

# Topology Control in Aerial Multi-Beam Directional Networks

Brian Proulx, Nathaniel M. Jones, Jennifer Madiedo, Greg Kuperman  
{brian.proulx, njones, jennifer.madiedo, gkuperman}@ll.mit.edu  
MIT Lincoln Laboratory  
Lexington, MA, USA

**Abstract**—In multi-beam directional networks, nodes are able to simultaneously transmit to all neighbors or receive from all neighbors. This spatial reuse allows for high throughputs, but in dense networks can cause significant interference. Topology control (i.e., selecting a subset of neighbors to communicate with) is vital to reduce the interference. Good topology control balances the number of links utilized to achieve fewer collisions while maintaining robust network connectivity.

In this work, we discuss the underlying challenges to topology control in multi-beam direction networks. Two topology control algorithms are developed: a centralized algorithm that retains robust connectivity at the cost of reduced throughput, and a distributed algorithm that offers higher throughput but with fewer links in the network. The performance of these algorithms is demonstrated by simulation using real beam patterns from a seven-element uniform linear array.

## I. INTRODUCTION

Directional communication systems offer many benefits over omnidirectional systems, such as increased spatial reuse, longer ranges, and in military networks, lower probability of detection and more resistance to jamming. New approaches to directional communication are increasingly becoming a reality due to recent advances in fully digital phased arrays [1]. In a fully digital array, each antenna element has an analog-to-digital converter behind it, allowing for precise control of the input and output beam patterns. This allows for simultaneously receiving (or transmitting) independent data streams in different directions. This multi-beam capability (i.e., simultaneous transmissions or receptions) allows for a dramatic increase in network capacity [2][3].

In addition to this new capability, emerging airborne networks are being developed as mobile ad-hoc networks (MANETs) [4], in which mobile nodes self-organize without infrastructure. As these nodes move through space, the topology of the network changes as new nodes become reachable, and previous connections are broken. This changing topology creates a challenge for directional networks. Although spatial reuse can be high, there is still interference in the system. For instance, though beams may be relatively narrow, multiple nodes can be located within the same main beam. In this case,

simultaneous transmission to all is impossible, only one of these neighbors can be transmitted to at a time. As an example, at a range of 50 nmi a 10 degree main beam covers over 60 nmi<sup>2</sup>.

This is the key challenge addressed in this paper: how to control the topology of a directional network by selecting neighbors in order to reduce the interference while still keeping as many links as possible. This problem has been well studied in omnidirectional networks [5], but many open directions of research exist for the directional case. Other works have addressed similar challenges, but with a slightly different goal. For instance in [6], an algorithm was presented to minimize the interference in the network by computing minimum degree spanning trees, resulting in low interference but also low connectivity. Additionally, the fully digital array allows for arbitrary beam pointing, as opposed to approaches considering sectored antennas. An example is [7], in which the connectivity and throughput are evaluated based on a sectored approach. Topology control with a degree constraint and the difficulty in finding optimal solutions has also been examined [8]. Others have focused on the effect of topology control on the number of hops between nodes, called hop stretch, and developed both near-optimal and more implementable solutions [9].

A similar approach has been taken, but with a different goal in mind. In [10], the beam directions are chosen to cover as many nodes as possible, assuming that there is a single flow to transmit to many neighbors. Conversely, we study the problem of independent flows to each neighbor, and thus seek to have as few nodes in the same beam as possible.

While high throughput is an important aspect in communications, robustness and a high node degree are also vital. A network with high throughput due to many links with low link utilization leads to long delays before packets are correctly transmitted and high interference. However, the high number of links results in fewer hops in a multi-hop flow, which may be desirable from a robustness standpoint. The tradeoff between throughput and connectivity is a function of the network traffic demand, which in general cannot be predicted, and the density of the nodes in the network. A highly dense network with a very low traffic load may not be well served by an aggressive topology control algorithm that removes many links.

