2016 IEEE 41st Conference on Local Computer Networks

# Topology Control in Wireless Sensor Networks: What Blocks the Breakthrough?

Michael Stein*, Tobias Petry*, Immanuel Schweizer*, Martina Brachmann†, Max Mühlhäuser*
*Telecooperation Group, Technische Universität Darmstadt
{michael.stein, schweizer, max}@tk.informatik.tu-darmstadt.de
tobias.petry@stud.tu-darmstadt.de
†Embedded Systems Lab, Technische Universität Dresden
martina.brachmann@tu-dresden.de

***Abstract*—Graph-based topology control adapts wireless topologies to achieve certain target graph structures. Wireless sensor networks seem well-suited for the expectations (in particular those on provided energy savings) raised by topology control. Nevertheless, topology control has never made the breakthrough in real-world deployments. This work explores the reasons for this, identifying five practical obstacles of today's topology control: (i) unrealistic assumptions, (ii) unsuitable graph structures, (iii) application agnosticism, (iv) unclear role in the stack, and (v) insufficient framework support.**

**To address the latter obstacle, we provide a re-usable framework for the implementation and evaluation of topology control.**

**Based on this framework, we conduct a testbed-based evaluation for two application scenarios and three topology control algorithms including a novel application-specific algorithm. Indeed, the identified obstacles hinder topology control from boosting the application. However, the achieved graph structures show the practical feasibility of topology control in principle.**

***Index Terms*—WSN, topology adaptation, graph, testbed**

## I. INTRODUCTION

Graph-based topology control has been an active research field in wireless networks, especially wireless sensor networks (WSNs), for a long time [31]. The core idea of topology control is to remove communication links from the topology to achieve specific graph properties like planarity, bounded node degree, spanner or low-weight [20], [21], [24]. These graph properties are typically assumed to serve general optimization goals, e.g., reducing energy consumption [25]. In other cases, topology control serves for providing a topological structure required by specific higher-layer protocols [15].

The expectations raised by topology control, in particular those on provided energy savings, are a perfect fit for the core requirements of WSNs. Nevertheless, although new topology control algorithms are presented on a regular basis [2], [4], [14], [25], topology control has never made the breakthrough in real-world deployments. This paper explores the reasons for this fact because understanding the key weaknesses of today's topology control algorithms is a first important step for the development of more practical algorithms in the future.

Based on our analysis, we identify five practical obstacles that hinder topology control from making its way into practical deployments. In particular, we observe that existing algorithms typically have (i) unrealistic assumptions and aim for (ii) unsuitable or irrelevant graph structures. Moreover, most topology control algorithms are (iii) agnostic of the application, and (iv) the role of topology control in the communication stack is still unclear. Lastly, (v) the lack of appropriate frameworks and tools prevents researchers from evaluating their topology control algorithms under realistic conditions.

To support a simple implementation and practical evaluation of topology control algorithms, we provide a publicly available, modular framework for the Contiki operating system [8]. Our open source framework allows for adding new topology control algorithms in just a few lines of code and can be executed in combination with any of Contiki's existing communication stacks.

Using the FlockLab WSN testbed [22], we conduct an evaluation of the identified obstacles for three topology control algorithms: LMST [19], a-kTC [25], and l*-kTC. The algorithm l*-kTC builds upon l-kTC [27], an application-specific topology control algorithm. l-kTC operates in a centralized way based on global knowledge, which is practically infeasible. Proposing l*-kTC, this paper presents the first distributed algorithm for application-specific topology control. None of the topology control algorithms achieves improvements in terms of energy consumption or packet delivery for the two considered applications. However, although the proven theoretical graph properties provided by the algorithms build upon unrealistic assumptions, these properties can be partially achieved, even in our harsh testbed environment. This shows the practical feasibility of topology control in principle.

In summary, this paper provides the following contributions:

1) We provide a general analysis of existing topology control algorithms with respect to their practicality.
2) We provide a modular framework for the implementation and evaluation of topology control algorithms.
3) We present l*-kTC, the first distributed algorithm for application-specific topology control.
4) We conduct an evaluation of topology control, demonstrating the identified obstacles in a practical setting.

The rest of this paper is structured as follows. Section II outlines the core idea and expectations of topology control. Our analysis in Section III identifies the practical obstacles. Section IV introduces the provided framework, followed by our testbed-based evaluation in Section V. Section VI discusses related work, and Section VII concludes the paper.

DOI 10.1109/LCN.2016.67

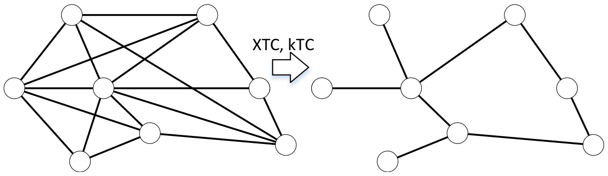


Fig. 1. Example topology and its modification by topology control

## II. The Vision of Topology Control

WSN devices are inherently resource-restricted [35], making efficient communication a core requirement. Based on this observation, many algorithms for graph-based topology control have been proposed [31]. The wireless topology is modeled as a graph $G = (V, E)$, where nodes $V$ and edges $E$ denote WSN devices and communication links, respectively. Topology control adapts $G$ to provide a target graph structure $G_{TC} = (V, E_{TC})$ with $E_{TC} \subseteq E$. That is, each node selects a subset of logical neighbors and limits communication to these nodes. This is the authors' definition of topology control. Alternative definitions can be found in Section VI.

