

Published in final edited form as:

Ad Hoc Netw. 2014 February 1; 13(Pt A): 153–169. doi:10.1016/j.adhoc.2011.04.011.

On Channel-Discontinuity-Constraint Routing in Wireless Networks[☆]

Swaminathan Sankararaman^{a,1}, Alon Efrat^{a,1,*}, Srinivasan Ramasubramanian^b, and Pankaj K. Agarwal^{c,2}

^a Department of Computer Science, The University of Arizona, Tucson, AZ, USA

^b Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA

^c Department of Computer Science, Duke University, Durham, NC, USA

Abstract

Multi-channel wireless networks are increasingly deployed as infrastructure networks, e.g. in metro areas. Network nodes frequently employ directional antennas to improve spatial throughput. In such networks, between two nodes, it is of interest to compute a path with a channel assignment for the links such that the path and link bandwidths are the same. This is achieved when any two consecutive links are assigned different channels, termed as “Channel-Discontinuity-Constraint” (CDC). CDC-paths are also useful in *TDMA* systems, where, preferably, consecutive links are assigned different time-slots.

In the first part of this paper, we develop a t -spanner for CDC-paths using spatial properties; a sub-network containing $O(n/\theta)$ links, for any $\theta > 0$, such that CDC-paths increase in cost by at most a factor $t = (1 - 2 \sin(\theta/2))^{-2}$. We propose a novel distributed algorithm to compute the spanner using an expected number of $O(n \log n)$ fixed-size messages.

In the second part, we present a distributed algorithm to find minimum-cost CDC-paths between two nodes using $O(n^2)$ fixed-size messages, by developing an extension of Edmonds’ algorithm for minimum-cost perfect matching. In a centralized implementation, our algorithm runs in $O(n^2)$ time improving the previous best algorithm which requires $O(n^3)$ running time. Moreover, this running time improves to $O(n/\theta)$ when used in conjunction with the spanner developed.

Keywords

algorithms; spanners; routing; directional antennas

[☆]A Preliminary version of this paper appeared in [1].

© 2011 Elsevier B.V. All rights reserved.

* Corresponding Author. Postal Address: Department of Computer Science, Gould-Simpson Building, The University of Arizona, Tucson AZ 85721-0077. Tel: (520) 626-8047. Fax: (520) 621-4246. alon@cs.arizona.edu.

¹A.Efrat and S.Sankararaman are supported by NSF CAREER Grant 0348000.

²P.K.Agarwal is supported by NSF under grants CNS-05-40347, CCF-06-35000, IIS-07-13498, and CCF-09-40671, by ARO grants W911NF-07-1-0376 and W911NF-08-1-0452, by an NIH grant 1P50-GM-08183-01, by a DOE grant OEG-P200A070505, and by a grant from the U.S.–Israel Binational Science Foundation.

swami@email.arizona.edu (Swaminathan Sankararaman), srini@ece.arizona.edu (Srinivasan Ramasubramanian),

pankaj@cs.duke.edu (Pankaj K. Agarwal)

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

1. Introduction

Wireless infrastructure networks (WINs) are gaining prominence as they are being increasingly deployed in metro areas to provide ubiquitous information access [2]. WINs provide a low-cost scalable network, support broadband data, and allow use of unlicensed spectrum. WINs have a wide area of applications, including public internet access [2], PORTAL [3], video streaming [4], and underground mining [5]. In order to increase the bandwidth in the WINs, nodes employ multiple wireless transceivers (interface cards) to achieve simultaneous transmission/reception over multiple orthogonal channels.

Recent research has focused on effectively harvesting the available bandwidth in a wireless network. Wireless interference is the key factor that limits the achievable throughput. The interference is encountered in two ways: (1) a node may not receive from two different nodes on the same channel at any given time; and (2) a node may not receive and transmit on the same channel at any given time. Moreover, if omnidirectional antennas are employed, then there may be at most one node transmitting on a channel in the vicinity of a node that is receiving on that channel. In a wireless network, a flow is defined as packets transmitted along a single path between a source-destination pair. At any given time, multiple flows may be present in the network. Based on this, the interference constraints in wireless networks may be divided into two categories [6]: *inter-flow* and *intra-flow*. The inter-flow interference refers to the scenario where two links belonging to two different flows cannot be active (on the same channel) at the same time as one receiver will experience interference due to the other transmission. The intra-flow interference refers to the scenario where two links belonging to the same flow cannot be active (on the same channel) at the same time. The same problem arises also in the TDMA setting, where a node can be used for streaming applications, but has to receive and transmit messages in different time slots.

The problem of routing and channel assignment in WINs refers to computing paths and channel assignment on the paths such that there is no inter-flow and intra-flow interference. If the bandwidth of a link is B , then the end-to-end throughput on the path is also B as all the links in the path can be active simultaneously. The problem of joint routing and channel assignment is challenging when nodes employ omnidirectional antennas and hence, it is typically solved as two independent sub-problems. For a given set of calls where the routing is known, the problem of channel assignment may be mapped to the distance-2 vertex and edge coloring problems [7]. The distance-2 coloring problems refer to coloring the edges or vertices such that two edges or vertices which are two hops or less from each other cannot have the same color. Thus, the objective is to compute a channel assignment, i.e., coloring of the edges satisfying the limit on the number of transceivers at each node. Distance-2 vertex and edge coloring problems are well known NP-hard problems and hence, the problem of channel assignment for networks employing omnidirectional antennas is challenging. Various approximation algorithms and heuristics have been developed for distance-2 coloring with different objectives, such as minimizing interference, maximizing throughput, and minimizing the number of required channels with combinations of these objectives [8, 9, 10, 11, 12, 13, 14]. In [15], the authors develop an approach based on balanced incomplete block design to assign channels such that the communication network is 2-edge-connected with minimum interference. In [16], a heuristic based on random channel assignment policy is developed to maintain connectivity of the network.

Among the works that compute paths for multi-channel networks, [17] and [18] take into account factors such as link loss rate, link data rate, channel diversity and, in the latter work, inter-flow interference in developing a new metric to be used in a routing protocol to find efficient paths with low interference.

When considering static traffic, [19] and [20] propose a complete system which handles routing and channel assignment separately. They aim to maximize throughput and do not provide any theoretical bounds on the performance of their system. [21, 22, 23] provide centralized and distributed schemes to provide routing and channel assignment in order to maximize throughput in the network for a given static traffic with guarantees on the performance. Again, these works handle routing and channel assignment separately and this leads to sub-optimal results. Finally, [24] considers the objective of achieving a given data rate under a simplified model of interference and provides approximation guarantees.

It has been shown in [25, 26, 27] that the capacity of WINs may be further improved by increasing spatial reuse by employing directional antennas. See Figure 1 for an illustration of the transmission range of directional antennas. The problem of channel assignment in networks employing directional antennas may be mapped to the edge-coloring problem [28]. In [29], a network architecture with nodes employing non-steerable directional antennas is developed. The authors develop approaches for routing and channel assignment by considering tree-based topologies rooted at “gateway” nodes.

There are indications that the problem of finding a route and channel assignment for the links along the route such that all links can be active simultaneously is NP-complete when the nodes employ omnidirectional antennas. Our algorithms are designed for the frequent cases for which the effect of interference between independent links can be minimized. This is due to the fact that because of increased spatial reuse and with good network planning regarding node placement and beam direction/width, the effects of both inter-flow interference and intra-flow interference between independent links is mitigated. The possibility of employing careful planning in deploying wireless networks using directional antennas is clearly increasing as their prices and accuracy are rapidly improving.

Topology Control

A fundamental technique used to speed up network operations is the use of spanners to perform topology control. These are subnetworks which make use of the spatial properties of the network in order to make the network operations more efficient. A subnetwork of the network is termed as a t -spanner if it maintains connectivity of the network and, the route between any two arbitrary nodes is longer than the original shortest route by at most a factor of t . There has been extensive research on spanners in the general setting (see [30, 31, 32, 33]). These works do not consider the interference constraints explicitly but instead try to achieve other properties such as planarity or bounded degree. In the case of networks employing directional antennas, it is interesting to consider the approach of controlling the width and orientation of the directional antennas. Such methods would aid in network planning to satisfy various criteria (e.g., guaranteeing the existence of high throughput paths while making the process of finding routes more efficient). Some related works are [34], in which the authors consider controlling the degree of a node by orienting the directional antennas, [35], which deals with satisfying robustness of the network, and [36], where spanning properties of the graphs induced by directional antennas with beamwidths approximated by circular sectors are analysed.

Channel-Discontinuity-Constraint

To the best of our knowledge, the first work addressing the problem of joint routing and channel assignment in such scenarios is the paper by Ahuja et. al. [6]. In such networks, the interference may be limited to primary interference between adjacent links (those that share an endpoint). Thus, the throughput along a flow is maximized when no two adjacent links along the path employ the same channel, i.e., the path bandwidth will be the same as the link

bandwidth. This leads to the definition of the “Channel-Discontinuity-Constraint” (CDC) on a path consisting of a flow: No two consecutive links can be assigned the same channel.

Any path which satisfies this constraint is termed as a CDC-path. In graph theory literature, CDC-paths are referred to as “*properly edge-colored*” paths, where the colors correspond to channels in under CDC. The main utility of this constraint is that any algorithm such as the one from [6] which computes a CDC-path performs “truly joint” routing and channel assignment in that the computation of the path automatically gives a channel assignment for the links along the path, i.e., no separate process is necessary. Moreover, as mentioned earlier, the bandwidth is maximized along this path. Finally, the algorithm of [6] works also in the case of weighted networks where weights on the links indicate power assignment, distance or other optimization factors. In this case, the algorithm provides an optimized (minimum-weight) path with maximum bandwidth (satisfying CDC).

From now on, whenever it is mentioned that a CDC-path is found, it is implied that a channel assignment for this path is also found.

We can overcome, to some extent, the difficulties of intra-flow interference when using omnidirectional antennas using the known technique of *physical layer network coding*. Consider a streaming application, where node s_i forwards the message $\langle m_j \rangle$ to node s_{i+1} , and receives the message $\langle m_{j+1} \rangle$ at the same time from s_{i-1} . Meanwhile, node s_{i+1} forwards a message $\langle m_{j-1} \rangle$ received earlier from s_i to s_{i+2} . In the case where omnidirectional antennas are used and the links satisfy CDC, it is quite possible that the channel on which both s_{i-1} and s_{i+1} transmit is f_1 , as it is different than the channel f_2 used by s_i (see Figure 2). However, as demonstrated in [37], since s_i “knows” $\langle m_{j-1} \rangle$ it can subtract it from the combined signal $\langle m_{j-1} \rangle + \langle m_{j+1} \rangle$. See [37] for details. However that is not a trivial hardware modification, so we concentrate in this paper mostly with other applications where CDC-paths are useful.

CDC-Paths and Time-Division-Multiplexing (TDM)

CDC-paths are also quite useful in the context of a Time-Division-Multiplexed (TDM) or Frequency-Division-Multiplexed/Time-Division-Multiplexed (FDM/TDM) system, where the channel may be thought of as a time slot or a combination of frequency and time slot. In TDM systems, it is desirable to assign different time slots to adjacent transmissions and this corresponds exactly to the Channel-Discontinuity-Constraint with time slots instead of channels. Thus, the throughput is maximized when two adjacent links along a path are using different time slots. Now, because of other “conversations” which have ended, new time slots may become available and finding new CDC-paths may help maintain the maximum bandwidth. So, the notion of channels in a FDM system considered in the CDC is equivalent to the time slots in a TDM system in that we are interested in finding paths such that no two adjacent links are scheduled on the same time slot.