This paper focuses on topology control with a goal of trading off between connectivity and throughput by presenting two

Distribution A: Public Release. This work is sponsored by the Assistant Secretary of Defense Research and Engineering via Air Force contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

algorithms to achieve different operating points. We begin by exploring the underlying phenomenology in order to develop an understanding of the main challenge. This results in the concept of a contention group, in which many links conflict with each other across multiple nodes. We start with a centralized solution that utilizes a full view of the network to attempt to untangle these contention groups in order to improve throughput while retaining a well-connected network. This algorithm results in a robust network with many connections but with potentially lower throughput. We also present a distributed algorithm, in which node coordination is minimized for ease of implementation. The distributed algorithm seeks to retain the shortest links in order to reduce interference, and then prune connections that interfere with these links. These two algorithms allow for a tailored approach based on the relative importance of throughput and connectivity. We simulate these two algorithms using the beam pattern from a seven-element uniform linear array, and compare against the original network and the distributed algorithm from [8].

The paper is organized as follows. In Section II, we present the assumptions about the underlying communications system. Next, in Section III the underlying challenges are examined, and then the two algorithms are presented in Section IV. The performance of these is discussed in Section V and the paper concludes in Section VI.

## II. SYSTEM OVERVIEW

In this work we consider a multi-beam directional communication system. In this section, we present a brief overview of the features of this system, but more detail can be found in [11]. At a glance, a fully digital antenna array is able to beamform in order to focus energy in certain directions on both transmit and receive. And rather than a single beam system, we consider a physical layer that can form an arbitrary number of beams on transmit and receive. For receive, an advanced post-processing capability allows for all directions to be processed simultaneously, combining the benefits of directional communications with the connectivity of omnidirectional systems. This allows each packet to have its own dedicated receive beam, pointed directly at the transmitter. Thus this system can be highly connected while offering the high gain and high throughput of a directional system.

Two key aspects to this system are power control and the channelization of the bandwidth. As we assume that every node knows the locations of its neighbors, the power for the transmit beam to that neighbor can be scaled to only that required to close the link. In order to protect against some low-level interference, each transmit power is increased a small amount above the minimum. This results in prioritizing shorter links, as the reduced transmit power results in lower interference for all other transmissions. Additionally, we assume that the bandwidth is channelized so that a number of orthogonal channels (i.e., non-overlapping frequencies) are available. For each transmit beam, one channel is picked at random from the available set. Two transmissions, even if they interfere in the main beam, are assumed to be completely orthogonal if

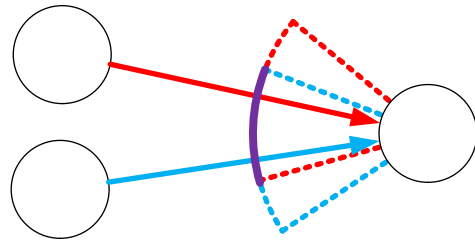


Fig. 1. Conflicting transmission example

they occur on different channels. Though the topology control algorithms assume only a single channel, we show later that having multiple additional channels results in significantly improved network performance, as would be expected with the additional bandwidth.

The topology control algorithms utilize the flattop antenna model for simplicity; see Figure 4 in [10] for an example. This model assumes a single **main beam** and outside of this main beam, the gain is negligible. Thus, for topology control in a multi-beam system, two nodes that are being simultaneously transmitted to or received from must not be in each other's main beam. Though the algorithms use this flattop approximation, the simulation results use the actual beam pattern of a seven-element uniform linear array with the actual side lobes. We denote the  $-10\text{dB}$  beam width as  $\theta$ , and for the seven-element uniform linear array the beam width is  $26$  degrees.

## III. PROBLEM FORMULATION

The main focus of this work is reducing the number of interfering communication links while retaining robust connectivity. Additionally, we assume that every node is interested in communicating with every other node, and if the two nodes are neighbors, then they will transmit backlogged flows to each other. Unlike some works mentioned before, each node has an independent data flow for all of its neighbors.

Next, some terminology is introduced. Choosing one node as a **central** node, two other nodes are within each other's main beam if the angle between them (centered on the central node) is less than one half beam width,  $\frac{\theta}{2}$ . In this case, these nodes are **contending** with each other when transmitting to the central node as both will be inside of the central node's receive beam for the other transmission. These two links form a **contention group**, that is a group that contains the links that contend with each other. Thus, each will experience enough interference that both transmissions will not be received. The reverse situation, where the central node is transmitting to the other two experiences the same phenomenon. Resolving the contention group is called **deconflicting**. Choosing to transmit to or receive from only one of these links results in correctly received packets. In dense networks, this overlap can be common and result in large percentages of failed transmissions. By not transmitting on some links, the overall network throughput can be improved as the increase in successful transmissions outweighs the reduced number of available links for communication.