Figure 1 shows an exemplary adaptation as conducted by the topology control algorithms XTC [32] and kTC [24]. The output topology $G_{TC}$, which results from breaking triangles in $G$, has interesting properties often promised by topology control and thus serves for the demonstration of expected benefits of topology control. First of all, most long-distance links (measured, e.g., geographically or based on signal strength) have vanished from the topology. This has two main implications. First, routing mechanisms use only short links. Provided a power control scheme is used [10], sending a packet over short links may save energy compared to long links because the transmission power is lower and less unintended sensors receive the packet. Second, using shorter links may result in higher throughput because transmissions in multiple network regions may be conducted in parallel [11]. Moreover, the node degree (i.e., the density) of the output topology is bounded. Sparse topologies may lead to smaller (and thus less complex) routing tables. Flooding a sparse network is often expected to be less costly, e.g., due to the broadcast storm problem [29]. Finally, the shown output topology is planar (i.e., it contains no crossing edges), which is for example useful for geographic routing [15].

On the other hand, removing edges from $G$ may also have undesired implications. Most importantly, the length of the shortest path between nodes increases (e.g., in terms of the number of hops), with a possibly negative impact on delay and packet delivery. Considering this, numerous topology control algorithms aim at providing so-called spanner properties [31], which bound the above distance increase by a constant factor.

In general, various papers report impressive results in their evaluations, raising high expectations on the practical benefit of graph-based topology control. Still, we do not observe widespread usage of this concept in real-world WSNs.

## III. Practical Obstacles of Topology Control

Even though the expectations raised by topology control seem well-suited for WSNs, topology control has never made the breakthrough in real-world deployments. In this section, we identify the main practical obstacles of topology control.

### A. Obstacle 1: Unrealistic Assumptions

The early research on topology control is deeply rooted in graph theory, aiming at achieving certain geometrical structures [31]. The theoretical roots of topology control are probably the main reason that most topology control algorithms make simplified assumptions not holding true in real-world deployments. These assumptions can be classified according to the following two classes: (i) unrealistic model of connectivity or (ii) negligence of WSN-specific characteristics.

(i) Employing a realistic connectivity model is essential for topology control because this model serves as input for deciding which edges should be removed from the topology. The most widespread connectivity model assumed in the topology control community is a unit disk graph (UDG). In a UDG, all nodes have a circular transmission range of equal size, and two nodes are connected if they are within the transmission range of each other. Unfortunately, the UDG model considers the complexity and dynamics of wireless links insufficiently [17]. Nevertheless, even some of the most recent topology control algorithms still build upon the UDG, e.g., [2]. Relying on an unsuitable connectivity model may have critical effects on a WSN. For example, several algorithms (e.g., RNG and GG [15]) partition the topology in Figure 2 by removing the edge between nodes 1 and 3 because of the implicit assumption of the existence of an edge between nodes 2 and 3, which in fact is nonexistent due to a wall.

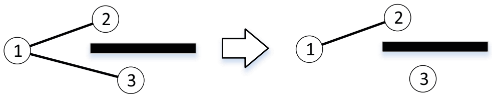


Fig. 2. Simple non-UDG topology being partitioned by GG and RNG

(ii) Despite significant technological progress in embedded system design, wireless sensors are still highly resource-restricted devices [35] with specific requirements. This important fact is considered insufficiently for the design of topology control. This results in questionable assumptions, like location awareness [19], round-based communication [24], a consistent view of the topology [15], reliable message passing [5] or the ability to conduct complex computations [14]. Most topology control algorithms acknowledge the infeasibility of gathering a global view of the network, restricting the local view to two hops [25], [19], [32]. Still, even this amount of local knowledge is infeasible in dense WSNs (e.g., office environments), where a local view of two hops may easily contain hundreds of edges. As WSN nodes, like the widespread TelosB [23], typically have only some few kBytes of RAM, storing the complete 2-hop view may lead to these nodes running out of RAM. This problem, which we call state space explosion, makes the successful execution of these algorithms impossible.

Symptomatic for the employment of unrealistic assumptions is that the vast majority of even the most recent topology control evaluations have not been carried out on real hardware but via simulation [2], [4], [5], [11], [14], [34]. This holds even true for algorithms focusing on harsh environments [34], leaving doubts about practicability in real-world deployments.

Still, some researchers have made efforts to enable topology control in testbed deployments. XTC [32] drops the UDG assumption at least partially. As small errors in the edge weight measurements of XTC may lead to network partitioning, kTC [24] adds a degree of robustness through aggregation of measurements. XTC and kTC have each been evaluated in a single testbed [3], [25]. Neither for XTC nor for kTC, it was shown whether a subset of graph properties proven only under the assumption of the UDG (e.g., planarity and bounded node degree) may actually be achieved in the testbed.

### B. Obstacle 2: Unsuitable Graph Structures

Topology control removes edges from the wireless topology to provide a specific target graph structure with properties like planarity, spanner, low-weight, etc. [20], [21], [24]. Typically, a certain graph property is expected to be desirable for one of two reasons. (i) First, the graph property is required by another network mechanism to correctly fulfill its functionality. While some graph properties (e.g., an undirected, connected output topology) are widely accepted as beneficial, other graph properties mainly have theoretical relevance. For example, graph planarity is primarily helpful for geographic routing [15], which has no practical relevance to date. (ii) The second motivation to provide a certain graph property is the expectation that the property facilitates general optimization metrics, like energy consumption [25], network lifetime [5] or throughput [11]. The practical use of these optimization metrics (ii) is unquestioned. However, neither for (i) nor for (ii), it is clear whether the raised expectations can be fulfilled in practice. Apart from this paper, results on achieving (i) solely rely on analytical proofs. Some few papers based on testbed deployments of topology control, including the paper at hand, investigate (ii), but thereby come to contradicting results. To the present day, it is unclear what graph properties can actually be achieved and thereby support today's communication stacks and applications.