Our Contributions

In this paper, we present two major contributions. The first contribution is a method to perform topology control by means of a spanning subnetwork consisting of $O(n/\theta)$ links (where n is the number of nodes and θ is a tunable parameter) for the problem of CDC routing. In this subnetwork, the minimum-weight CDC-path between 2 nodes may be longer than the original minimum-weight CDC-path by at most a factor t (termed as the stretch factor). Our spanner is based on the Yao graph [38], i.e., we divide the region around each node into a minimal number of disjoint sectors of angle at most θ , and we connect to a few neighbors in each sector. θ may be tuned in order to obtain better stretch factor at the expense of running time. This is particularly applicable when using directional antennas. For

example, if the antenna's beamwidth can be abstracted as a sector of angle 30 degrees, then by considering only a few neighbors in this range, we retain connectivity and the weights of the routes is increased by at most a factor of 4.3. This spanner is also highly dynamic, so insertion/deletion of links is carried out easily. To construct the spanner, we propose a novel distributed algorithm based on the ideas of Arango et.al. [39]. The main advantage of the spanner is that while being easily implementable when using directional antennas, it improves the time required to compute CDC-paths to $O(n)$ with the algorithm outlined below. This is a significant improvement over the algorithm of [6] which has a running time of $O(n^3)$ in the worst case. To the best of our knowledge, this is the first work which deals with topology control where we explicitly consider interference constraints.

The second contribution is a distributed algorithm for finding the shortest CDC-path between two given nodes by developing an extension of Edmonds' algorithm for minimum-cost perfect matching. The algorithm requires the exchange of $O(n^2)$ fixed-size messages in total ($O(n)$ per node). In a centralized implementation, our algorithm runs in $O(n^2)$ time improving upon the previous algorithm presented in [6] which requires $O(n^3)$ running time. Moreover, as mentioned above, with the use of the spanner, the algorithm's running time is improved to $O(n)$ when implemented in a centralized manner and the CDC-paths discovered are only longer than the shortest ones by a factor of t .

The rest of the paper is organized as follows. In Section 2, we discuss the network model and interference constraints. In Section 3, we discuss a t -spanner for WINs containing $O(n)$ links which may be used to achieve significantly faster routing. In Section 4, we present a distributed algorithm for finding a single CDC-path between two nodes s and d which requires the sending of a total of $O(n^2)$ messages. In Section 5, we present simulation results on our algorithm. Finally, Section 6 presents our conclusions.

2. System Model and Problem Formulation

We consider multi-channel multi-radio wireless networks modeled as a connectivity graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ where \mathcal{N} is the set of nodes and \mathcal{L} is the set of links. Let n and m denote the cardinalities of \mathcal{N} and \mathcal{L} respectively. Each node is equipped with C transceivers each of which may be tuned to a number of orthogonal channels. We assume a possibly heterogeneous network in which the channels available at a node may differ from other nodes such as when using both 802.11a and 802.11b in the network. Let the number of channels at a node u be $\mathcal{C}(u)$ and the cost of a link (u, v) be $w(u, v)$. The cost of a path P , denoted by $d(P)$ is the sum of the cost of the links along the path. For the purposes of topology control, we assume that each node is given a location in \mathbb{R}^2 . A link between two nodes is present in \mathcal{G} if they are within transmission range of each other.

When employing directional antennas, a careful placement of the nodes and orientation of the antenna beams results in WINs may be designed such that the number of links that can be active simultaneously between multiple flows can, for the most part, be maximized. However, there may still be dependence upon non-adjacent links. As explained in [6], a node z is said to be *collinear* with a transmission from nodes u to v if z cannot receive on the same channel that is used by the transmission from u to v . The interference at node z due to the transmission from u to v results in signal-to-noise ratio that is lower than the threshold for decoding the received signal. This collinearity constraint results in the dependence on channel assignment between links that are not adjacent to each other (thus resulting in distance-2 coloring problems). When employing directional antennas, the locations of nodes u , v and z need to be approximately collinear geometrically due to the constrained beam of directional antennas. Hence, the problem is of significantly lower impact than when using omnidirectional antennas when the interference regions are larger. Again, by carefully

planning the placement of nodes and orientations of the antenna beams, this constraint may be eliminated completely (or at least, significantly reduce).

Interference Model

In such carefully planned wireless infrastructure networks, there are only two interference constraints: **(1)** a node may not receive on a channel from more than one node at the same time; and **(2)** a node may not transmit and receive on a channel at the same time. These two constraints may lead to intra-flow interference, however it is limited to only two adjacent links. In order to avoid intra-flow interference, no two consecutive links in the path are assigned the same channel.

Using the above model of interference, we get the following definition:

Definition 1 (**Channel-Discontinuity-Constraint**)

A path \mathcal{P} together with a channel assignment on the links of \mathcal{P} satisfies the ‘‘Channel-Discontinuity-Constraint’’ (CDC) if no two adjacent links are assigned the same channel. We call a path a **CDC-path** if we can assign channels to each link (u, v) along this path such that CDC is satisfied.

We make the following assumptions on the network:

- We assume that the list of channels that a node can transmit/receive on is known at all times. The channels available on a link is the set of common channels between the two constituting nodes.
- Every call is assumed to have a bandwidth requirement of one channel capacity. We assume that calls are bidirectional and the same channel will be shared in both directions on a link.
- We also assume that the communications required for the algorithms need not satisfy CDC, i.e., interference may not be taken into account since, typically, a control channel is used to transmit network-related information rather than the actual data transmissions and the control messages are also typically very short.

Under the above conditions, we provide algorithms for two important problems in wireless networks. First, we need the following definition.

Definition 2

We say that a sub-network $\mathcal{G}'(\mathcal{N}, \mathcal{L}')$ where $\mathcal{L}' \subseteq \mathcal{L}$ is a **CDC t -Spanner** of \mathcal{G} if and only if for every $u, v \in \mathcal{N}$, $d(\mathcal{P}_{\mathcal{G}'}, (u, v)) \leq t \cdot d(\mathcal{P}_{\mathcal{G}}, (u, v))$ where $\mathcal{P}_{\mathcal{G}}$ and $\mathcal{P}_{\mathcal{G}'}$ are the shortest CDC-paths between u and v in \mathcal{G} and \mathcal{G}' respectively and t is a constant. The constant t is termed as the **Stretch Factor** of the spanner.

First, we handle the problem of performing topology control (by constructing a sub-network \mathcal{G}' which is a CDC t -spanner)

Problem 1 (CDC t -Spanner)

Given a connectivity graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ with n nodes having locations in \mathbb{R}^2 and the channels available at each node u , $\mathcal{C}(u)$, construct a sparse subgraph of \mathcal{G} , $\mathcal{G}'(\mathcal{N}, \mathcal{L}')$ which is a CDC t -spanner of \mathcal{G} and the number of links $m' = |\mathcal{L}'|$ is $O(n)$.

It is important to note that spanner is sparse in the number of links. This property is especially useful when considering the centralized implementation of the algorithm of [6] and our improved algorithm for the same problem. In the first case, it reduces the running time to $O(n^2)$ in the worst case and in the latter, we get a $O(n)$ algorithm in the worst case.

Problem 2 (Shortest CDC-path)

Given a connectivity graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ with n nodes and m links, the channels $\mathcal{C}(u)$ available at each node u , and costs on the links in \mathcal{L} , and a source-destination pair (s, d) , find a distributed algorithm to construct the minimum-cost CDC-path in \mathcal{G} between s and d .

The algorithm for Problem 2 which we develop may also be implemented in a centralized manner with improved running time over [6].

3. Spanner for CDC Routing

We are given the connectivity graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ with n nodes and m links together with locations of the nodes in \mathbb{R}^2 . We are also given the set of channels $C(u)$ available at each node u .

It is worth mentioning here that connectivity of the network does not imply that a CDC-path exists between any two nodes. For example, in the graph of Figure 3, the network is connected but there is still no CDC-path between nodes s and d and hence, we define the cost of the shortest CDC-path between s and d to be ∞ .

Definition 3

Let $T = \{T_1, \dots, T_p\}$ denote a partition of nodes in $\mathcal{T} = \{T_1, \dots, T_p\}$ into maximally disjoint sets such that $\mathcal{C}(u) = \mathcal{C}(v)$ for every $u, v \in T_i$. We define the **type** of a node u , denoted by $T(u)$, to be the set T_i containing u .

Hence, the number of types $p \leq 2^c$, where c is the number of channels over the network. Lemma 5 below shows that we may consider a much smaller number of types but for simplicity of exposition, we present the spanner using this bound and defer the claimed improvement to Lemma 5.

Spanner Construction—The CDC t -Spanner $CDCYG_k(\mathcal{N}, \mathcal{L}', \mathcal{C})$ is an extension of the Yao graph [38]. First, we describe how to create the set of links \mathcal{L}' . For each node $v \in \mathcal{N}$, divide the region around v into $k = \lceil 2\pi/\theta \rceil$ interior-disjoint sectors centered at v with opening angle θ . In each sector, connect v to its two nearest nodes of each type $T \in \mathcal{T}$ and connect these two nodes (see Figure 4a). This process can be performed in a distributed manner as described later. We prove that this graph is a CDC t -Spanner for the graph \mathcal{G} where $t = (1 - 2 \sin(\theta/2))^{-2}$. This is particularly applicable when directional antennas are used where the range of transmission of the directional antenna may be abstracted as a sector. Also, the computation of the above spanner does not require that we know in advance the specific source s and destination d between which a CDC path needs to be computed.

It is worth noting is that in a homogeneous network (all nodes have the same set of channels), the problem is much more simplified. In this case, all nodes have the same type and as a result, the Yao graph may be employed without any modification (connect only to the nearest neighbor in each sector). If all nodes are equipped with two or more channels, the cost of a path between two nodes is increased by only a constant factor and channels

assignment is easily accomplished by alternating between two channels along the path. However, the problem is more complicated in the heterogeneous network.

Remark 1

We note that no sub-quadratic size spanner can accommodate the degenerate case where the shortest CDC path between two nodes s and d is only one edge. For example, in the case where s and d are in two clusters A and B and there is only one channel shared between nodes in A and nodes in B (see Figure 4b), all edges from A to B may be required. However, no routing algorithm is needed here since s can simply check to see if d is within range and transmit to it.

When not considering the degenerate case mentioned in Remark 1, we obtain the following theorem:

Theorem 1

CDCY G_k is a CDC t -Spanner of \mathcal{G} for CDC-paths of more than one hop with $|\mathcal{L}'| = O(n/\theta)$ links where $t = (1 - 2 \sin(\theta/2))^{-2}$.

The following lemma was proven by Ruppert and Seidel [40] as a part of their proof of the stretch factor of the Yao Graph. This property is used repeatedly in the proofs of the stretch factor of *CDCY G_k* .

Lemma 1. (From [40, Theorem 1])

If there are two nodes q and r in a sector whose apex is p and $|p - q| \geq |p - r|$, then

$$|q - r| \leq |p - r| - \left(1 - 2 \sin\left(\frac{\theta}{2}\right)\right) |p - q|.$$

Figure 5 shows nodes p , q and r referred to in Lemma 1.