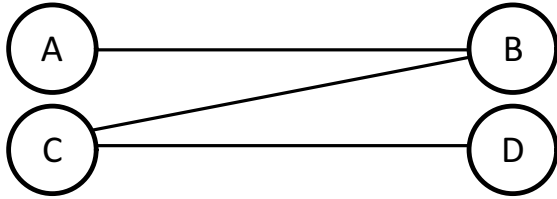


Fig. 2. Coupled links example

For an example of contending transmissions, see Figure 1. Here the red and blue transmissions are very close in angle at the receiver, the central node in this scenario. Due to this, each transmission is in the main receive beam of each other. The red arc represents the receive beam for the red transmission, and likewise for the blue shapes, and the purple line represents the overlap of the receive beams. If both of these transmissions occur simultaneously, then neither will be correctly decoded and both are failed transmissions. Due to symmetry, should the central node transmit simultaneously to both nodes on the left, then both of those transmissions would interfere with each other as they would be within each other’s main transmit beam.

Not only can links on a single node be conflicting, but also conflict groups can contain links among multiple nodes. When two links are contending, if one of those links contends with another link then those three links are part of a single contention group. These form a single group, not two separate groups, due to the fact that resolving the contention must be done jointly. If they stayed two separate groups, one group may keep one link, while the other may remove that link. Thus, they are one contention group because it must be considered jointly.

An example is shown in Figure 2. Here, edge AB conflicts with edge BC, and edge BC conflicts with edge CD. Although edges AB and CD do not directly conflict, they are coupled due to conflicting with edge BC, as any attempts to deconflict these edges individually will affect the other. That is, attempting to deconflict AB and BC affects attempts to deconflict BC and CD. This results in a contention group of {AB, BC, CD}. Though this example only spans three nodes, these conflict groups can be spread across many nodes in a network. In Figure 3, a 20 node network is shown, with each contention group a different color. The largest group, purple, contains 43 links spanning 14 nodes.

It should be noted that while this work considers only removing links from the network, many strategies are available for deconflicting networks. For instance, in a time division scheduled system, each link can be scheduled only with links that do not interfere with the transmission. Or in a multi-channel system, each link can be given orthogonal channels for transmission.

#### IV. ALGORITHMS

Next, we present two algorithms for topology control. The first is a centralized solution, while the second is a completely distributed algorithm, running locally on each node. Not relying on a centralized solution makes the algorithm very robust, but

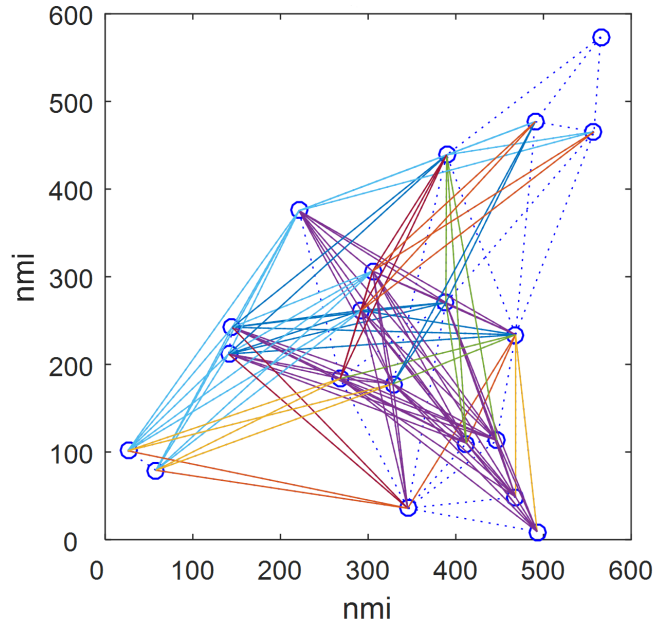


Fig. 3. Contention group example

utilizing only local information and decision making can result in a solution with worse performance. These two algorithms combine to span the space of solutions for topology control, from retaining many links for a robust network at the cost of more interference and lower throughput to removing many links, resulting in a less connected network but with higher overall throughput due to reduced interference. This allows for a choice of algorithm based on the desired operating point.