### C. Obstacle 3: Application Agnosticism

Today, topology control algorithms are application-agnostic, i.e., they are designed for all-to-all communication. For example, kTC [24] and LSΘGG [20] focus on the generic broadcast and unicast patterns. In practice, WSNs are application oriented [35], and the applications are manifold, ranging from industrial automation [16] to volcano monitoring [33] and health tracking [12]. Consequently, WSN applications follow specific communication patterns, e.g., the many-to-one pattern with all nodes sending data to a common base station. Being application-agnostic, topology control algorithms might remove links that are favored by application-specific communication patterns, leading to severe implications for application performance [27]. Thus, we argue that topology control should be application-specific.

### D. Obstacle 4: Unclear Role in the Stack

In WSN devices, several mechanisms form an integrated communication stack. To the current day, the role of topology control with respect to other mechanisms (in particular those with similar functionality) in this stack is not yet finally clarified. For example, clustering or routing implicitly span a topology that might employ only energy-efficient links anyway (without the need for removing unsuitable links). Topology control might even downgrade the performance of other mechanisms in the stack (e.g., because topology control restricts the solution space of routing decisions). While different papers have proposed to integrate topology control with other mechanisms in the stack [1], [30], the interplay of topology control with other mechanisms in practical stack implementations has been investigated insufficiently.

### E. Obstacle 5: Missing Tool and Framework Support

Most pseudo code descriptions of topology control algorithms consist only of a few lines of code. Nevertheless, implementing such algorithms for their testbed-based evaluation is extremely time-consuming. The reason is that topology control, which essentially selects a subset of logical neighbors from the initial set of physical neighbors, depends on complex auxiliary functionality in the communication stack. For example, topology control relies on a topology abstraction component to hide removed network links from routing. Moreover, topology control may require smart transmission power control, efficient protocols and data structures to discover and store the local graph structure of the surrounding topology, etc. This is no default functionality of real-world communication stacks as, e.g., provided by Contiki. Although some few testbed-based implementations of topology control algorithms have been conducted [3], [25], no publicly available source code to build upon has been published for these evaluations. In other words, the framework and tool support for the implementation and evaluation of topology control algorithms on real hardware is insufficient. We believe that this lack of support hinders researchers from conducting practical evaluations of their topology control algorithms.

## IV. FRAMEWORK

We argue that the lack of appropriate frameworks and tools is a main reason why topology control algorithms are hardly ever evaluated on real hardware (Obstacle 5). Thus, this paper provides a corresponding publicly available[1] framework and toolset for the operating system Contiki[2]. The framework allows for the rapid implementation and testbed-based evaluation of new topology control algorithms for different hardware platforms, communications stacks, and applications.

---

[1] The source code is available at https://github.com/steinmic/TopologyControl.

[2] The implementation is based on Contiki 3.0, http://www.contiki-os.org/.

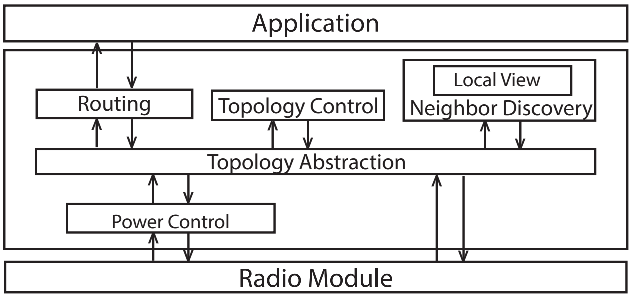


Fig. 3. Architecture of the proposed topology control framework

Figure 3 shows the main components and the packet flow among them. Most importantly, the topology abstraction component manages a node's local view of the topology. That is, this component hides removed links from the routing mechanism and the application. Packets received over removed links (e.g., via broadcast) are discarded. The neighbor discovery component provides the initial, graph-based local view (typically, two hops). This local view is used by the topology control component for providing the topology abstraction component with the set of removed edges.

The topology control, neighbor discovery and power control components have a considerable impact on the system performance. Thus, these components may be exchanged by the developer. In the following, we will describe the components in detail and provide an overview of the evaluation toolset.

### A. Topology Control

The topology control component selects a logical neighborhood based on a simple, generic interface. The component is provided with a graph-based local view of the topology, and the framework will handle the actual removing of links. This simple interface allows for implementing additional topology control algorithms in a few lines of code.

### B. Neighbor Discovery

Typically, topology control algorithms rely on a 2-hop view of the surrounding topology [19], [20], [24], [32]. To decrease the effort for adding new topology control algorithms, the neighbor discovery functionality was separated into its own component. Our (easily exchangeable) default implementation is based on a simple, broadcast-based protocol. In this protocol, each node regularly broadcasts its known 1-hop view. In dense networks, storing the local view, which essentially consists of node identifiers and edge weights, in a naive way may consume a tremendous amount of RAM. Network addresses are often referenced multiple times in a local view because WSN devices may be incident to more than one link. We exploit this redundancy by providing a simple address reference service. This service stores each network address once and provides locally unique references to this address. In the graph-based representation provided to topology control, these significantly smaller internal addresses are used as node identifiers, thus leading to a smaller local neighborhood table.

### C. Power Control

Topology control typically removes long-distance neighbors. Consequently, the remaining neighbors are close to the local node. Based on this resulting topology, most topology control papers propose to control the transmission power, typically with the goal of saving energy [5]. Topology control (i.e., the pure neighbor selection) is usually developed independent of power control, which has led to two widely independent but related research directions. This paper focuses on topology control. Nevertheless, power control may have a fundamental impact on the network performance [10]. For this reason, we provide four different (exchangeable) power control implementations in our framework, including the state of the art algorithm P-TPC [10]. Moreover, we provide some features facilitating the implementation of power control algorithms. Among others, this includes a sliding window approach to determine frequently used links, which allows power control to restrict power optimizations to these links.