Let $\mathcal{P} = (u_0, u_1, \dots, u_{m-1}, u_m)$ denote the (s, d) CDC-path in \mathcal{G} where $s = u_0$ and $d = u_m$. In the following lemmas, we assume that $m \geq 2$. The case where $m = 1$ is discussed in Remark 1 above. We describe a procedure that shows that existence of an (s, d) CDC-path \mathcal{P}'' in *CDCY G_k* such that $|\mathcal{P}''| \leq t \cdot |\mathcal{P}|$. Note that this is not the algorithm used in practice to find the CDC-path but is described just to prove the spanner property of *CDCY G_k* . From now on, for every path we create we also assign channels to its links.

The algorithm to show the existence of an (s, d) CDC-path in *CDCY G_k* performs the following two steps in order:

1. **Link Replacement:** Each link connecting u_i, u_{i+1} in \mathcal{P} is replaced by a path. Note that the resulting path \mathcal{P}' , obtained by replacing all the links in \mathcal{P} , may not be simple. However, as we prove shortly in Lemma 3, $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ and we may assign channels to \mathcal{P}' satisfying CDC.
2. **Untangling:** A path \mathcal{P}'' is obtained from \mathcal{P}' such that $|\mathcal{P}''| \leq |\mathcal{P}'|$ and \mathcal{P}'' is simple. The path \mathcal{P}'' is constructed by repeatedly removing loops in \mathcal{P}' so that the CDC property is maintained when retaining the same channel assignments done during Link Replacement. This is proved in Lemma 4.

It is important to note that the notion of types is used at each step of the above procedures to maintain paths that satisfy the CDC property. We explain in more detail below.

Link Replacement—We divide the links in \mathcal{P} into three cases - **(i)** the intermediate links (u_i, u_{i+1}) for $i = 1 \dots m$, **(ii)** the first link (s, u_1) and **(iii)** the last link (u_{m-1}, d) .

Now, consider the first case where link $(u_i, u_{i+1}) \in \mathcal{P}$ is an intermediate link. We replace this link with a path $\mathcal{P}'_{u_i, u_{i+1}} = (u_i = v_0, v_1, \dots, v_{r-1}, v_r = u_{i+1})$ in the following iterative manner. Assume that the path added so far ends at v_j . If $(v_j, u_{i+1}) \in \mathcal{L}'$, add u_{i+1} . Otherwise, add the nearest and second nearest nodes of v_j of type either $T(u_i)$ or $T(u_{i+1})$ (both nearest and second nearest neighbors must be of the same type) in the sector ψ of v_j containing u_{i+1} . Repeat the process for v_{j+2} . See Figure 6 for an illustration of this process.

In the case of the first link, if s has exactly one channel available, we may only construct the above path through nodes of type $T(u_1)$. In the case of the last link, if d has exactly one channel available to it, then we may construct a path back from d to u_{m-1} through nodes of type $T(u_{m-1})$.

Let \mathcal{P}' be the path obtained by concatenating $\mathcal{P}'_{u_{i+1}, u_{i+2}}$ after $\mathcal{P}'_{u_i, u_{i+1}}$ ($i=1 \dots m-1$). We prove in Lemma 2, that \mathcal{P}' always exists. Next, in Lemma 3, we show that $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ and we can assign channels to the links in \mathcal{P}' satisfying the CDC if the channel assignment for P satisfies CDC.

Untangling—Note that \mathcal{P}' might not be simple, since when constructing $\mathcal{P}'_{u_i, u_{i+1}}$, no caution is taken in order that we do not use nodes of $\mathcal{P}'_{u_j, u_{j+1}}$ where $j < i$. Lemma 4 shows how to create an (s, d) CDC-path from \mathcal{P}' , without increasing its length. Note that in the case of spanners in the general case (when the CDC property is not required), such non-simple paths may be reduced to simple paths easily by shortcutting. The problem is more complicated in our case since we need to ensure that we can still assign channels to the new path satisfying CDC. Again, the notion of types plays the major role in generating simple paths which also satisfies CDC. Intuitively, it is sometimes sufficient to directly shortcut (while satisfying CDC). At other times, we may need to take a small diversion while shortcutting (replacing one link by two new links). We show that this does not increase the cost over the cost of \mathcal{P}' and hence, the spanner property is satisfied.

Lemma 2

If there is a path \mathcal{P} from s to d in \mathcal{G} , then the path \mathcal{P}' from s to d in $\text{CDCY } G_k$ constructed by the Link Replacement Procedure exists.

Proof—Consider a link $(u, v) \in \mathcal{P}$. We first look at the case where (u, v) is an intermediate link. Consider a step of the Link Replacement procedure where we are at a node v_j , for some even j . Now, in the sector of v_j containing v , we check if the v is among the two nearest neighboring nodes of type $T(v)$. If this is the case, we may directly connect to it. Now, since v is a neighbor of u and for each v_j , $|v_j - v| \geq |v_{j-2} - v|$ we deduce that v is a neighbor of v_j . Hence, there cannot be a case that v_j does not have any neighbors of type $T(v)$. This implies that the distance to v is monotonically decreasing and v_j always has neighbors of type $T(v)$. Therefore, there is a path from u to v in $\text{CDCY } G_k$.

In the case where (u, v) is the first link, we may be restricted to paths through nodes of type $T(v)$ since $T(v)$ has more than one channel because the number of links in \mathcal{P} is at least two under our assumption that $m \geq 2$. Hence, the above reasoning holds in this case also. In the case where (u, v) is the last link, similar to the previous case, we deduce that $T(u)$ has more

than one channel. Therefore, we may find a path from v to u consisting of nodes of type $T(u)$.

Lemma 3

$|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ for $t = (1 - 2 \sin(\theta/2))^{-2}$ and it is possible to assign channels the links in \mathcal{P}' satisfying the Channel Discontinuity Constraint.

Proof—Let $\mathcal{P}'_{u_i, u_{i+1}} = (u_i = v_0, v_1, \dots, v_{r-1}, v_r = u_{i+1})$ be the portion of \mathcal{P}' from u_i to u_{i+1} for some i . Consider the set of nodes $\{u = v_0, v_2, \dots, v_{r-3}, v_{r-1}, v_r = u_{i+1}\}$ consisting of nodes v_j , where j is even, and the node u_{i+1} . For each node v_j in this set, where $0 \leq j \leq r-3$, from Lemma 1, we know that

$$|v_{j+2} - v_r| \leq |v_j - v_r| - \frac{|v_j - v_{j+2}|}{c}, \quad (1)$$

where $c = \left(1 - 2 \sin \frac{\theta}{2}\right)^{-1}$. Summing over all such nodes v_j and rearranging, we get

$$\begin{aligned} \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - v_{j+2}| &\leq c \cdot \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} [|v_j - v_r| - |v_{j+2} - v_r|] \\ &= c \cdot [|v_0 - v_r| - |v_{r-1} - v_r|], \\ &= c \cdot [|u_i - u_{i+1}| - |v_{r-1} - u_{i+1}|]. \end{aligned} \quad (2)$$

Now, for each pair of nodes (v_j, v_{j+2}) , where j is even and $0 \leq j \leq r-3$, from Lemma 1,

$$\begin{aligned} |v_j - v_{j+2}| - \frac{1}{c} \cdot |v_j - v_{j+1}| &\geq |v_{j+1} - v_{j+2}| \\ \Rightarrow |v_j - v_{j+1}| + |v_{j+1} - v_{j+2}| &\leq c \cdot |v_j - v_{j+2}|, \end{aligned} \quad (3)$$

since $c \geq 1$. Summing over all j , we get,

$$\begin{aligned} \sum_{j=0}^{r-2} |v_j - v_{j+1}| &\leq c \cdot \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - v_{j+2}| \\ \Rightarrow \sum_{j=0}^{r-1} |v_j - v_{j+1}| &\leq |v_{r-1} - u_{i+1}| + \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - v_{j+2}| \\ &\leq c^2 \cdot |u_i - u_{i+1}|, \end{aligned}$$

by substituting for $\sum_{j=0, j \text{ is even}}^{r-3} |v_j - v_{j+2}|$ from Equation 2 and using the fact that $c \geq 1$. Summing over all links (u_i, u_{i+1}) , we have proved that $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ for a constant $t = (1 - 2 \sin(\theta/2))^{-2}$.

Let the nodes u_i and u_{i+1} share the channel C_2 and that the links (u_{i-1}, u_i) and (u_{i+1}, u_{i+2}) are assigned the channels C_1 and C_3 respectively. We now have a channel assignment for the path $\mathcal{P}'_{u_i, u_{i+1}}$ such that no two consecutive links in \mathcal{P}' have the same channel as follows:

- For every link (v_j, v_{j+1}) , where $j = \{0, 2, 4, \dots, r-3\}$, assign channel C_2 (this is possible because v_j and v_{j+1} are of type $T(u_i)$ or $T(u_{i+1})$, both of which have the channel C_2),

- For every link (v_{j+1}, v_{j+2}) for $j = \{0, 2, 4, \dots, r-3\}$, assign channel C_1 or C_3 depending on whether v_{j+1} and v_{j+2} are of type $T(u_i)$ or $T(u_{i+1})$,
- Finally, assign the channel C_2 to the link v_{r-1}, u_{i+1} .

This channel assignment is shown in Figure 6. If (u_i, u_{i+1}) is the first (resp. last) link in \mathcal{P} , it does not matter what channel is assigned to the first (resp. last) link in $\mathcal{P}'_{u_i, u_{i+1}}$. Hence, we may assign channels to $\mathcal{P}'_{u_i, u_{i+1}}$ in a similar manner using channels (C_2, C_3) (resp. C_2, C_1). Note that, in \mathcal{P}' , two non-consecutive links may be adjacent since it is not simple and hence, may share the same channel. Hence, we still need to make it simple while satisfying the CDC property.

Lemma 4

If, for some i , $\mathcal{P}'_{u_i, u_{i+1}}$ overlaps with the portion of \mathcal{P}' following u_{i+1} , then we can generate a new path \mathcal{P}'' such that

- $|\mathcal{P}''| \leq |\mathcal{P}'|$,
- \mathcal{P}'' has fewer such overlaps, and
- no two consecutive links are assigned the same channel.

Proof—Let the portion of \mathcal{P}' preceding u_i be \mathcal{P}'_{s, u_i} and that following u_{i+1} be $\mathcal{P}'_{u_{i+1}, d}$.

Now, either \mathcal{P}'_{s, u_i} or $\mathcal{P}'_{u_{i+1}, d}$ or both share nodes with $\mathcal{P}'_{u_i, u_{i+1}}$. We can obtain a new CDC-path \mathcal{P}'' as follows.

We show the CDC-path for the case where $\mathcal{P}'_{u_{i+1}, d}$ shares nodes with $\mathcal{P}'_{u_i, u_{i+1}}$. Assume that $\mathcal{P}'_{u_{i+1}, d}$ uses a node v_j in $\mathcal{P}'_{u_i, u_{i+1}}$ and this is the last node in $\mathcal{P}'_{u_{i+1}, d}$ which is in $\mathcal{P}'_{u_i, u_{i+1}}$. Let the channel assigned to the outgoing link from v_j to a node w in the path \mathcal{P}' be C_{out} as shown in Figure 7. There are three cases for C_{out} , **(i)** $C_{out} = C_1$ and $C_{out} = C_2$, **(ii)** $C_{out} = C_1$ and **(iii)** $C_{out} = C_2$. Now, we obtain a new path (not necessarily simple) \mathcal{P}'' satisfying the CDC property by replacing the portion of \mathcal{P}' from u_i to v_j with a new path \mathcal{P}'_{u_i, v_j} which we describe below.