We assume that these algorithms are used on a homogeneous network, in which all nodes use the same communication hardware. In particular, this implies that there is a single beam width for each node, which is common for all transmit and receive beams formed by the array. These algorithms could be easily extended to cover the case in which different nodes could use different beam widths. For all of these algorithms, we denote the number of nodes in the network as  $N$ , the set of edges as  $\mathcal{E}$ , and the graph of the network as  $(N, \mathcal{E})$ .

##### A. Centralized Edge Delete

The first algorithm is **Centralized Edge Delete (CED)**. As the name implies, this algorithm runs on a central node with a complete and accurate map of all nodes in the network. The solution is then passed around the network, until all nodes are updated with their new list of neighbors. At a high level, this algorithm constructs a list of all contention groups, e.g. Figure 3, and then examines each edge in each contention group. If the edge in the contention group has a neighbor that is within its main beam, that is less than one-half beam width away in angle, then the original edge is removed from the network.

The first step of this algorithm is to find all of the contention groups. For readability, we break it out into its own subroutine, called **Find Contention Groups**. This process begins by

looping over each node and comparing the angle between any two of its neighbors to the beam width. If the angle between the two neighbors is less than one half of the beam width, then those edges are marked as contending edges and stored in a list denoted  $\mathcal{CE}$ .

Next, form contention groups from this list of contending edges. Start with the first entry, an edge, from this list and add it to an empty contention group, labelled  $CG$ . Next, loop over  $\mathcal{CE}$  and find all of the other edges that contend with the original edge and add them to the contention group. Then add all of the edges that contend with those new edges to the contention group. Continue this process until no new edges are added; the contention group is complete. Remove all of the edges in  $CG$  from  $\mathcal{CE}$ , and pick the next remaining edge from  $\mathcal{CE}$  and add it to a new, empty contention group. Continue this process until no edges remain in  $\mathcal{CE}$ . The output of this algorithm is the set  $\mathcal{CG}$  which contains all of the individual contention groups. This process is shown in detail as Algorithm 1.

---

**Algorithm 1:** Find Contention Groups

---

```

1 for  $i \in 1 : N$  do
2   if angle between any two neighbors ( $j, k$ ) at  $i < \theta/2$ 
3     then
4        $\mathcal{CE} = \mathcal{CE} \cup (i, j) \cup (i, k)$ 
5     end
6   end
7    $CG = \emptyset$ 
8   for  $e \in \mathcal{CE}$  do
9      $CG = \{e\}$ 
10     $NE = \{e\}$ 
11    do
12       $NE = \emptyset$ 
13      for any  $e' \in \mathcal{CE}$  do
14        if angle between  $e$  and  $e' < \theta/2$  then
15           $CG = CG \cup e'$ 
16           $NE = NE \cup e'$ 
17        end
18      end
19    while  $NE \neq \emptyset$ ;
20       $\mathcal{CE} = \mathcal{CE} \setminus CG$ 
21       $CG = CG \cup \{CG\}$ 
22    end
23  end
24 end

```

---

Having calculated all of the contention groups, the next step is to remove edges to deconflict the graph. As mentioned, this algorithm considers each edge in a contention group, and if there is another edge within one half beam width in the same contention group, the original edge is removed from the network. The detailed version of this is shown as Algorithm 2. The output of this algorithm is the set of all remaining edges  $\mathcal{E}$ .

### B. Distributed Distance Conflict Removal

To avoid the overhead associated with a centralized solution, we present a fully distributed algorithm, called the **Distributed**

---

**Algorithm 2:** Centralized Edge Delete

---

```

1 run Find Contention Groups
2 for  $CG \in \mathcal{CG}$  do
3   for  $e \in CG$  do
4     for  $e' \in CG \setminus e$  do
5       if angle between  $e$  and  $e' < \theta/2$  then
6          $\mathcal{E} = \mathcal{E} \setminus e$ 
7          $CG = CG \setminus e$ 
8       end
9     end
10  end
11 end

```

---

**Distance Conflict Removal (DDCR)** algorithm. In this, each node makes its own local decisions based on the knowledge of the locations of its neighbors. The key idea is to keep the shortest edges which do not interfere with each other (are within one half beam width of each other). This is accomplished via a greedy algorithm. First, the shortest link is added as a neighbor. Any links that are within one half beam width are then removed, as they would interfere. Next, the shortest remaining edge is added as a neighbor and again all conflicting edges are removed. Once there are no more edges to add, the algorithm is finished. The detailed version is presented as Algorithm 3, in which the set of all possible neighbors for a node is  $\mathcal{N}$ , and the output is the set of neighbors  $\mathcal{M}$ . The nodes that were not selected as neighbors are informed that they are no longer linked with this node.