### D. Evaluation Toolset

The framework provides the developer with some useful tools to support the specific requirements of topology control evaluations. A key function is to export a graph-based snapshot of the network topology including both the communication topology and the routing topology. In addition, the framework provides re-usable scripts for the automated analysis of graphs and visualization of relevant metrics.

## V. EVALUATION

Our testbed-based evaluation demonstrates Obstacles 1 to 4. We will first outline our experimental setup, followed by the discussion of our results.

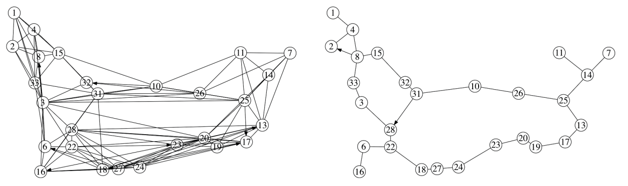
### A. Experimental Setup

In the following, we will describe the core parameters of our experimental study. Each configuration was run five times, and data points in the plots give the average over all runs. The experiment duration was set to 60 minutes.

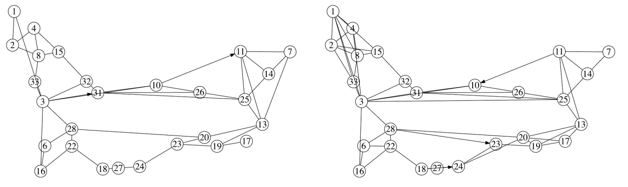
*1) Testbed:* All experiments were conducted in the Flocklab WSN testbed [22]. We used the testbed's indoor nodes based on the TelosB platform [23], which reflect typical resource-restricted WSN devices. The nodes are placed in an office environment, having the node distribution shown in Figure 4(a).

*2) Topology Control Algorithms:* As it is infeasible to compare a large number of algorithms in testbed environments, our evaluation focuses on the following selection of algorithms.

*a) LMST:* The LMST algorithm [19] is one of the most famous topology control algorithms (with roughly one thousand citations in August 2016). According to the theoretical analysis, LMST guarantees a connected output topology with a bounded node degree of 6. Executing LMST, each node constructs a minimum spanning tree based on a 2-hop local view of the topology. Each node then removes all edges not contained in this spanning tree. LMST uses the Euclidean distance between nodes as edge weights. As FlockLab nodes are unaware of their location, we explicitly provided the nodes



(a) NoTC (b) LMST



(c) a-kTC (d) l*-kTC

Fig. 4. Testbed snapshots of topologies provided by the different algorithms

with distance information. The testbed snapshot in Figure 4(b) shows a sample LMST topology.

*b) a-kTC:* We consider a-kTC [25] because it is a recent topology control algorithm with an explicit focus on practicality. The algorithm has been evaluated in the TUD$\mu$net WSN testbed [13] with respect to energy consumption and packet delivery. However, most graph properties (planarity, bounded node degree, and $\theta$-separation of neighbors) have only been proven based on the UDG model [24]. For a-kTC, each node constructs a 2-hop local view of the topology. We used the received signal strength indicator to provide edge weights. a-kTC aggregates the two edge weight measurements of bidirectional links to a single measurement. Then, in each triangle, a-kTC removes the longest edge if the edge weight is $k$ times larger than the shortest edge weight (here, $k = 1.2$), resulting in a topology as shown in Figure 4(c).

*c) l*-kTC:* We consider l*-kTC, a distributed variant of l-kTC [27], an application-specific topology control algorithm for many-to-one communication based on a-kTC. First, a-kTC marks edges for removal. Then, it is checked for each marked edge whether its removal may result in a hop count increase from an incident node to the communication target (e.g., the base station) of more than factor $a$ (here, $a = 1.5$). In this case, removal of the marked edge is prevented.

The original variant, l-kTC [27], operates globally and centrally on a graph-based model. As global knowledge and centralized execution are practically infeasible, we developed l*-kTC as a distributed protocol that operates solely based on locally available knowledge. Each node periodically broadcasts its hop count from the communication target. Based on this knowledge, whenever an edge is checked for removal from an identified triangle, the two incident nodes check independently whether using the two remaining edges of the triangle will violate the distance constraint given above. An l*-kTC topology snapshot is shown in Figure 4(d).

*d) NoTC:* We also consider a configuration without topology control as baseline (Figure 4(a)).

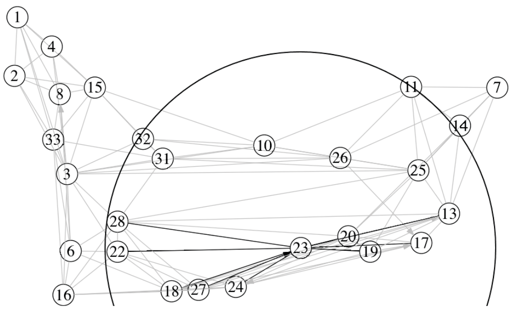


Fig. 5. Real topology snapshot in contrast to a UDG topology (for node 23)

*3) Algorithm Execution:* For each configuration but NoTC, the first 10 minutes served neighbor discovery to exchange required information, followed by the execution of topology control. As this is regularly proposed in the literature, we executed topology control (with exception of NoTC) in conjunction with power control: For each frequently used incident link, multiple test messages are sent with varying transmission power (according to a binary search scheme) to find a minimal power level that still ensures a packet reception rate of at least 80% for that link. Having determined a power value, the algorithm keeps this value for the rest of the experiment.

*4) Applications:* Each application introduces its own communication pattern, thus introducing specific requirements for topology control. We evaluated two applications reflecting different communication patterns: all-to-all and many-to-one. In the all-to-all application, each node randomly selects another node at the beginning of the experiment and then periodically sends a packet to this selected node. This enables communication between arbitrary pairs of nodes. In the many-to-one application, each node periodically sends a packet to node 1. This reflects a typical data collection scenario with exactly one sink node. For each application, the packet size was set to 50 Bytes, and the inter-packet time was uniformly distributed within $[25s, 35s]$.