There are two cases for the node v_j depending on whether j is odd or even. We first handle the case when j is odd.

In this case of odd j , v_j and v_{j+1} are in the sector of v_{j-1} which contains u_{i+1} . This is because, according to the link replacement procedure, from v_{j-1} , we would have picked the two nearest neighbors in the sector of v_{j-1} containing u_{i+1} . If j is odd, the two cases where $C_{out} = C_2$ and $C_{out} = C_2$ are depicted in Figure 7.

If $C_{out} = C_2$, we create \mathcal{P}''_{u_i, v_j} by following from u_i to v_j along $\mathcal{P}'_{u_i, u_{i+1}}$. The (s, d) path obtained using this \mathcal{P}''_{u_i, v_j} satisfies CDC because, according to the channel assignments for $\mathcal{P}'_{u_i, u_{i+1}}$, the link (u_i, v_1) is not assigned channel C_1 and the link (v_{j-1}, v_j) is assigned channel C_2 which is different from C_{out} . We are not increasing the cost of \mathcal{P} by doing this. This is depicted in Figure 7a.

If $C_{out} = C_2$, we obtain \mathcal{P}''_{u_i, v_j} by going from u_i to v_{j-1} along $\mathcal{P}'_{u_i, u_{i+1}}$ and adding the links (v_{j-1}, v_{j+1}) and (v_{j+1}, v_j) . Assigning channels C_2 and C_1 to links (v_{j-1}, v_{j+1}) and (v_{j+1}, v_j) respectively, we get a path \mathcal{P}'' from s to d . This is depicted in Figure 7b.

Now, we prove that the cost of \mathcal{P}'' is no larger than the cost of \mathcal{P}' . Let the portion of \mathcal{P}' from s to v_{j-1} be $\mathcal{P}'_{s, v_{j-1}}$ and the portion from v_j to d be $\mathcal{P}'_{v_j, d}$. Applying triangle inequality in the triangles (v_{j-1}, v_j, v_{j+1}) and (v_j, v_{j+1}, v_{i+1}) , we have

$$\begin{aligned} |v_{j-1} - v_{j+1}| &\leq |v_{j-1} - v_j| + |v_j - v_{j+1}| \quad \text{and} \\ |v_j - v_{j+1}| &\leq |v_{j+1} - u_{i+1}| + |u_{i+1} - v_j|. \end{aligned}$$

Adding, we get,

$$|v_{j-1} - v_{j+1}| + |v_j - v_{j+1}| \leq (|v_{j-1} - v_j| + |v_j - v_{j+1}|) + (|v_{j+1} - u_{i+1}| + |u_{i+1} - v_j|)$$

Adding the cost of the portions of \mathcal{P}' from s to v_{j-1} and from v_j to d , we get,

$$\begin{aligned} |\mathcal{P}''| &\leq |\mathcal{P}'_{s, v_{j-1}}| + (|v_{j-1} - v_j| + |v_j - v_{j+1}|) \\ &\quad + (|v_{j+1} - u_{i+1}| + |u_{i+1} - v_j|) + d_{\mathcal{P}'_{v_j, d}} \end{aligned}$$

Since the cost of the portions of \mathcal{P}' from v_{j+1} to u_{i+1} and from u_{i+1} to v_j are not less than $|v_{j+1} - u_{i+1}|$ and $|u_{i+1} - v_j|$ respectively, $|\mathcal{P}''| \leq |\mathcal{P}'|$. The case where \mathcal{P}'_{s, u_i} overlaps with $\mathcal{P}'_{u_i, u_{i+1}}$ is analogous to the above case. Instead of using the last node of overlap, we would use the first node of overlap to obtain the shortened path from s to d . The number of overlaps in \mathcal{P}'' is less than that in \mathcal{P}' because in \mathcal{P}'' , the portion of the path from u_i to v_j now no longer overlaps with the portion from v_j to d .

Proof of Theorem 1—Consider a CDC-path \mathcal{P} between nodes s and d . The case where the number of links in \mathcal{P} is one has been discussed in Remark 1.

If the number of links in \mathcal{P} is two or more, we may obtain the path \mathcal{P}' in $CDCY G_k$ as described in the link replacement procedure, which may not be simple. From \mathcal{P}' , we may apply Lemma 4 to $\mathcal{P}'_{u_i, u_{i+1}}$ to remove any overlaps with $\mathcal{P}'_{u_{i+1}, d}$ followed by removal of overlaps with \mathcal{P}'_{s, u_i} (for $i = 1 \dots m-1$), to obtain \mathcal{P}'' a simple CDC-path in $CDCY G_k$ from s to d . Note that \mathcal{P}'_{s, u_i} and $\mathcal{P}'_{s, u_{i+1}}$ cannot overlap at the same node on $\mathcal{P}'_{u_i, u_{i+1}}$ as the original CDC-path is simple. Since we are eliminating all overlaps of $\mathcal{P}'_{u_j, u_{j+1}}$ for $j < i$ before $\mathcal{P}'_{u_i, u_{i+1}}$, this case will not occur because such a shared node will no longer be shared after the first time it is found to overlap. Hence, at each step, we reduce the number of overlaps and at the final step when we are handling $\mathcal{P}'_{u_{m-1}, u_m}$, there will be no overlaps.

By Lemmas 3 and 4, we know that $|\mathcal{P}''| \leq |\mathcal{P}'| \leq t|\mathcal{P}|$. Hence, $CDCY G_k$ is a CDC t -Spanner for $t = (1 - 2 \sin(\theta/2))^{-2}$. Now, for each node v in $CDCY G_k$, we have k sectors.

In each sector, there are links from v to two nodes of every type. Hence, there are $O(k|\mathcal{T}|) = O(|\mathcal{T}|/\theta)$ links for each node v . In total, there are $O(|\mathcal{T}|(n/\theta))$ links.

Remark 2

When considering the common Rayleigh Fading Model where the cost/power of transmitting across a link between nodes u, v is proportional to $|u - v|^\beta$ where β typically lies between 2 and 4, it can be easily seen that CDCY G_k is a t -spanner with $t = ((1 - 2 \sin(\theta/2))^{-2})^\beta$. If $/P = (v_1 = s, v_2, \dots, v_k = d)$ is the CDC-path in \mathcal{G}' , this follows from the following equation:

$$\sum_{i=1}^{k-1} (|v_i - v_{i+1}|^\beta) \leq \left[\sum_{i=1}^{k-1} |v_i - v_{i+1}| \right]^\beta$$

The total number of links in CDCY G_k seems to be $O(k2^c n) = O(2^c(n/\theta))$ since the total number of types is $O(2^c)$. We now argue that the number of links can actually be bounded by $O(c^2(n/\theta))$ by constructing the graph in the following manner. At a node v , for each pair $\{C_a, C_b\}$ of channels, connect to the two nearest neighbors which have these channels. Let this graph be CDCY G'_k . The following lemma shows that CDCY G'_k is a t -spanner for CDC routing.

Lemma 5

CDCY G'_k is a CDC t -Spanner where $t = (1 - 2 \sin(\theta/2))^{-2}$ and the number of links in CDCY G'_k is $O(c^2(n/\theta))$.

Proof—Consider an edge (u, v) which is part of some CDC-path in \mathcal{P} in \mathcal{G} . Note that we are still under the assumption that the number of nodes in \mathcal{P} is greater than 2. Let us consider the three cases: (i) (u, v) is one of the intermediate links, (ii) (u, v) is the first link in \mathcal{P} and (iii) (u, v) is the last link in \mathcal{P} . Consider case (i). If the only channel shared between u and v is C_a , then v has at least one more channel, say C_b , available at it. Now, we may construct a path $\mathcal{P}_{u,v}$ through nodes which have channels $\{C_a, C_b\}$ satisfying CDC since we may alternately assign C_a and C_b to links in $\mathcal{P}_{u,v}$. In cases (ii) and (iii), for (u, v) , we may still construct paths from u to v because, again, at least one of them has two or more channels. Lemmas 2, 3 and 4 still hold for these paths. Hence, CDCY G'_k is a CDC t -Spanner for \mathcal{G} . The number of links in CDCY G'_k at a node v is $O(c^2)$ per sector since we have three links for each pair of channels $\{C_a, C_b\}$ and the total number of pairs is $O(c^2)$. Hence, the total number of links in CDCY G'_k is $O(c^2(n/\theta)) = O(n/\theta)$, since $|\mathcal{T}|$ is independent of n .

Constructing the spanner in a distributed manner: We now outline the algorithm to compute the spanner in a distributed manner using $O(n \log n)$ expected number of messages.

Let p be a node, and let $\{\Psi_1 \dots \Psi_{\lceil 2\pi/\theta \rceil}\}$ be the set of sectors around p . For each type T , p needs to know the first two nearest neighbors of type T in each sector ψ_i . In order to find these neighbors, p sends a message, specifying the sector and type. Each node q that hears the message, estimates the distance to p , and picks a **random backoff time** δ_q during which it waits and listen to see how many nodes w it hears whose distance $|w - p| < |p - q|$. If when the timer of q expires, it did not hear at least one other node w , then q broadcasts its distance $|q - p|$. This process is repeated to find the second nearest neighbor (the nearest neighbor is indicated in the message sent by p and is silent this time). In this scenario, for the purposes of analysis, we assume an ideal network where there are no collisions or transmission errors

as in [39]. Note that the effects of collisions decrease with increasing node density since enough nodes might be able to hear the message inspite of collisions. On the other hand, the probability of collisions is lower when the node density is low.

Theorem 2

The construction of the spanner using the above procedure can be performed using expected $O((n/\theta) \log n)$ number of messages.

Proof—Our proof follows the ideas of Arango et.al. [39]. The first step is to show that for each node p , the expected number of messages transmitted is $O((1/\theta) \log d)$ where d is the degree (number of neighbors) of p .

Consider a sector ψ of p . Any two nodes q_i and q_j in ψ are within transmission range of each other, because the the maximum distance between any two points in a sector of angle less than $\pi/3$ is the radius. Consider the nodes $\{q_1, q_2, \dots, q_{deg(\psi)}\}$ in ψ where $deg(\psi)$ is the number of nodes in ψ . Without loss of generality, let q_i be the i^{th} node to transmit (has the i^{th} smallest backoff time). Let $\{q_1, q_2, \dots, q_i\}$ be increasingly ordered based on the distance to p . The probability that q_i appears left-most (has least distance to p among $\{q_1, q_2, \dots, q_{i-1}\}$) is exactly $1/i$, since in an ordering of $\{q_1, q_2, \dots, q_{i-1}\}$ based on their distance to p , there are i positions to place q_i . Hence, the expected number of messages transmitted over all q_i is

given by $\sum_{i=1}^{deg(\Psi)} (1/i) = 2 \log deg(\Psi)$ (since the process is repeated twice to find the second nearest neighbor). Summing over all sectors, the expected number of messages transmitted overall is $O(\lceil 4\pi/\theta \rceil \log d) = O(\log d/\theta)$.