---

**Algorithm 3:** Distributed Distance Conflict Removal

---

```

1 Sort  $\mathcal{N}$  by shortest distance
2 for  $n \in \mathcal{N}$  do
3    $\mathcal{M} = \mathcal{M} \cup n$ 
4    $\mathcal{N} = \mathcal{N} \setminus n$ 
5   for  $i \in \mathcal{N}$  do
6     if angle between  $n$  and  $i < \theta/2$  then
7        $\mathcal{N} = \mathcal{N} \setminus i$ 
8     end
9   end
10  Inform nodes not in  $\mathcal{M}$  that they are not neighbors
11 end

```

---

## V. RESULTS

In order to evaluate the performance of these algorithms, each was implemented in MATLAB and then the resulting graph was imported into the simulator described in [11]. In it, backlogged traffic is sent from every node to each of its neighbors. This is the most demanding traffic load, and may not be representative of realistic traffic flows, but it does provide an upper bound to network performance. The nodes are distributed uniformly at random in a square with a side length of 600 nmi, and the maximum communication range is set to 300 nmi. The burst

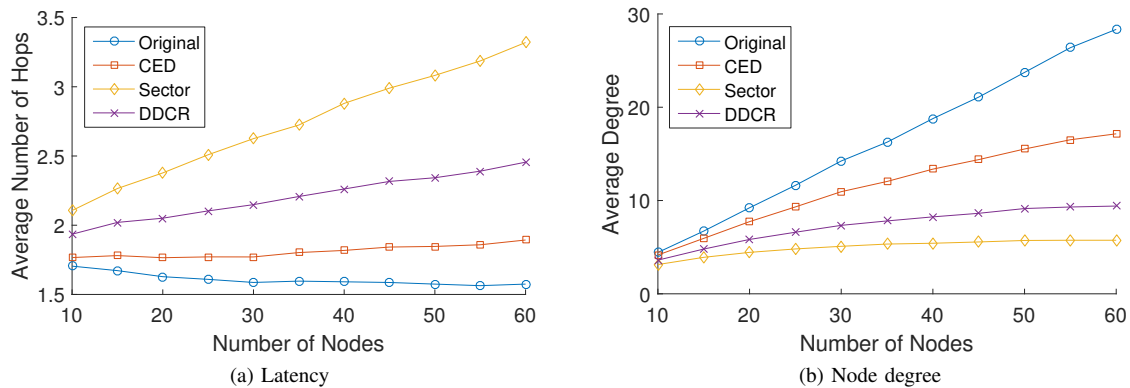


Fig. 4. Algorithm connectivity metrics

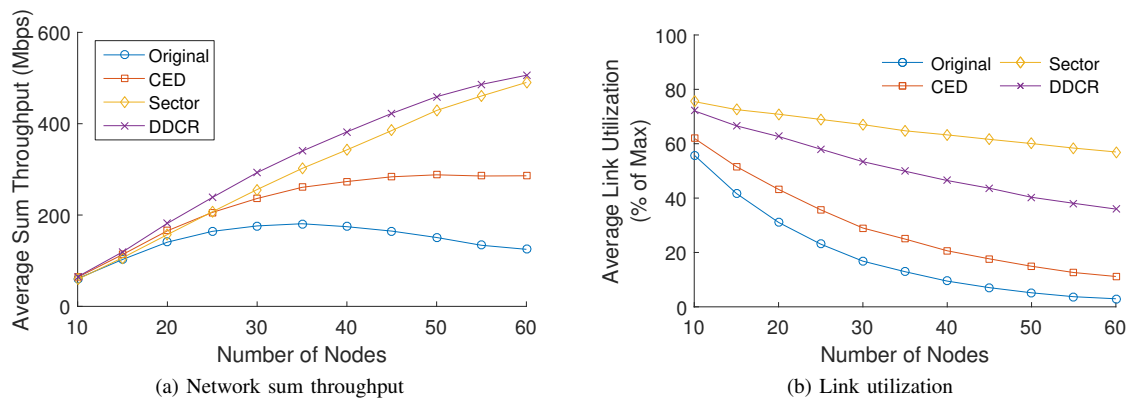


Fig. 5. Algorithm traffic metrics with one channel

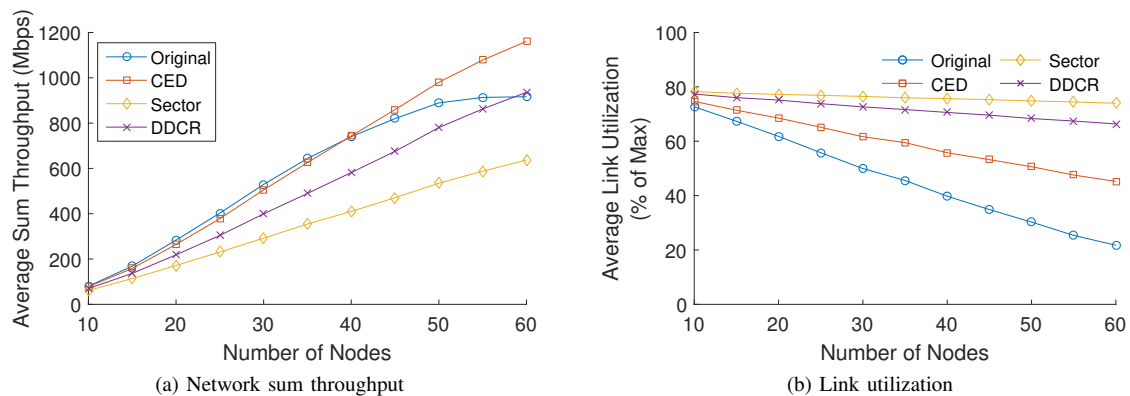


Fig. 6. Algorithm traffic metrics with five channels

rate of each link is 10 Mbps. All of these algorithms are tested by varying the number of nodes in the network, from 10 to 60, and the results are based on an average of 100 runs.

Several metrics allow for the evaluation of these different algorithms. A single metric is insufficient for characterizing the performance of topology control algorithms in this scenario due to the many trade-offs between different aspects of the network. For instance, a dense original network will be very strongly connected, but will have very high packet loss and result in low sum network throughput. Aggressive pruning of connections will result in very low packet loss, but at the cost of

a narrowly connected network, i.e. the failure of a single node could partition the network. Additionally, the density of the network plays a large role in the performance of the algorithm. Removing many links is necessary to thin very dense networks as there are many contending edges, but in a network that is already sparse this approach could remove too many edges.