*5) Communication Stack:* All experiments are based on the Rime stack. For multi-hop packet delivery, Contiki's Mesh routing protocol was used. On the MAC layer, we employed ContikiMAC [9], an energy-efficient radio duty cycling mechanism, in conjunction with the CSMA module, which performs retransmissions. We used the Energest component to estimate the energy consumption of the nodes.

### *B. Obstacle 1: Unrealistic Assumptions*

We shortly demonstrate that even comparably practical topology control algorithms like a-kTC and LMST suffer from unrealistic assumptions. For both algorithms, the theoretical analyses for some graph properties rely on a UDG. Figure 5 shows a snapshot of the FlockLab topology. The circle around the exemplarily chosen node 23 includes all nodes to which the node would have a link if the UDG model would hold true in practice, i.e., the circle's radius is equal to the Euclidean distance from the farthest neighbor. As expected, many links contained in the UDG are nonexistent in the snapshot. Other links are unidirectional, even for short geographical distances. This demonstrates that the UDG is an invalid assumption.

Figure 5 also demonstrates the state space explosion problem (Section III-A). Recall that most topology control algorithms take their decisions based on a locally stored copy of the 2-hop view of the topology. Although the shown office environment exhibits a rather low density, the 2-hop view of node 23 contains more than 30 links. Assuming that storing one link consumes 5 Bytes of RAM (2 Bytes for each Rime address and 1 Byte for each link weight), the storage of the 2-hop view requires in total more than 150 Bytes of RAM. In more dense environments, storing the local view may easily require more than 1 kByte of RAM, which reflects 10% of the RAM capacity of the widespread TelosB nodes [23]. This demonstrates that storing even only a 2-hop view contradicts a lightweight conception of topology control. We envision the specification of topology control algorithms based on a high-level language like TARL [26]. This will allow for automatically optimizing the size of the local view based on constraints of specific hardware platforms and environments.

### C. *Obstacle 2: Unsuitable Graph Structures*

Topology control aims at providing target graph properties. We will now evaluate whether these graph properties are achieved. Then, we will explore the impact on the application.

*1) Graph Properties:* Table I summarizes the graph properties promised and achieved by the considered topology control algorithms (based on five testbed runs). Undoubtedly, the most important graph property to be ensured by topology control is connectivity (i.e., a path between any pair of nodes exists). This property is achieved by each considered algorithm.

Another graph property often considered to be important is edge symmetry (i.e., each edge is bidirectional). Although the theoretical analyses of a-kTC and l*-kTC promise edge symmetry, this property cannot be guaranteed in the testbed. This can be explained by testbed dynamics, which prevent the nodes from having a consistent view of the topology.

On the other hand, LMST as well as a-kTC are able to provide the promised bounded node degree of 6 and 7, respectively. This is surprising as the corresponding theoretical analyses rely on a UDG. Also, the observed maximum node degrees of 4 and 6 are even smaller, showing that topology control is able to provide sparse but still connected topologies.

Nevertheless, it is important to state that a-kTC fails to provide $\theta$-separation (i.e., there is a minimum angle of $\theta$ between any pair of 1-hop neighbors in the output topology) and planarity (i.e., no edges are crossing), where the corresponding proofs are again based on a UDG. LMST and l*-kTC do not guarantee planarity and $\theta$-separation according to the theoretical analysis. Surprisingly, LMST nevertheless produced planar output topologies in our experiments, which may be explained by the low density of the output topology.

In summary, all considered topology control algorithms are able to provide the most important graph properties.

*2) Application Performance:* Section V-C1 shows that the considered algorithms actually achieve some of the desired graph properties in practice. A widespread expectation is that specific graph properties are associated with an improved network performance, thus supporting the application. To evaluate whether this is the case in our settings, we conducted experiments with two application scenarios reflecting different communication patterns: all-to-all communication and many-to-one communication. For both applications, we measured the application performance in terms of the energy consumption and the packet reception rate (as perceived by the application).

| | LMST [19] | a-kTC [25] | l*-kTC |
|---|---|---|---|
| Connectivity | **yes**<br>(yes) | **yes**<br>(yes) | **yes**<br>(yes) |
| Edge Symmetry | **-**<br>(-) | **no**<br>(yes) | **no**<br>(yes) |
| Max. Node Degree | **4**<br>(6) | **6**<br>(7) | **-**<br>(-) |
| $\theta$-Separation | **no**<br>(no) | **no**<br>(yes) | **no**<br>(no) |
| Planarity | **yes**<br>(no) | **no**<br>(yes) | **no**<br>(no) |

TABLE I
ACHIEVED (BOLD) AND PROMISED (GRAY IN PARANTHESES) PROPERTIES

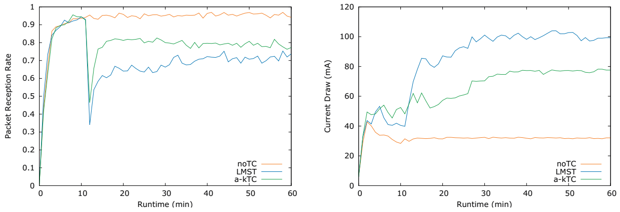
Figure 6 shows the application performance for all-to-all communication. Given performance values refer to the average over all nodes and are approximated for intervals of 1 minute. Unfortunately, neither a-kTC nor LMST improve the application performance (l*-kTC was not evaluated as it is application-specific for the many-to-one pattern). More precisely, the execution of each topology control algorithm leads to a reduction of the packet reception rate and to an increase of the energy consumption of the WSN devices compared to a stack configuration without topology control.