Hence, if d_i is the degree of s_i the total number of messages transmitted over all nodes is given by

$$\sum_{i=1}^n O(\log d_i/\theta) = \sum_{i=1}^n O(\log n/\theta) = O((n/\theta) \log n)$$

Remark 3

The small number of messages is particularly useful in a dynamic setting in which nodes, are added or deleted frequently and may be mobile, and the spanner needs to be updated after each event.

Our results might be of a theoretical nature, since the constants in the bounds might be too large. However, we use them as evidence that, in practice, connecting every node to a small number of its nearest neighbors, in each of the sectors (lobes), either user-defined or explicitly when directional antennas are used, will yield a network with almost same properties as the original network.

4. Finding the Shortest CDC Path

We now switch to the second main result of this paper, showing how to find the minimum-cost CDC-path between two nodes in an arbitrary wireless network. Again, we are given the connectivity graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ with n nodes and m links. Let $w(u, v)$ be the cost of link (u, v) and let $\mathcal{C}(u)$ be the set of channels available at node u . Let the channels available in the network be $C = \{1, 2, \dots, c\}$. We assign channels $\mathcal{C}(u, v) = \mathcal{C}(u) \cap \mathcal{C}(v)$ to the link (u, v) since communication may take place only on these channels. It is important to note here that

the cost on the links $w(u, v)$ need not be dependent on the distance between the nodes u and v for the purposes of this algorithm.

Ahuja et al. [6] showed the equivalence of computing the minimum cost CDC-path to the computation of minimum cost perfect matching (MCPM) using Edmonds-Szeider (ES) expansion [41] for nodes. Since we are going to use the same expansion, we give a description of their method here. Given the connectivity graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ with a source destination pair s, d , we define the ES-expansion graph $\mathcal{G}'(\mathcal{N}', \mathcal{L}')$ as follows:

1. We expand each node u , except s and d nodes, into $2c + 2$ sub-nodes, denoted $(u_1, u'_1), (u_2, u'_2), \dots, (u_c, u'_c)$, each pair corresponding to a channel in \mathcal{C} . Finally, we add two sub-nodes u_g, u'_g . Thus,

$$\mathcal{N}' = \left[\bigcup_{u \in \mathcal{N}, u \notin \{s, d\}} \{u_1, u'_1, u_2, u'_2, \dots, u_c, u'_c, u_g, u'_g\} \right] \cup \{s, d\}.$$

2. For each node $u \in \mathcal{N}, u \notin \{s, d\}$, we add edges $\{(u_i, u'_i), (u'_i, u_g), (u'_i, u'_g)\}, \forall i = \{1, 2, \dots, c\}$ to \mathcal{L}' . Assign cost 0 to all the above links. The links $(u_i, u'_i), \forall i = \{1, 2, \dots, c\}$ are referred to as **channel links**.
3. For each link $(u, v) \in \mathcal{L}, u, v \notin \{s, d\}$ with channels $\mathcal{C}(u, v)$, we add the links u_i, v_i for all $i \in \mathcal{C}(u, v)$ to \mathcal{L}' . Each of these links assigned a cost $w(u, v)$.
4. For each link (s, u) for some u (respectively, (v, d) for some v), add links $(s, u_i), \forall i \in \mathcal{C}(s, u)$ (respectively, $(v_i, d), \forall i \in \mathcal{C}(v, d)$). The costs of these links are $w(s, u)$ (respectively, $w(v, d)$).

Let $n' = |\mathcal{N}'|$ and $m' = |\mathcal{L}'|$. Clearly, $n' = O(cn)$ and $m' = O(cm + n)$.

Ahuja et al. [6] also noted that these expansions may be modified to reduce the number of links. For links with less than three channels available on them, we perform the expansion. However, for links with three or more channels, only three channel links are necessary since, for any CDC-path which uses this link, we can always find a channel assignment from the three channels available.

Now, the problem of computing the shortest CDC-path is the same as computing the minimum-cost perfect matching in the expanded graph \mathcal{G}' (see [6, 41] for more details). A perfect matching in a graph is a set of non-adjacent edges, i.e., no two edges share the same vertex, such that all the vertices in the graph are covered. A minimum cost perfect matching is defined as a perfect matching where the sum of the cost of the edges in the matching is minimum. An interesting point is that although the network is connected, there may be no CDC-path between some two nodes s and d since there may be no perfect matching in the corresponding expanded network.

We illustrate the equivalence of MCPM and CDC-path using the example from [6]. Consider the example network and its ES-expansion as shown in Figure 8. Numbers over links represent the available channels on the corresponding links. The link costs are assumed to be 1 for this illustration, hence the objective is to compute the minimum hop CDC-path. The edges in the matching are shown in bold in the expanded graph. The nodes not present in the shortest path from s to d would find matching within itself (see nodes y and z).

Intermediate nodes (if any) in the shortest s to d path will have exactly two incident external edges in the matching (see node x). Clearly, the returned path $s-x-d$ with channels 1 and 3 over links $s-x$ and $x-d$, respectively, is the minimum-hop CDC-path between s and d .

The best known sequential implementation of Edmonds' minimum cost perfect matching algorithm is by Gabow [42] with time complexity $O(n'm')$. Since the number of nodes in the expanded graph is $O(cn)$ and the number of links is $O(cm + n)$, the complexity of computing a solution to MCPM is $O(nmc^3)$. However, this complexity can be reduced by observing that most of the matching computed in the MCPM are within the nodes itself (except for the links involved in the path). A closer look reveals that we can start with an expanded graph with a partial matching where, at each expanded node $u \notin \{s, d\}$, all the channel links (u_i, u'_i) and (u_g, u'_g) are matched. If this is the case, then the problem of computing the minimum cost perfect matching is transformed to computing the minimum cost alternating path. We thank Kasturi Varadarajan for this observation. It is interesting to note that Varadarajan and Agarwal have designed algorithms for exactly [43] and approximately [44] finding a minimum-cost matching in a geometric setting. Note that using Gabow's scheme for perfect matching, the time taken to find a single alternating path in a centralized manner is $O(m')$. Hence, using the t -spanner from Section 3 containing $L = O(m)$ links, this may be reduced to $O(cn)$ time. However, the constraints of a distributed system require more careful consideration. Also, as mentioned earlier, our algorithm will run in $O(cm)$ time in a centralized implementation even with no assumptions on the costs of the links (they do not need to be distances between nodes or any function of the distance).

We now move on to the description of our distributed algorithm to find minimum-cost CDC-paths. We are given the expanded graph $\mathcal{G}'(\mathcal{N}', \mathcal{L}')$ together with the matching \mathcal{M} of zero cost (all vertices except s and d internally matched). For any node $u \in \mathcal{N}'$ let $MATE(u)$ denote the mate of u in the matching \mathcal{M} . In this section, we follow the definitions in Edmonds' algorithm for minimum-cost perfect matching [45, 46, 47, 48]). For combinatoric interpretations of Edmonds' algorithm, see [47] and [49]. Gabow [47] introduces the notion of **co-link paths** used in this paper. Our method is similar to these combinatoric approaches and takes advantage of the constraints available in our setting. An important point to note here is that for our algorithm, we do not assume that the costs of links is a function of the distance between the nodes. The costs assigned to the links could be arbitrary.

Definition 4

Between any two nodes s and d , we define an **alternating path** to be a path with alternating unmatched and matched edges. An **augmenting path** is an alternating path between two unmatched vertices.

Definition 5

For any node u , we define a **co-link path** from s to u as an alternating path $\pi = \{u_0 = s, u_1, u_2, \dots, u_k = u\}$ where $(s, u_1), (u_{k-1}, u) \notin \mathcal{M}$.

Our algorithm is an extension with Edmonds' algorithm. Edmonds' algorithm works in multiple phases by repeatedly finding augmenting paths between unmatched vertices and terminates when there are no more such augmenting paths. Our algorithm works by finding minimum cost co-link paths similar to each phase of Edmonds' algorithm. Each node u maintains two values: $d_T[u]$ and $d_S[u]$, corresponding to the minimum cost co-link paths to itself and $MATE(u)$ respectively. Through the course of the algorithm, each node is given labels from the set $\{S, T, F\}$. We say a node $u \in S$ if it is given an S label and similarly for

the other labels. Initially, $s \in S$ and every other node $u \in F$. Each node u also maintains the current known values d_S and d_T of its neighbors in S . During the algorithm, certain odd subsets of vertices are termed as *blossoms*.

Definition 6

A **blossom** B is an odd circuit in G for which $M \cup B$ is a perfect matching for all vertices in B except one. The lone unmatched vertex is termed as the **base** of the blossom.

Through the course of the algorithm, a node u may be added to one or more blossoms. Hence, it maintains the ID of the base of the outermost blossom it belongs to (the reader is referred to the book by Lawler [48] for the definition of outermost blossom). Let $b[u]$ denote this ID. When a node u is added to S for the first time, it stores $P_T[u]$, the corresponding ID of the parent of u . When a node u is added to T for the first time, if it is not the base of a blossom, it stores the corresponding ID of the parent of u , $P_T[u]$. If it is the base of a blossom B , then B becomes the sub-blossom of a new blossom B' . In this case u stores the edge (x, y) which is the first unmatched edge in B' connected to B and is not in B . For more details, see the book by Lawler [48].

At each stage of the algorithm, we have the sets S , T and F . Each node $u \in S$ finds the following values for each of its neighbors v :

$$val[v] = \begin{cases} \frac{d_S[u] + d_S[v] + w(u, v)}{2} & \text{if } v \in S, \\ d_S[u] + w(u, v) & \text{if } v \in F \end{cases}$$

Each node u maintains an “examined” state for each neighbor $v \in S, F$. Let

$$\begin{aligned} minval[u] &= \min_{v \text{ not examined}} val[v], \text{ and} \\ v_{min} &= \arg \min_{v \text{ not examined}} val[v] \end{aligned}$$

The algorithm works in $O(n)$ iterations. Initially, no co-link paths to any node are known. Hence, $d_S[u] = d_T[u] = \infty$ for all $u \neq s$. For s , we may consider the co-link path to be a path of 0 edges, hence, $d_S[s] = 0$. Also, initially, $s \in S$ and all other nodes are in F . In each iteration, it executes two of three phases: find_min, blossom and grow. First, it executes find_min which finds the following value:

$$minval = \min_{u \in S} minval[u].$$

Let (u, v) be the corresponding edge on which this was attained. Now, if $v \in F$, we have found the shortest alternating paths to v and $MATE(v)$ from s . Thus, grow is executed where appropriate labels and distances are assigned to v and $MATE(v)$. If $v \in S$, then v has already been visited by the algorithm and hence, we have closed a blossom at the edge (u, v) . Thus, blossom is executed which assigns appropriate distances and blossom ids to the nodes in the new blossom.

Steps of find min:

1. Each node u computes $minval[u]$ and the corresponding v_{min} .
2. If $v_{min} \in S$, it sends a message to v_{min} requesting $b[v_{min}]$. If $b[v_{min}] = b[u]$, the two nodes belong to the same blossom and no blossom discovery is necessary. In this

case, u marks v as examined. The process repeats from Step 1 to discover a new v_{min} . This continues until the minimum is achieved for a node $v \in F$ or a node $v \in S$ and $b[v] = b[u]$.

