exists a } u \text{’s link } (u, z, k) \in q \text{ and } \text{power}(u) < P_{k} \text{, then:} \\
& \quad \text{power}(u) = P_{k}.
\end{align}

Fig. 2. The per-node CTC executed at $u$ with the min_sum metric.
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 (2)-(5) are met.

Constraint (2) 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 restricts the search space by eliminating the paths that would have a dilation higher than \( t \). Constraint (3) limits the maximum hop count of a path to \( d \). Constraint (4) 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 (5) 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 (5) 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 (2) and evolve into valid replacement paths in following iterations. It can be seen that this property allows \( \text{LabelSet} \) to find the optimal replacement path (e.g., with the minimum total power) under constraints (2)-(4).

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 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 (5) needs to be changed to \( \exists(R^i_q, P^i_q) \in L_j \). \( R^i_j \leq R^i_q + R_{i,j,k} \) and \( P^i_q \leq \max(P^i_q, P_k) \).

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 \( \text{power}(u,v) \) 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 \( \text{power}(u,z) < P_k \), \( \text{power}(u,z) = P_k \). Notice that both per-node and per-link control share the same procedure for searching replacement paths (step 1 to 4 of function \text{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 1: Suppose \( M \) is the power assignment where each link is assigned the maximum power. \( \Omega \) is the power assignment produced by the CTC algorithm with a DTC bound \( t \geq 1 \). Then the network \( G_{\Omega} \) satisfies the DTC bound \( t: \max_{u,v \in V} \frac{\text{DTC}(u,v)}{\text{DTC}(\Omega(u,v))} \leq t \).

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 \( G_{\Omega} \) with a dilation no greater than \( t \). We prove this holds after the execution of CTC at each node.

Suppose \( u \) finds a replacement path \( F^u_{v,w} \) for \((v,w,\max)\), and \( u \) lies on the path. Note that \( F^u_{v,w} \) corresponds to the label \((R^u_q, P^u_q)\) found by \( u \) at step 5.a. According to step 5.a, \( F^u_{v,w} \) 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 \( F^u_{v,w} \). That is, \( u \) assigns power \( P_i \) to \( x \). We now show that the power choice of node \( x \) itself, \( \text{power}(x) \), is no lower than its power, \( P_i \), assigned by \( u \) on path \( F^u_{v,w} \). It can be seen that \( x \in V_1(v) \) since a node in \( V_2(v) - V_1(v) \) does not have any outgoing links, and hence \((x,y,i)\) would not exist.

First, \( \text{power}(x) \geq P_i \) holds if \( x = v \), according to step 5.b (in \( x \)'s execution of CTC) in Fig. 2. We now show \( \text{power}(x) \geq P_i \) also holds for any \( x \in V_1(v) - \{v\} \). We define graph \( G^* = (V^*, E^*) \) as follows.

\[
V^* = \bigcap_{k \in V_1(v)} V_1(k)
\]

\[
E^* = \bigcup_{(a,b) \in V^* \cap \{E_1(u)\}} (a,b,i)
\]

\( V^* \) is the intersection of the one-hop neighbor set of \( v \) and the one-hop neighbor sets of all \( v \)'s one-hop neighbors. \( E^* \) comprises the links between these nodes.

In the following, we will show that \( G^* \) is the search space for the replacement path of \((v,w,\max)\) in both \( x \) and \( u \)'s local executions of CTC. First, according to constraint (4), all the nodes on any replacement path of \((v,w,\max)\) must
be one hop from each other. Hence, all of them lie on \( G^* \). Second, \( E^* \subseteq E_2(u) \cap E_2(x) \), because \( E^* \) contains only the links between the nodes in \( V^* \) that are included in both \( V_1(u) \) and \( V_1(x) \). Hence \( G^* \subseteq N_2(u) \cap N_2(x) \). Therefore, both \( u \) and \( x \) search for the replacement path of \( (v, w, \text{max}) \) on \( G^* \) in their local executions of CTC. This property, combined with the fact that the replacement path is optimal (in terms of total power) among all paths on \( G^* \) that satisfy the dilation bound and constraints (2) and (3), leads to the conclusion that \( x \) and \( u \) find the same replacement path \( F_{u,v}^* \). Hence, according to step 5.b, the power of \( x \) after its execution of CTC is no lower than its power on the replacement path found by \( u \).

We have shown that each replacement path 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 \( \min_{\text{sum}} \) or \( \min_{\text{max}} \) 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 \( \min_{\text{sum}} \) or \( \min_{\text{max}} \) metric) on graph \( G^* \) that is shared by the two-hop neighborhoods of all the nodes on the path.

**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 \( \text{LabelSet}(v) \) without constraints (3) and (4) 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 (5), 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 \( \text{LabelSet}(v) \) is lower due to the constraints (3) and (4) in Fig. 2. Specifically, (3) 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 \( \text{LabelSet} \) is bounded by \( O(|V_1|^{d-1}) \). Hence the time complexity of \( \text{LabelSet} \) is bounded by \( O(\min(|V_1|^{d-1}, |E_2|W)) \). Since \( \text{LabelSet} \) 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 (4). 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 V. This is, in part, because it is unlikely to find long replacement paths within a two-hop neighborhood.

**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). When a node failure is detected, rerunning CTC for all nodes in the one-hop neighborhood of the failed node suffices to preserve the DTC bound for the network. 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 re-running 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.

2) **Handling contention and interference**: CTC is designed for a common class of WSNs that only impose light load. However, such networks may still experience transient interference and contention due to concurrent transmissions that occur occasionally. Consequently, the transmission counts of the affected links may increase temporarily. In such a case, CTC should avoid increasing the transmission power because the increase in transmission counts is not caused by link quality degradation. This problem may be mitigated by measuring transmission counts at a lower frequency. A more effective solution is to employ MAC-layer mechanisms [32] that can distinguish collisions from pack loss. If the transmission failure is caused by collisions of concurrent transmissions, CTC avoids increasing transmission power until the transient interference or contention disappears.

3) **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.

V. 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 [28]. To create a realistic simulation environment, we implemented the probabilistic link model from USC [37] in both simulators. Previous experiments showed that the USC model produces lossy and asymmetric links that approximate those in the networks of Mica2 motes [37].

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 [37].

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.

We compare CTC against an existing topology control algorithm called LMST [17]. 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 designed based on traditional deterministic link models, and earlier results [17] showed that it yields a better power efficiency than several earlier algorithms such as CBTC [15] and R&M [26].

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 therefore extend LMST to handle lossy networks by blacklisting all links with a transmission count higher than a threshold. A node includes another node in its neighborhood only when the link to the node has a transmission count lower than the preset threshold.

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. 3 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. 3 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 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. 5 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.

B. Simulation Settings on Prowler

Prowler [28] 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 [24]. The maximum number of retransmissions before dropping a packet is 3. DSDV [23] is used as the routing layer. We modified DSDV [23] to use transmission count as the routing metric, which is more suitable than hop count in lossy wireless networks [33], [9], [6].

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 [33] to estimate the link quality in its two-hop neighborhood. The network follows a traffic pattern common in data collection applications [30]. 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. 6 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. 7 shows the average delay of the packets received at 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. 8 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% ~ 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. 9 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.

VI. CONCLUSION

In this paper, we first provide a new formulation of the topology control problem that captures the stochastic nature of WSNs. We then proposes 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


