Localized and Configurable Topology Control in Lossy Wireless Sensor Networks

Guoliang Xing
Department of Computer Science
City University of Hong Kong, Hong Kong
Email: glxing@cs.cityu.edu.hk
Tel: (852) 2788–7525 Fax: (852) 2788–8614

Chenyang Lu and Robert Pless
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO USA
Email: {lu,pless}@cse.wustl.edu.

***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 that 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; Localized 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 [1] 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 testbed 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].

Topology control is a key technique to reducing network transmission power while maintaining desired network properties. A multitude of topology control algorithms [2] 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], [3] 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, while sporadic data loss is tolerable for data collection in dense WSNs since sensor data usually has high redundancy [4]. 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 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. (3) We conducted extensive simulations based on a realistic link model [5] that captures lossy link characteristics of Mica2 motes. Our results show that CTC significantly outperforms a representative topology control scheme called LMST [6] in terms of delivery rate, data latency and energy consumption.

II. RELATED WORK

Topology control aims at maintaining desirable properties of wireless networks (e.g., connectivity and power efficiency).

This work is funded in part by City University of Hong Kong under a grant #9610062, and the NSF under a NeTS-NOSS grant # CNS-0627126.

We refer to [2] for a comprehensive survey on the existing works. They fall into two basic classes: per-link control and per-node control. 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 [7], [8] 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 [9] 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 [10] 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.

The metric of dilation of transmission count in this paper is related to the *stretch factor* in graph spanner problems. We refer to [11] for a review of the existing centralized algorithms for constructing graph spanners. Recently, localized algorithms have also been proposed [12], [13]. 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 [14] 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 [5], or using a link estimator [3], [15] 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.

In this paper, we mainly focus on the WSNs that experience little interference or contention caused by concurrent transmissions. Accordingly, we assume $P_i > P_j \Rightarrow R_{u,v,i} < R_{u,v,j}$. 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 [4], each of the 98 motes wakes up every 20 minutes to send its data to the base station. Furthermore, interference can be eliminated or significantly reduced by scheduling interfering nodes to communicate at different times in TDMA MAC protocols [16].

B. Topology Control Problems

In this paper, we consider both *per-node* and *per-link* power control strategies with two optimization metrics: min_sum and min_max that minimizes the total and the maximum power of all nodes or links, respectively. We now formulate the problem with per-node control and the min_sum metric, and the formulations of other cases can be found in [17].

G_M denotes the topology where each node is assigned the maximum power. G_M achieves the best path quality under any possible power assignment when the network workload is light. G_Ω represents the topology 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. This metric closely relates to network performance like reliability, throughput, and delay.

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, choose a power assignment Ω with the minimum sum while the DTC bound of the induced topology under Ω is no greater than t:

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

$\Gamma_{G_X}(u, v)$ 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 $\max_{P_i \in \Omega} P_i$.

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. 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.

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.*, $(u, v, i) \in E_1(u) \Leftrightarrow (v, u, j) \in E_1(v)$.

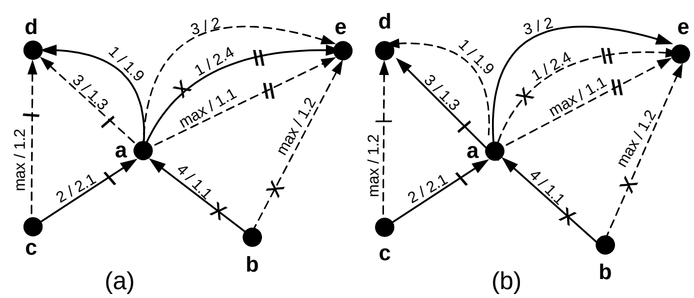

Fig. 1. The execution of two algorithms with a required DTC bound of 3. (a) illustrates a naive algorithm in which each node only replaces its own max-power links. (b) illustrates the CTC algorithm with the min_sum metric. Each link is labeled by *power / transmission count*. max represents the maximum 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 an example in Fig. 1. The DTC bound required is 3. We first describe a naive 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. In Fig. 1(a) 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 power 3 and 1 on the three paths. If each node sets its power according to the paths it finds, a will choose a power of 1 as it is not aware of the existence of the other replacement paths. 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. 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 all the nodes on a replacement path find the same path when they replace the same max-power link. In addition to replacing its own max-power links, each node running CTC 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.

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. u invokes the function *LabelSet(v)* for each node $v \in V_1(u)$ including itself. In doing so, u "simulates" the execution of CTC at all one-hop neighbors, which enforces consistent replacement paths found by different nodes. *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) [18] 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:
$$R_i^q + R_{i,j,k} \leq W \quad (1)$$
$$|q| < d \quad (2)$$
$$j \in \bigcap_{k \in V_1(v)} V_1(k) \quad (3)$$
$$\nexists (R_j^q, P_j^q) \in L_j, \; (R_j^q \leq R_i^q + R_{i,j,k}) \wedge (P_j^q \leq P_i^q + P_k) \quad (4)$$
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 (1)-(4) are met.

Constraint (1) 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 (2) limits the maximum hop count of a path to d. Constraint (3) 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 (4) 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 (4) 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 (1) 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 (1)-(3).

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 (4) 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. Analysis of CTC

The replacement path $F_{u,v}$ of link (u, v, max) found by u in CTC is also found by all the nodes on the path. Therefore, after the execution of CTC, each node on $F_{u,v}$ chooses a power no lower than the value assigned on $F_{u,v}$, which ensures the dilation bound of the topology is no higher than requested. The detailed proof is omitted due to space limitation and can be found in [17].