3. Each node u sends a message $\langle v, w, minval[u] \rangle$ to s in the following manner. Each node u waits until it has received messages from all its children $v \in S \cup T$. Then, u finds the minimum $minval[v]$, $v \in N(u)$ where $N(u) = \{v: v \in S \cup T \text{ and } (u, v) \in \mathcal{L}'\}$. u sends the message $\langle w, x, minval[u] \rangle$ to its parent where (w, x) is the edge for which the minimum was attained. For nodes u in blossoms, the parent depends on which set it was added to first. When s receives all messages, it sends a message to nodes w and x where edge (w, x) is the one for which the minimum was attained. w and x mark each other as examined.

Steps of blossom:

1. Let (u, v) be the edge which closes the new outermost blossom Q . u and v send messages towards s till the base of the blossom b is discovered. The way in which these messages are sent follows the description given by Shieber and Moran[50]. Essentially, u and v send messages back along the paths through $P_S[u]$ and $P_S[v]$ to s . Along this path, all nodes are marked with whether they belong to both the paths from s to u and s to v . Finally, s sends messages along the opposite direction along these paths to discover the base of the blossom.
2. b stores the edge (u, v) as the edge which caused the blossom to be discovered. The bases of all old blossoms which now became sub-blossoms of b send messages by flooding among themselves to inform the nodes of their new membership in b . Finally, b sends messages to u and v informing any nodes which newly became part of the blossom of their membership.
3. Finally, for all new nodes in the blossom, the corresponding parents and the corresponding alternate distance d_s or d_t are assigned.

Steps of grow:

1. Let (u, v) be the edge which is used to discover v newly. v is added to T and $MATE(v)$ is added to S . v and $MATE(v)$ are removed from F .
2. The following values are assigned:

$$\begin{aligned} d_t[v] &= d_s[u] + w(u, v) \\ d_s[MATE(v)] &= d_t[v] \\ P_S[MATE(v)] &= v \\ P_T[v] &= u \end{aligned}$$

3. $MATE(v)$ sends a message $\langle v, d_S[MATE(v)] \rangle$ to all its neighbors $w \in S$ informing them of its new S label and its distance d_s .

The overall steps of the algorithm are outlined in Figure 9.

We now prove the correctness of the algorithm by showing that, at each stage of **SHORTEST-CDC**, the same edge is selected as in each phase of Edmonds' algorithm. From now on, whenever we mention Edmonds' algorithm, we refer to each phase of Edmonds' algorithm.

Fact 1—For each edge $(u, v) \in \mathcal{M}$, $w(u, v) = 0$

We now give an outline of Edmonds' algorithm for the sake of completeness. For more details, see [48]. Each node u is associated with a dual variable $y(u)$ and each odd subset of vertices Q is assigned a dual variable $z(Q)$. Note that $z(Q) > 0$ only for blossoms and $z(Q) = 0$ for every other odd subset. Each node and blossom belong to one of three sets $\{S, T, F\}$ corresponding to three cases: (i) S indicates that an alternating path from s ending in a matched edge has been found, (ii) T indicates that an alternating path from s ending in an unmatched edge is found and (iii) F indicates that no alternating path has been found. Initially, $s \in S$ and everything else is in F . An edge is termed as tight if $y(u) + y(v) + \sum_{u,v \in Q} z(Q) = 0$. A tree rooted at s is maintained through the course of the algorithm. At each step, Edmonds' algorithm searches for tight edges in order to shrink a blossom or grow the tree from s . If there are no tight edges, it makes a dual change. We choose $\delta = \min(\delta_1, \delta_2, \delta_3)$, where

$$\delta_1 = \min_{\text{non-trivial blossom } Q \in T} \frac{-z(Q)}{2} \quad (4)$$

$$\delta_2 = \min_{u,v \in S} \frac{w(u,v) - (y(u) + y(v))}{2} \quad (5)$$

$$\delta_3 = \min_{u \in S, v \in F} w(u,v) - (y(u) + y(v)) \quad (6)$$

For each node $u \in S$, we set $y(u) = y(u) + \delta$ and for each node $u \in T$, $y(u) = y(u) - \delta$. For each outermost blossom $Q \in S$ (resp. $Q \in T$), $z(Q) = z(Q) - 2\delta$ (resp. $z(Q) = z(Q) + 2\delta$). Note that if a new blossom Q is constructed during the algorithm, Q is always labeled S . The only case when a non-trivial blossom Q is labeled T is when it is already shrunk at the start of the algorithm, i.e., it has been shrunk by a previous search for an alternating path. In our case, there are no shrunk blossoms at the beginning. Hence, we only need to worry about δ_2 and δ_3 .

Lemma 6

For each edge $(u, v) \in \mathcal{M}$,

$$y(u) + y(v) + \sum_{u,v \in Q} z(Q) = 0$$

throughout the course of Edmonds' algorithm.

Proof—Since each edge $(u, v) \in \mathcal{M}$ is tight throughout the course of the algorithm,

$$y(u) + y(v) + \sum_{u,v \in Q} z(Q) = w(u, v)$$

This, combined with Fact 1 gives the desired result.

Note that $d_T[u]$ is the cost of the shortest co-link path to u . This leads to the following lemma.

Lemma 7

For each node $u \in T$,

$$d_T[u] = y(s) + y(u) + \sum_{u \in Q} z(Q)$$

Proof—The proof of this lemma follows from the description given by Gabow [47]. Let the shortest alternating path from s to u ending in an unmatched edge be $\pi = \{u_0 = s; u_1; u_2; \dots; u_k = u\}$. The cost of this path $W(s; u)$ is given by,

$$\begin{aligned} W(s, u) &= \sum_{i=0}^{k-1} w(u_i, u_{i+1}) \\ &= (w(s, u_1) + w(u_2, u_3) + \dots + w(u_{k-1}, u)) + \\ &\quad (w(u_1, u_2) + w(u_3, u_4) + \dots + w(u_{k-2}, u_{k-1})) \quad (7) \\ &= (w(s, u_1) + w(u_2, u_3) + \dots + w(u_{k-1}, u)) - \\ &\quad (w(u_1, u_2) + w(u_3, u_4) + \dots + w(u_{k-2}, u_{k-1})) \end{aligned}$$

by Fact 1, since the costs of the matched edges is zero.

Since each edge (u_i, u_{i+1}) is tight, we have

$$y(u_i) + y(u_{i+1}) + \sum_{u_i, u_{i+1} \in Q} z(Q) = w(u_i, u_{i+1}).$$

Substituting this into Equation 7, we get the result of the lemma, i.e.,

$$d_T[u] = y(s) + y(u) + \sum_{u \in Q} z(Q).$$

The above holds true since s is not part of any blossoms (it can only be the base of some blossom but not an interior vertex).

Using the above lemma, we can prove that the execution of our algorithm is exactly the same as Edmonds' algorithm for finding an augmenting path between s and d . This implies the correctness of the algorithm based on the correctness of Edmonds' algorithm.

Lemma 8

If edge (u, v) becomes tight at some point in Edmonds' algorithm, it becomes tight at the same point in the algorithm **SHORTEST-CDC**.

Proof—At some intermediate step of Edmonds' algorithm, let some value δ be chosen as the dual change for that step. Let $y(u)$ be the dual variables of any node u before the dual change of δ . Let $u' = MATE(u)$. Now, according to Lemma 7,

$$y(s) + y(u') + \sum_{u' \in Q} z(Q) = d_T[u'] \quad (8)$$

A t -blossom is formed during Edmonds' algorithm only when we encounter a blossom which is already shrunk at the start, i.e., shrunk by a previous search for an alternating path.

Since there are no non-trivial (shrunk) blossoms at the start of the algorithm, and since all new blossoms are s -blossoms, by induction, no t -blossoms are formed during the algorithm and it is not necessary to consider δ_1 , given by Eqn. 4. Therefore, $\delta = \min(\delta_2, \delta_3)$, where δ_2 and δ_3 are given by Equations 5 and 6. We may compute $\delta = y(s) + \min(\delta_2, \delta_2) = \min(y(s) + \delta_2, y(s) + \delta_2)$.

Taking the case of δ_2 , we have,

$$\begin{aligned}\delta_2 &= \min_{u,v \in S} y(s) + \frac{w(u,v) - (y(u) + y(v))}{2} \\ &= \min_{u,v \in S} \frac{w(u,v) + (y(s) - y(u)) + (y(s) - y(v))}{2} \\ &= \min_{u,v \in S} \frac{w(u,v) + d_T[u'] + d_T[v']}{2}\end{aligned}\quad (9)$$

The above equations follow from Equation 8 and Lemma 6. In the case of δ_2 , we may similarly compute it as

$$\begin{aligned}\delta_3 &= \min_{u \in S, v \in F} w(u, v) - (y(u) + y(v)) \\ &= \min_{u \in S, v \in F} w(u, v) - y(u) \\ &= \min_{u \in S, v \in F} w(u, v) + y(s) - y(u) \\ &= \min_{u \in S, v \in F} d_T[u] + w(u, v)\end{aligned}\quad (10)$$

Since $d_T[u'] = d_S[u]$, Equations 9 and 10 are exactly the values computed by the algorithm **SHORTEST-CDC** during the find min phase. Hence, the lemma is proved.

Theorem 3

The algorithm **SHORTEST-CDC** finds the shortest CDC path from s to t using $O(n'^2)$ fixed-size messages.

Proof—From Lemma 8, it is clear that the steps of Algorithm **SHORTEST-CDC** follow the steps of Edmonds' algorithm exactly. Hence, we indeed find the shortest augmenting path from s to t . This implies that we have found the shortest CDC path from s to t .

We now analyze the communication complexity of Algorithm **SHORTEST-CDC**. There are three main phases to the algorithm:

- The find min phase. Here each node scans its adjacent edges at most once through the algorithm. Hence, each edge is scanned at most twice leading to a message complexity of $O(n'^2)$.
- The grow phase. Clearly, at each grow phase, $O(1)$ messages are sent leading to a total message complexity of $O(n')$.
- The blossom phase. Using the approach of [50], for each edge, we can achieve the backward and forward processes using $O(n')$ messages:
 - During the step of finding the base of the blossom, no more than $O(n')$ messages are sent since they are only sent along two paths to the source s .
 - In order for the new blossom with base b to inform its members of their membership and new values of d_S and d_T , we need to flood all newly formed sub-blossoms. These sub-blossoms are all disjoint and hence, the flooding takes only $O(n')$ overall. Finally to add any new nodes to the

blossom, we send messages along the path from b to the blossom edge which is again $O(n')$.

There can be no more than $O(n')$ blossoms found during the course of the algorithm leading to a complexity of $O(n'^2)$. Note that, although the approach of [50] is for unweighted graphs, since we do not have blossom expansions, their method can be used in exactly the same manner since the rest of the blossom discovery is identical to unweighted graphs [48]. Furthermore, although the distributed computing model of [50] does not take into account other factors such as interference, it applies equally to this scenario since, as mentioned in Section 2, we assume that communications required during the algorithm need not take interference into account since, typically, a control channel is used for this purpose and these messages are very short.

Each message sent is clearly of fixed size. Hence, the total number of fixed-size messages sent during the algorithm is $O(n'^2)$. Since $n' = O(cn)$ where n is the number of nodes in the network and c is the number of channels available which is typically small compared with n , we have the message complexity of **SHORTEST-CDC** to be $O(c^2n^2)$.