In order to quantify these trade-offs, we use four metrics seen in the relevant literature, namely latency [10], node degree [9], sum network throughput [8], and packet loss [6]. The first two quantify the connectivity of the network. Having multiple paths to destinations protects against the sudden loss of some

links, important in challenging communication environments. The degree of each node represents the ability of the network to remain connected, the robustness of the network. Here, latency is not packet delay, but instead captures the fact that removing links from the network causes more multi-hop traffic. Nodes that were once directly connected may now have additional nodes to send data through in order to communicate. This metric is calculated by comparing the number of hops between each node in the network. As the network size is larger than the communication range, even the original network will require some multi-hop traffic.

The second two metrics, network throughput and packet loss, represent the overall data transmission ability of the resulting network. The sum network throughput is calculated by adding the throughput of every link in the network. Packet loss represents the robustness of the average link. Links with high packet loss require multiple retransmissions and interfere with other packets. So while the sum network throughput may be high, that may be due to having many links with a high loss rate compared to the low network throughput of a few very low packet loss links.

As an additional comparison, Algorithm 1 from [8] is included. This is a distributed algorithm that partitions the azimuth space into sectors and connects with the closest neighbor in that sector. Note that aligning the sectors is done arbitrarily; here we set the first sector as starting from the right side and proceed counterclockwise. The width of the sectors is also a choice. Setting the width to precisely the beam width allows for more sectors (and thus more neighbors), but this potentially results in more interference. For this comparison, we choose the sector size to be 1.5 times the beam width. Note that in the results, this algorithm is labelled "Sector".