Similar results apply for the many-to-one application (Figure 7). Again, topology control performs worse than NoTC. Solely l*-kTC achieves an application performance close to NoTC. The reasons for this will be discussed in Section V-D.

These results should not be interpreted in the sense that topology control is always unable to support the application. In fact, two studies with positive observations on energy consumption [25] and throughput [3] exist. However, our results demonstrate the gap between promising theoretical results (with respect to achieved graph properties in simplified models) and metrics of interest in practical applications (e.g., energy consumption). Thus, just including existing topology control algorithms into today's communication stacks is unlikely to be successful. Instead, additional practical research will be required to better understand the effect of certain graph properties on relevant performance metrics.

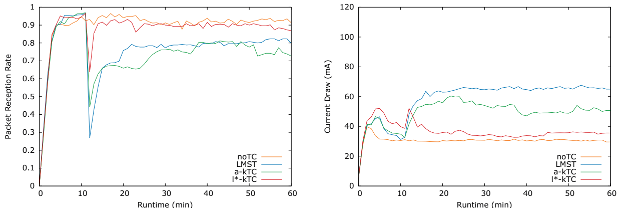
### D. *Obstacle 3: Application Agnosticism*

Our results show that executing topology control algorithms may even have a negative impact on the application performance. We argue that one reason for these disappointing results is that today's topology control is application-agnostic, i.e., application-specific communication patterns are not considered for edge removal decisions. To demonstrate



(a) Average packet reception rate (b) Average energy consumption

Fig. 6. Application performance for all-to-all communication



(a) Average packet reception rate (b) Average energy consumption

Fig. 7. Application performance for many-to-one communication

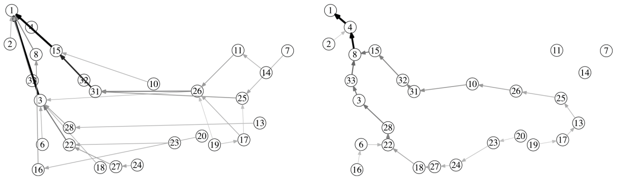
the impact of topology modifications on the many-to-one application, Figure 8 shows routing topologies constructed during exemplarily chosen testbed runs for each considered algorithm. The thickness of an edge reflects its packet load, increasing linearly with the number of application packets routed through this edge (normalized over all subfigures).

As shown in Figure 8(a), in the initial topology, each node reaches the base station within a few hops, resulting in a reliably high packet reception rate (Figure 7(a)). This picture changes fundamentally when topology control is executed. In the very sparse LMST topology, packets must travel many hops through the network before reaching the target node. This results in a poor application performance compared to NoTC because each additional hop includes the risk of packet loss. For a-kTC, at least some links directing to the base station are still available. However, only l*-kTC, which constructs a topological pattern where edges directing toward the base station are preserved, is able to provide a similar application performance as NoTC.

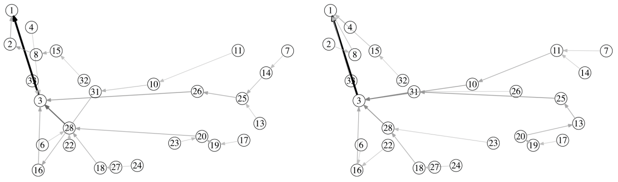
Not considering application-specific communication patterns has another negative impact: Executing topology control results in an unbalanced load distribution, which may be measured in terms of the energy consumption of individual nodes (Figure 9). In the initial topology, many redundant paths to the base station exist. Consequently, the energy consumption is fairly distributed (Figure 9(a)). LMST (Figure 9(b)) and a-kTC (Figure 9(c)) exhibit a much more unbalanced consumption pattern. In particular those nodes very close to the base station and those far away from the base station suffer from an increased overhead. Only l*-kTC provides a load distribution pattern comparable to NoTC (with exception of node 3).

### *E. Obstacle 4: Unclear Role in the Stack*

To demonstrate that topology control requires a smart integration into the communication stack, Figure 10 visualizes

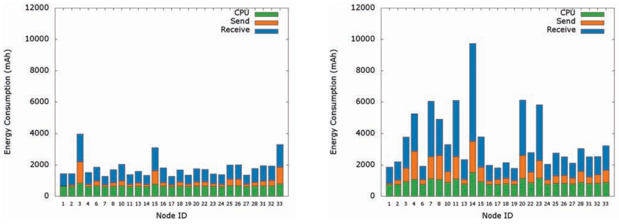


(a) NoTC (b) LMST

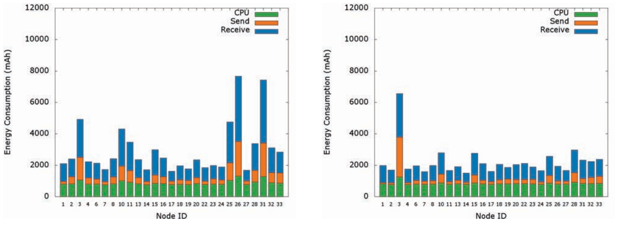


(c) a-kTC (d) l*-kTC

Fig. 8. Sample packet-count weighted routing graphs (many-to-one)



(a) NoTC (b) LMST



(c) a-kTC (d) l*-kTC

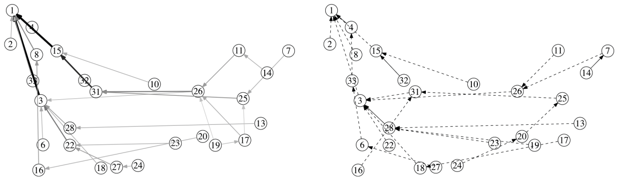
Fig. 9. Energy consumption for each node over five runs (many-to-one)

which edges of the routing overlay are removed by topology control. Actually, many edges used by routing are removed by each topology control algorithm and can thus not be used anymore. Apart from the problem of longer routing paths (Section V-D), this also introduces the problem of adding a new dimension of dynamics to the network because routing has to determine new routing paths. For all considered topology control algorithms, the tremendous drop in the packet reception rate after 10 minutes shows these immediate negative implications of edge removal on the routing performance.