5. Simulations

We have conducted experiments to analyze the performance of **SHORTEST-CDC** under several settings. We consider network topologies ranging of 100 to 800 nodes distributed randomly in a region of 100×100 unit². We analyze two cases: (i) The density of the network is constant, i.e., the transmission range is inversely proportional to the number of nodes in the network and (ii) The transmission range is constant. In the first case, the transmission range varies from 11 units for 100 nodes to 4 units for 800 nodes and in the second case, the transmission range of the nodes is set at 5 units. The total number of channels is 10 and each node picks 4 channels at random out of these. The parameters of the system under consideration are (i) The number of nodes and (ii) The number of channels available at a node. We evaluate the following performance metrics:

Communication Complexity

We evaluate the dependence of the communication complexity (number of messages sent) of the algorithm on number of nodes and channels. The worst-case complexity is $O(n'^2)$ in the expanded graph which is a complexity of $O(c^2n^2)$ in the original network. We are interested in seeing which parameter has more effect on the communication complexity and how many messages are sent in practice.

Number of Blossoms

Since the major portion of the complexity is due to the formation and detection of blossoms in the graph, we evaluate the number of blossoms which are formed on average and correlate this with the communication complexity.

Figure 10 shows the ratio of the average number of messages sent to n^2 as a function of the number of nodes in the network. In case of constant density, we can see that as the number of nodes increases, the ratio decreases very gradually and hence, we may infer that the dependence on the number of nodes is not prominent. In the case of variable density (constant transmission range), we can see that there is a steady increase in the ratio which seems to be quadratic in n . This is expected since, the denser the network gets, the more the number of blossoms and our intuition is that the blossoms are the major factor in the complexity. In both cases, we can see that the number of messages does not stray far beyond n'^2 and is well below c^2n^2 .

Figure 11 shows the ratio of the average number of blossoms to the number of nodes n as a function of n for the experiments run in Figure 10. As can be seen, it closely correlates with the graph of Figure 10. Similar to Figure 10, in the variable density case, there is a steady increase in the ratio with number of nodes n . This is consistent with our intuition that the number of blossoms is a major factor in determining the number of messages passed during the course of the algorithm.

6. Conclusion

In this paper, we studied the problem of path selection and channel assignment with channel discontinuity constraint (CDC) in a wireless infrastructure network. We developed a t -spanner for CDC routing, where the number of links employed is significantly reduced yet guaranteeing that the minimum-cost CDC path in the spanner is no worse than t times the minimum-cost CDC-path in the original network. We showed the following relationship between the stretch factor t and the number of links: for a graph with $3N/\theta$ links, the stretch factor is given by $t = (1 - 2 \sin(\theta/2))^{-2}$. In addition, we showed that the problem of computing a CDC path is equivalent to computing the minimum cost augmenting path in an expanded graph. We developed a distributed algorithm to compute the minimum cost CDC path with $O(n^2)$ fixed-size messages. Through experimental evaluations on random network topologies, we study the trends on the number of messages and blossoms exchanged per path computation as the network size increases. The most message-intensive process in this algorithm is the blossom-forming process as it is the most difficult to do distributedly. Thus, we also study the variation in the number of blossoms with the network size and conclude that this is indeed in line with our conclusions.

Biography

Alon Efrat is an associate professor in the Department of Computer Science at the University of Arizona. He has earned his PhD from Tel-Aviv University under the supervision of Prof. Micha Sharir. He was also a postdoctorate research assistant at Stanford University, and at IBM Almaden Research Center. His research areas include geometric algorithms and their applications to sensor networks, robotics and computer vision. He is the author or co-author of nearly 95 publications, almost all in peer-reviewed, prestigious venues. He won the NSF CAREER award in 2004. He has served on many NSF panels and technical program committees in different areas, on the editorial board of the International Journal of Computational Geometry and its Application (IJCGA), was a guest editor of this journal, and served on the TPC of many prestigious conferences including ACM GIS, FOCS, SoCG, INFOCOM and others.



Dr. Pankaj K. Agarwal earned his PhD in Computer Science from the Courant Institute of Mathematical Sciences at New York University. He joined the Department of Computer Science of Duke University in 1989 where he is now the Chair and Professor of Computer Science and Professor of Mathematics. His research interests include geometric algorithms and data structures, computational molecular biology, spatial databases, global change, geographic information systems, sensor networks, and robotics. He has authored four books, and more than 250 scholarly articles in various journals, edited volumes, and international conferences. He has received many awards, including National Young Investigator, Sloan Fellow, and ACM Fellow, and he serves on the editorial boards of a number of journals.



Srinivasan Ramasubramanian received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science (BITS), Pilani, India, in 1997, and the Ph.D. degree in Computer Engineering from Iowa State University, Ames, in 2002. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at the University of Arizona, where he held the position of Assistant Professor from August 2002 to July 2008. He is a co-developer of the Hierarchical Modeling and Analysis Package (HIMAP), a reliability modeling and analysis tool, which is currently being used at Boeing, Honeywell, and several other companies and universities. His research interests include architectures and algorithms for optical and wireless networks, multipath routing, fault tolerance, system modeling, and performance analysis. He has served as the TPC Co-Chair of BROADNETS 2005 and ICCCN 2008 (Optical Networking Symposium) conferences and is an editor of the Springer Wireless Networks Journal.



Swaminathan Sankararaman received the B.E. degree in Computer Science and Engineering from Anna University, Chennai, India, in 2006. He received his M.S. degree in Computer Science in 2008 from the University of Arizona, Tucson and is currently a Ph.D. Candidate at the Department of Computer Science, University of Arizona. His research interests lie in the applications of geometric algorithms to wireless/sensor networks.



References

- [1]. Sankararaman, S.; Efrat, A.; Ramasubramanian, S.; Agarwal, PK. On channel-discontinuity-constraint routing in wireless networks. Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM), Mini-Conference;
- [2]. Akyildiz IF, Wang X, Wang W. Wireless mesh networks: a survey. *Computer Networks*. 2005; 47:445–487.
- [3]. Bruno R, Conti M, Gregori E. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*. 2005; 43:123–131.
- [4]. Mastronarde, N.; Andreopoulos, Y.; van der Schaar, M.; Krishnaswamy, D.; Vicente, J. Cross-layer video streaming over 802.11e-enabled wireless mesh networks. Proceedings of the IEEE International Acoustics, Speech and Signal Processing Conference (ICASSP);
- [5]. Kennedy, GA.; Foster, PJ. High resilience networks and microwave propagation in underground mines. Proceedings of the 9th European Conference on Wireless Technology (ECWT);
- [6]. Ahuja, S.; Gopalan, A.; Ramasubramanian, S. Joint routing and channel assignment in multi-channel wireless infrastructure networks. Proceedings of the 5th IEEE International Conference on Broadband Communications, Networks and Systems (BROADNETS);
- [7]. Bodlaender, HL.; Kloks, T.; Tan, RB.; Leeuwen, J. v. lambda-coloring of graphs. Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS); Springer-Verlag; 2000.
- [8]. Barrett, CL.; Istrate, G.; Kumar, VSA.; Marathe, MV.; Thite, S.; Thulasidasan, S. Strong edge coloring for channel assignment in wireless radio networks. Proceedings of the 4th Annual IEEE International Conference Pervasive Computing and Communications (PerCom) Workshops;
- [9]. Bertossi, A.; Pinotti, M.; Rizzi, R. Channel assignment on strongly-simplicial graphs. Proceedings of International Parallel and Distributed Processing Symposium (IPDPS);

- [10]. Hsu, C-C.; Liu, P.; Wang, D. wei; Wu, J-J. Generalized edge coloring for channel assignment in wireless networks. Proceeding of the International Conference on Parallel Processing (ICPP);
- [11]. Krumke SO, Marathe MV, Ravi SS. Models and approximation algorithms for channel assignment in radio networks. *Wireless Networks*. 2001; 7:575–584.
- [12]. Ramachandran, KN.; Belding, EM.; Almeroth, KC.; Buddhikot, MM. Interference-aware channel assignment in multi-radio wireless mesh networks. Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM);
- [13]. Subramanian, AP.; Gupta, H.; Das, SR. Minimum interference channel assignment in multi-radio wireless mesh networks. Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON);
- [14]. Xu, L.; Xiang, Y.; Shi, M. A novel channel assignment algorithm based on topology simplification in multi-radio wireless mesh networks. Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC);
- [15]. Huang, H-J.; Cao, X-L.; Jia, X-H.; Wang, X-L. A bibd-based channel assignment algorithm for multi-radio wireless mesh networks. Proceedings of the International Machine Learning and Cybernetics Conference;
- [16]. Shin, M.; Lee, S.; Kim, Y. ah Distributed channel assignment for multi-radio wireless networks. Proceedings of the IEEE International Mobile Adhoc and Sensor Systems Conference (MASS); p. 417-426.
- [17]. Draves, R.; Padhye, J.; Zill, B. Routing in multi-radio, multi-hop wireless mesh networks. Proceedings of the 10th Annual ACM International Conference on Mobile Computing and Networking (MobiCom);
- [18]. Subramanian, AP.; Buddhikot, MM.; Miller, S. Interference aware routing in multi-radio wireless mesh networks. Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh);
- [19]. Raniwala A, Gopalan K, Chiueh T.-c. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review*. 2004; 8:50–65.
- [20]. Raniwala, A.; Chiueh, T.-c. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. Proceedings of the 24th Annual IEEE International Conference on Computer Communications (INFOCOM); p. 2223-2234.vol. 3
- [21]. Alicherry M, Bhatia R, Li LE. Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. *IEEE Journal on Selected Areas in Communications*. 2006; 24:1960–1971.
- [22]. Meng, X.; Tan, K.; Zhang, Q. Joint routing and channel assignment in multi-radio wireless mesh networks. Proceedings of the IEEE International Conference on Communications (ICC);
- [23]. Han, B.; Kumar, V.; Marathe, M.; Parthasarathy, S.; Srinivasan, A. Distributed strategies for channel allocation and scheduling in software-defined radio networks. *INFOCOM 2009, IEEE*; p. 1521-1529.
- [24]. Kodialam, M.; Nandagopal, T. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. Proceedings of the 9th Annual ACM International Conference on Mobile Computing and Networking (MobiCom);
- [25]. Spyropoulos, A.; Raghavendra, CS. Capacity bounds for ad-hoc networks using directional antennas. Proceedings of the IEEE International Conference on Communications (ICC);
- [26]. Yi, S.; Pei, Y.; Kalyanaraman, S. On the capacity improvement of ad hoc wireless networks using directional antennas. Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc);
- [27]. Zhang, J.; Liew, SC. Capacity improvement of wireless ad hoc networks with directional antennae. Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC);
- [28]. Ramanathan R, Redi J, Santivanez C, Wiggins D, Polit S. Ad hoc networking with directional antennas: a complete system solution. *IEEE Journal on Selected Areas in Communications*. 2005; 23:496–506.
- [29]. Das SM, Pucha H, Koutsonikolas D, Hu YC, Peroulis D. Dmesh: Incorporating practical directional antennas in multichannel wireless mesh networks. *IEEE Journal on Selected Areas in Communications*. 2006; 24:2028–2039.