The performance of these algorithms is shown in Figures 4 and 5, compared with the original, unmodified graph, labelled "Original". The latency results in Figure 4(a) show a clear preference towards the centralized algorithm, CED, over the decentralized ones. CED retains many more connections than the distributed ones, resulting in less multi-hop traffic. Also, Figure 4(b) demonstrates that the centralized algorithm has much higher node degree due to retaining more links. Additionally in both of these Figures, DDCR outperforms the Sector algorithm by generating a more strongly connected network.

The traffic metrics with a single channel system show that DDCR results in the highest sum network throughput, shown in Figure 5(a), which highlights the fact that the CED retains more connections, but these connections result in higher interference. This increased interference results in a lower network throughput than the less connected networks of DDCR and Sector. This can be seen in Figure 5(b), as the link utilization factor is much higher for the sparser networks of DDCR and Sector. The reduction in overall number of links is outweighed by the increased throughput of each link. Also, comparing the two distributed algorithms shows that DDCR outperforms Sector in throughput by having more links that operate at a slightly lower link utilization. So although more packets are lost, the increase

in the number of links results in a higher throughput.

For comparison, the traffic metrics for a system with five channels are shown in Figure 6. These demonstrate that CED performs the best when the contention is reduced by including more channels from which to choose. This demonstrates the utility of both CED and DDCR, as different networks require different topology control. In the single channel case, DDCR created the best network, and in the five channel case, CED performs the best.

## VI. CONCLUSION

In this paper, we studied topology control in a multi-beam directional network. The concept of a contention group was introduced, and the difficulty in resolving this contention was presented. Two algorithms were presented: the centralized CED maintained more robust connectivity whereas the distributed DDCR offered higher throughput. The performance of these algorithms was simulated using real beam patterns to highlight their relative strengths and weaknesses.

As shown in the results, the topology control problem changes when multiple orthogonal channels are available. This is a rich vein for future work, and could result in dramatically improved network throughput and robustness.

## REFERENCES

- [1] S. Y. Kim, O. Inac, C. Y. Kim, D. Shin, and G. M. Rebeiz, "A 76-84-ghz 16-element phased-array receiver with a chip-level built-in self-test system," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 8, pp. 3083–3098, Aug 2013.
- [2] R. B. MacLeod and A. Margetts, "Networked airborne communications using adaptive multi-beam directional links," in *2016 IEEE Aerospace Conference*, March 2016, pp. 1–7.
- [3] G. Kuperman, R. Margolies, N. M. Jones, B. Proulx, and A. Narula-Tam, "Uncoordinated mac for adaptive multi-beam directional networks: Analysis and evaluation," in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*. IEEE, 2016, pp. 1–10.
- [4] B. N. Cheng and S. Moore, "A comparison of manet routing protocols on airborne tactical networks," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, Oct 2012, pp. 1–6.
- [5] 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, Dec 2013.
- [6] U. Kumar, H. Gupta, and S. R. Das, "A topology control approach to using directional antennas in wireless mesh networks," in *2006 IEEE International Conference on Communications*, vol. 9, June 2006, pp. 4083–4088.
- [7] Z. Huang, C.-C. Shen, C. Srisathapornphat, and C. Jaikao, "Topology control for ad hoc networks with directional antennas," in *Proceedings. Eleventh International Conference on Computer Communications and Networks*, Oct 2002, pp. 16–21.
- [8] T. Stahlbuhk, B. Shrader, and E. Modiano, "Topology control for wireless networks with highly-directional antennas," in *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2016, pp. 1–8.
- [9] E. Gelal, G. Jakllari, S. V. Krishnamurthy, and N. E. Young, "Topology management in directional antenna-equipped ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 590–605, May 2009.
- [10] Z. Huang and C.-C. Shen, "Multibeam antenna-based topology control with directional power intensity for ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 508–517, May 2006.
- [11] B. Proulx, G. Kuperman, N. M. Jones, and T. Goff, "Simulation and modeling of a new medium access control scheme for multi-beam directional networking," in *2017 IEEE Aerospace Conference*, March 2017, pp. 1–9.