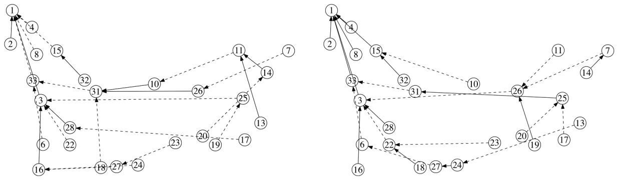
Our experiments focused on Contiki's Mesh routing protocol. However, our observations are not limited to this specific routing protocol, but generalize to all protocols that rely on state information on the underlying topology. This includes all but very simplistic routing schemes.

For the very sparse topology created by LMST, we even observed that routing was unable to find new paths in the new topology, although this topology was connected (Figure 9(b)).

To improve the interplay of topology control and routing,



(a) Sample routing graph for NoTC (b) Sample modifications by LMST



(c) Sample modifications by a-kTC (d) Sample modifications by l*-kTC

Fig. 10. Routing graph modifications for the algorithms (many-to-one). Dashed lines indicate edges to be removed by topology control.

one possible approach would be to actively inform routing on topology adaptations to enable proactive route discovery. Another promising approach would be to integrate topology control into routing, as proposed by various researchers [1], [30]. Both approaches have not yet been evaluated practically.

## VI. RELATED WORK

In the related work, different definitions of topology control exist. Some of these definitions refer to controlling arbitrary network parameters modifying the network topology in some way [1], [18], [36]. This paper refers to the traditional definition of topology control. That is, topology control is the process of selecting a logical neighborhood based on the physical neighborhood, typically (but not necessarily) complemented by power control. Many such topology control algorithms have been presented during the last decade. For a comprehensive overview, please refer to the survey by Wang [31]. This section focuses on giving an overview of topology control evaluations.

Most evaluations of topology control are simulation-based, thus leaving doubts concerning the practical relevance of their results. While practical (e.g., testbed-based) evaluation has become the usual method of evaluation in other fields of WSN research, e.g., power control [10], the majority of even the most recent topology control evaluations have been conducted analytically or via simulations [2], [4], [5], [11], [14], [34]. Fuchs et al. [11] conduct an extensive simulation-based evaluation study, comparing various different topology control algorithms. They conclude that topology control in networks with low node density increases the throughput and decreases the energy consumption in some cases. Unfortunately, the evaluation study is based on IEEE802.11g devices, which, in contrast to typical WSN protocols, decrease the data rate when packets are received with low power. This makes it difficult to transfer the interesting results to the WSN domain.

Practical evaluations of topology control algorithms are rare. The few existing practical evaluations focus on individual topology control algorithms, while the paper at hand analyzes and evaluates the concept of topology control in general. Moreover, in contrast to this paper, none of the existing testbed deployments of topology control evaluates the achieved graph properties. Schweizer et al. [25] deployed the a-kTC algorithm in the TUD$\mu$Net WSN testbed [13], for the first time demonstrating energy savings through topology control in a WSN testbed. The authors restrict the evaluation to the application performance, leaving the evaluation of the provided topological structure (and its achieved graph properties) open. Unfortunately, TUD$\mu$Net is not maintained anymore, which makes further evaluations impossible. In another study, Burri et al. [3] deployed the XTC algorithm [32] in an office environment and observed a positive impact on throughput, neglecting other relevant properties of the topology. Again, the sensor deployment is publicly unavailable, which makes reproducing the results difficult. Other existing studies referring to topology control do not define this term as a way to achieve desirable graph properties for the connectivity topology (see above), but focus on other aspects like power control [7] or routing overlay construction [6].

For none of the existing practical evaluations of topology control, the used source code has been published. This makes comparisons of the results difficult. In contrast, this paper provides an open source framework, supporting future implementations and evaluations of topology control algorithms.

In summary, this is the first paper that critically discusses and practically evaluates the general concept of graph-based topology control.

## VII. CONCLUSION

Various graph-based topology control algorithms have been proposed. Nevertheless, topology control has never made the breakthrough into practical deployments. This paper explores the reasons for this, identifying five practical obstacles of today's topology control: (i) unrealistic assumptions, (ii) unsuitable graph structures, (iii) application agnosticism, (iv) unclear role in the stack, and (v) missing framework support.

We provide a publicly available framework that will support researchers in developing and evaluating new topology control algorithms. Moreover, we provide the first distributed application-specific algorithm for topology control. Our testbed-based evaluation shows that all three considered topology control algorithms suffer from the identified practical obstacles, resulting in suboptimal application performance. However, this paper is the first to show that topology control achieves interesting graph properties in practice, which demonstrates the feasibility of topology control in principle.

Our framework and application-specific algorithm reflect contributions on obstacles (iii) and (v). In the future, we will consider the other obstacles. Tackling (ii), we will focus on the question which graph properties actually support specific communication patterns induced by modern communication mechanisms. Another interesting branch of research is the impact of emerging sensor generations with less strict constraints on energy, processing or storage (e.g., OpenMote [28]) on the feasibility of topology control.

## ACKNOWLEDGMENT

This work has been cofunded by the German Federal Ministry for Education and Research (BMBF, Software Campus project DynamicINP, grant no. 01IS12054), and by the German Research Foundation (DFG) as part of project A01 within the Collaborative Research Center 1053 – MAKI. The authors assume responsibility for the content.