The time complexity of $LabelSet(v)$ without constraints (2) and (3) is similar to the original GPLA algorithm that has a complexity of $O(|E_2|W)$. The overall time complexity can be shown to be $O(|V_1| \cdot min(|V_1|^{d-1}, |E_2|W))$ [17] after accounting for the constraints (2) and (3). Although this bound is exponential in search depth d, we show experimentally that small search depth, (*e.g.*, choosing $d = 2$ or 3) gives a very good performance in Section V.

V. Evaluation

We have evaluated CTC by two sets of simulations. The DTC of network topologies produced by CTC are compared against that of LMST using a simple simulator. The results can be found in [17]. We now present the packet-level simulation results on a network simulator called Prowler [19].

A. Simulation Settings

Prowler [19] is an open-source WSN simulator that has a layered event-driven structure similar to TinyOS. To create a realistic simulation environment, we implemented the probabilistic link model from USC [5] in Prowler. Previous experiments showed that the USC model produces lossy and asymmetric links that approximate those in the networks of Mica2 motes [5]. The MAC layer employs a CSMA/CA scheme similar to B-MAC [20]. The maximum number of retransmissions before dropping a packet is 3. DSDV [21] is used as the routing layer. We modified DSDV to use transmission count as the routing metric, which is more suitable than hop count in lossy wireless networks [3].

In each simulation, 100 nodes are uniformly deployed in a 150×150 m^2 region. The base station is located in the right border of the region and sources are randomly chosen from the left 60% of the region to increase the distance to the base station. Every source sends a packet to the base station every 5 minutes, which is similar to the traffic pattern in a habitat monitoring application [4]. We vary the number of sources from 5 to 50. Nodes can transmit at 11 different power levels from -20 dbm to 10 dbm, at an increment of 2 dbm. Node bandwidth is 40 Kbps and data packet size is 120 bytes. Each node runs an online link estimator similar to the one described in [3] to estimate the link quality in its two-hop neighborhood. Each simulation lasts 80 minutes. Results presented are the average of five different topologies with 90% confidence interval.

B. Performance Results

We choose an existing topology control algorithm called LMST [6] as the baseline for performance comparison. 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 [6] to outperform several earlier algorithms.

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. This threshold is set to 1.67 in the simulations as it yields the best communication performance without causing network partitions in our settings. 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.

Two configurations of CTC are evaluated: CTC with node control and min_max metric (ctc-node-mm), and CTC with link control with min_sum metric (ctc-link-ms). The required DTC bounds are 2 and 3, respectively. The search depth is set to 3. Fig. 3 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. 4 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. 5 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 $27\% \sim 36\%$

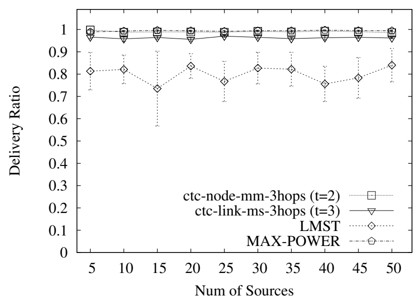

Fig. 3. Packet delivery ratio

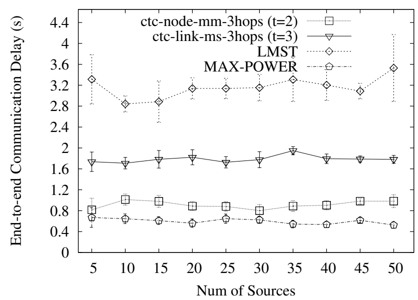

Fig. 4. Average delay of the received packets at the sink

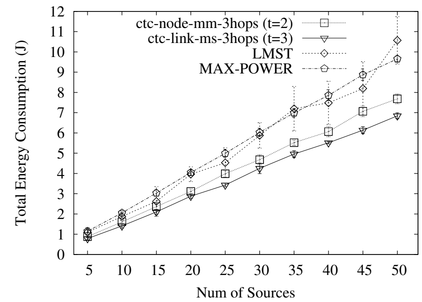

Fig. 5. Transmission energy consumption

compared with MAX-POWER. This result demonstrates the importance of considering lossy link models in both the design and evaluation of topology control algorithms.

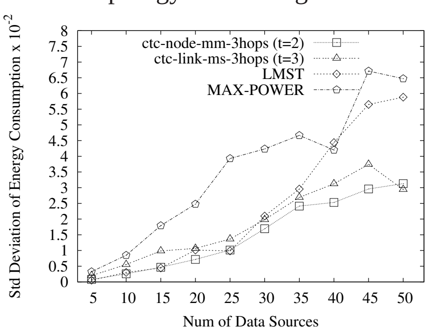

Fig. 6. The standard deviation of transmission energy of all nodes.

Fig. 6 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 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] Paolo Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Comput. Surv.*, vol. 37, no. 2, 2005.

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

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

[5] Marco Zuniga and Bhaskar Krishnamachari, "Analyzing the transitional region in low power wireless links," in *SECON*, October 2004.

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

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

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

[9] Roger Wattenhofer and Aaron Zollinger, "Xtc: A practical topology control algorithm for ad-hoc networks," in *WMAN*, 2004.

[10] 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.

[11] David Eppstein, "Spanning trees and spanners," Tech. Rep. ICS-TR-96-16, 1996.

[12] 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.

[13] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu, "Geometric spanner for routing in mobile networks," in *MobiHoc*, Oct. 2001, pp. 45–55.

[14] Crossbow, "Mica and mica2 wireless measurement system datasheets," 2003.

[15] 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 *IWQoS*, 2006.

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

[17] Guoliang Xing, Chenyang Lu, and Robert Pless, "Configurable topology control algorithms for lossy wireless sensor networks," Tech. Rep. WUCSE-05-34, Washington University, Department of Computer Science and Engineering, 2005.

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

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

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

[21] Charles E. Perkins and Pravin Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *SIGCOMM*, 1994.