## REFERENCES

[1] A. A. Aziz, Y. A. Şekercioğlu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 121–144, 2013.

[2] H. Bagci, I. Korpeoglu, and A. Yazc, "A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 914–923, 2015.

[3] N. Burri, P. von Rickenbach, R. Wattenhofer, and Y. Weber, "Topology control made practical: Increasing the performance of source routing," in *Mobile Ad-hoc and Sensor Networks*, ser. Lecture Notes in Computer Science (LNCS), J. Cao, I. Stojmenovic, X. Jia, and S. K. Das, Eds. Springer, 2006, vol. 4325, pp. 1–12.

[4] H. Chen and K. Shi, "Topology control for predictable delay-tolerant networks based on probability," *Ad Hoc Networks*, vol. 24, pp. 147–159, 2015.

[5] X. Chu and H. Sethu, "Cooperative topology control with adaptation for improved lifetime in wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2012, pp. 262–270.

[6] W. S. Conner, J. Chhabra, M. Yarvis, and L. Krishnamurthy, "Experimental evaluation of synchronization and topology control for in-building sensor network applications," in *Proceedings of the ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, 2003, pp. 38–49.

[7] P. Costa, M. Cesana, S. Brambilla, L. Casartelli, and L. Pizziniaco, "A cooperative approach for topology control in wireless sensor networks: Experimental and simulation analysis," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2008.

[8] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the IEEE International Conference on Local Computer Networks (LCN)*, 2004, pp. 455–462.

[9] A. Dunkels, "The ContikiMAC radio duty cycling protocol," Swedish Institute of Computer Science, Tech. Rep. T2011:13, 2011.

[10] Y. Fu, M. Sha, G. Hackmann, and C. Lu, "Practical control of transmission power for wireless sensor networks," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2012, pp. 1–10.

[11] F. Fuchs, M. Völker, and D. Wagner, "Simulation-based analysis of topology control algorithms for wireless ad hoc networks," in *Design and Analysis of Algorithms*, ser. Lecture Notes in Computer Science (LNCS), G. Even and D. Rawitz, Eds. Springer, 2012, vol. 7659, pp. 188–202.

[12] T. Gao, D. Greenspan, M. Welsh, R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, 2005, pp. 102–105.

[13] P. E. Guerrero, A. P. Buchmann, A. Khelil, and K. Van Laerhoven, "TUD$\mu$Net, a Metropolitan-Scale Federation of Wireless Sensor Network Testbeds," in *Proceedings of the European Conference on Wireless Sensor Networks (EWSN)*, 2012, pp. 1–2.

[14] J. Gui and Z. Zeng, "Joint network lifetime and delay optimization for topology control in heterogeneous wireless multi-hop networks," *Computer Communications*, vol. 59, pp. 24–36, 2015.

[15] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2000, pp. 243–254.

[16] A. Kumar Somappa, K. Øvsthus, and L. Kristensen, "An industrial perspective on wireless sensor networks – a survey of requirements, protocols, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.

[17] E. Lebhar and Z. Lotker, "Unit disk graph and physical interference model: Putting pieces together," in *Proceedings of the IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2009, pp. 1–8.

[18] M. Li, Z. Li, and A. V. Vasilakos, "A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2538–2557, 2013.

[19] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005.

[20] X.-Y. Li, W.-Z. Song, and W. Wang, "A unified energy-efficient topology for unicast and broadcast," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2005, pp. 1–15.

[21] X. Li, Y. Wang, P. Wan, W. Song, and O. Frieder, "Localized low-weight graph and its applications in wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2004, pp. 1–12.

[22] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "FlockLab: a testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proceedings of the ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN)*, 2013, pp. 153–165.

[23] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proceedings of the ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 364–369.

[24] I. Schweizer, M. Wagner, D. Bradler, M. Mühlhäuser, and T. Strufe, "kTC - robust and adaptive wireless ad-hoc topology control," in *Proceedings of the International Conference on Computer Communication Networks (ICCCN)*, 2012, pp. 1–9.

[25] I. Schweizer, R. Zimmermann, M. Stein, and M. Mühlhäuser, "a-kTC: integrating topology control into the stack," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, 2015, pp. 414–417.

[26] M. Stein, A. Frömmgen, R. Kluge, F. Löffler, A. Schürr, A. Buchmann, and M. Mühlhäuser, "TARL: Modeling topology adaptations for networking applications," in *Proceedings of the International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2016, pp. 57–63.

[27] M. Stein, G. Kulcsár, I. Schweizer, G. Varró, A. Schürr, and M. Mühlhäuser, "Topology control with application constraints," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, 2015, pp. 229–232.

[28] O. Technologies. Openmote-cc2538. [Online]. Available: http://www.openmote.com/hardware/openmote-cc2538-en.html

[29] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2, pp. 153–167.

[30] H. Üster and H. Lin, "Integrated topology control and routing in wireless sensor networks for prolonged network lifetime," *Ad Hoc Networks*, vol. 9, no. 5, pp. 835–851, 2011.

[31] Y. Wang, "Topology control for wireless sensor networks," in *Wireless Sensor Networks and Applications*, ser. Signals and Communication Technology, Y. Li, M. Thai, and W. Wu, Eds. Springer, 2008, pp. 113–147.

[32] R. Wattenhofer and A. Zollinger, "XTC: a practical topology control algorithm for ad-hoc networks," in *Proceedings of the IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2004, pp. 216–223.

[33] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2006, pp. 381–396.

[34] G. Xing, C. Lu, X. Jia, and R. Pless, "Localized and configurable topology control in lossy wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1345–1358, 2013.

[35] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[36] X. M. Zhang, Y. Zhang, F. Yan, and A. V. Vasilakos, "Interference-based topology control algorithm for delay-constrained mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 742–754, 2015.