- [30]. Alzoubi K, Li X-Y, Wang Y, Wan P-J, Frieder O. Geometric spanners for wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*. 2003; 14:408–421.
- [31]. Li, X-Y.; Calinescu, G.; Wan, P-J. Distributed construction of a planar spanner and routing for ad hoc wireless networks. *Proceedings of IEEE 21st Annual IEEE Conference on Computer Communications (INFOCOM)*;
- [32]. Schindelhauer C, Volbert K, Ziegler M. Geometric spanners with applications in wireless networks. *Computational Geometry: Theory and Applications*. 2007; 36:197–214.
- [33]. Wang, Y.; Li, X-Y. Distributed spanner with bounded degree for wireless ad hoc networks. *Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS)*;
- [34]. Kumar, U.; Gupta, H.; Das, S. A topology control approach to using directional antennas in wireless mesh networks. *Proceedings of the IEEE International Conference on Communications (ICC)*;
- [35]. Dong, Q.; Bejerano, Y. Building robust nomadic wireless mesh networks using directional antennas. *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM)*;
- [36]. Damian, M.; Flatland, R. Spanning properties of communication graphs induced by directional antennas. *Electronic Proc. of the 20th Fall Workshop on Computational Geometry*; Stony Brook, NY. Stony Brook University;
- [37]. Zhang, S.; Liew, SC.; Lam, PP. Hot topic: Physical-layer network coding. *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom)*;
- [38]. Yao, AC. Technical Report. Stanford University; Stanford, CA, USA: 1977. On constructing minimum spanning trees in k -dimensional spaces and related problems.
- [39]. Arango J, Efrat A, Ramasubramanian S, Pink S, Krunz M. Retransmission and backoff strategies for wireless broadcasting. *Ad Hoc Networks*. 2010; 8:77–95.
- [40]. Ruppert, J.; Seidel, R. Approximating the d -dimensional complete Euclidean graph. *Proceedings of the 3rd Canadian Conference on Computational Geometry (CCCG)*;
- [41]. Abouelaoualim A, Das KC, Faria L, Manoussakis Y, Martinhon C, Saad R. Paths and trails in edge-colored graphs. *Theoretical Computer Science*. 2008; 409:497–510.
- [42]. Gabow, HN. Data structures for weighted matching and nearest common ancestors with linking. *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*;
- [43]. Varadarajan, KR. A divide-and-conquer algorithm for min-cost perfect matching in the plane. *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*;
- [44]. Varadarajan, KR.; Agarwal, PK. Approximation algorithms for bipartite and non-bipartite matching in the plane. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*;
- [45]. Edmonds J. Maximum matching and a polyhedron with 0, 1 vertices. *Journal of Research of the National Bureau of Standards*. 1965; 69 B:125–130.
- [46]. Edmonds J. Paths, trees, and flowers. *Canadian Journal of Mathematics*. 1965; 17:449–467.
- [47]. Gabow, HN. Ph.D. thesis. Stanford University; 1974. Implementation of algorithms for maximum matching on nonbipartite graphs.
- [48]. Lawler E. *Combinatorial Optimization. Networks and Matroids*, Holt, Rinehart, Winston. 1976
- [49]. Derigs U. A shortest augmenting path method for solving minimal perfect matching problems. *Networks*. 1981; 11:379–390.
- [50]. Shieber, B.; Moran, S. Slowing sequential algorithms for obtaining fast distributed and parallel algorithms: maximum matchings. *Proceedings of the 5th annual ACM Symposium on Principles of Distributed Computing (PODC)*;

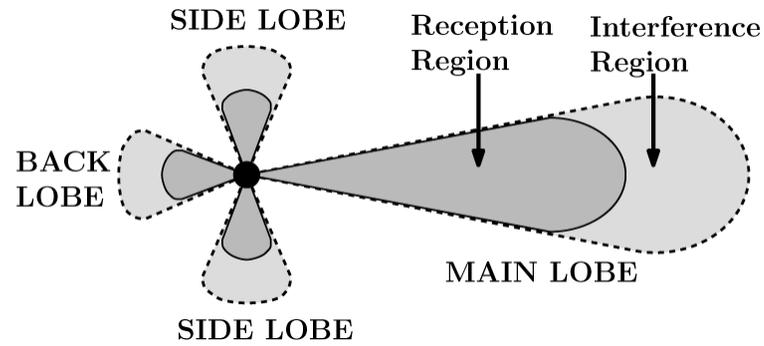


Figure 1. Beam patterns of directional antennas. Signal is reinforced in the main lobe and transmitted weakly in the side and back lobes. The darker regions represent regions of reception and nodes in the lighter regions experience interference.

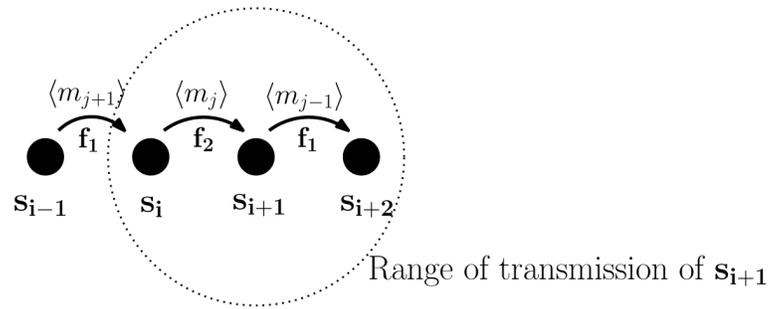


Figure 2.

A streaming application where node s_i forwards message $\langle m_j \rangle$ to node s_{i+1} using channel f_2 and receives message $\langle m_{j+1} \rangle$ at the same time from s_{i-1} using channel f_1 . Node s_{i+1} uses channel f_1 to send message $\langle m_{j-1} \rangle$ received earlier to s_{i+2} causing interference at s_i .

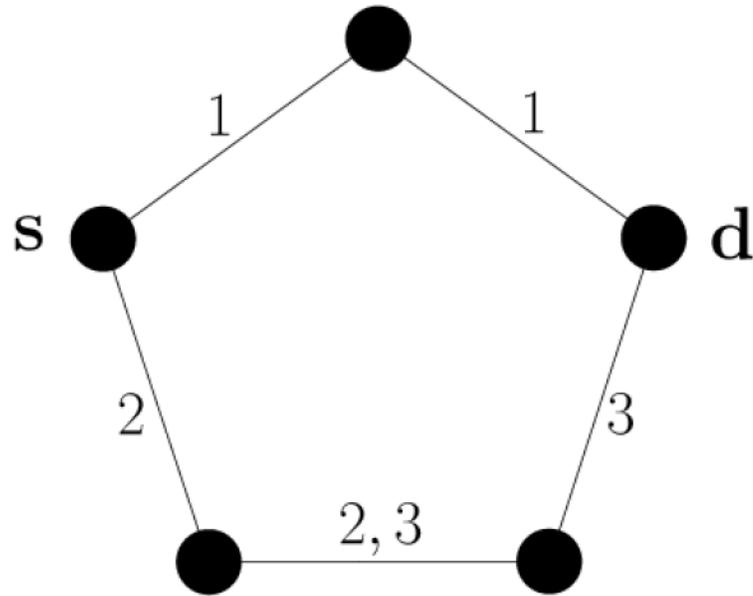


Figure 3. A connected network which contains no CDC-path from s to d . The labels next to each link represent available channels on the link.

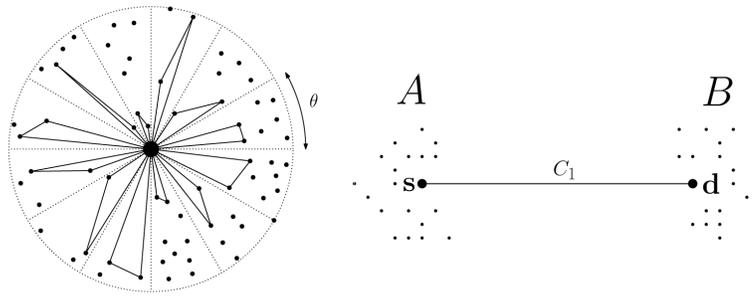


Figure 4.

Spanner construction at a node and the degenerate case when the shortest CDC-path between two nodes s and d is only one edge.

- (a) The partition of the region around the node into sectors of angle θ . We connect to the two closest nodes in each sector as shown.
- (b) s and d are located in clusters A and B respectively. The only path from s to d is the edge (s, d) which is not present in $CDCY G_k$.

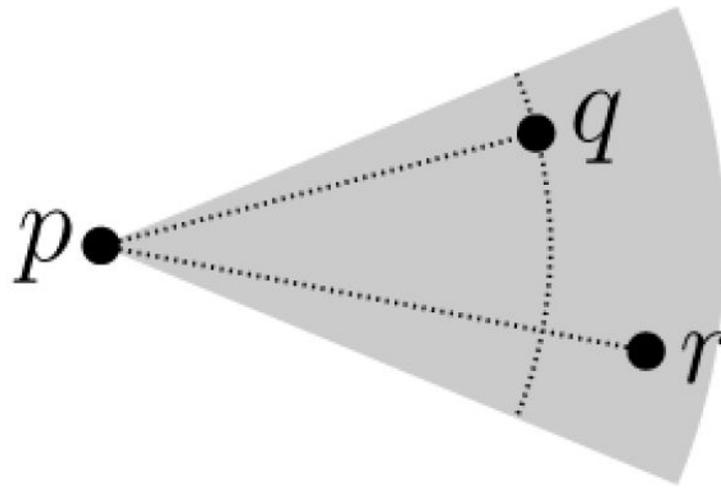


Figure 5.
Nodes q and r are in a sector of p and $|p - q| < |p - r|$.

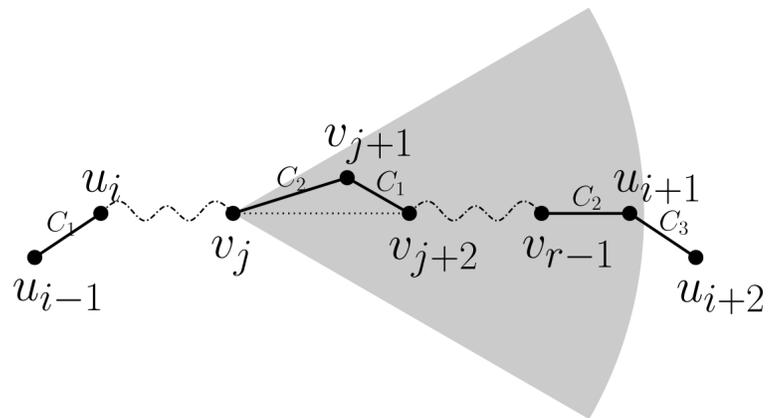


Figure 6.

A CDC-path from u_i to u_{i+1} constructed by repeatedly adding the links (v_j, v_{j+1}) and (v_{j+1}, v_{j+2}) from v_j until u_{i+1} is a neighboring node of v_j . v_{j+1} and v_{j+2} are the two nearest nodes of v_j in the sector of v_j containing u_{i+1} whose set of channels is the same as either u_i or u_{i+1} .

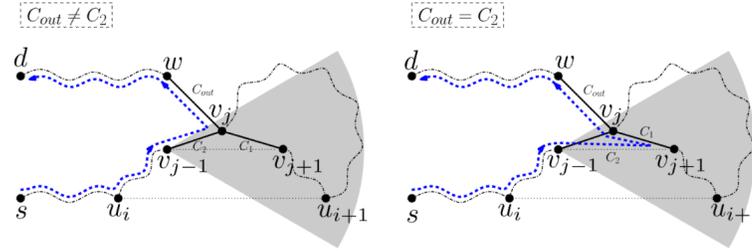


Figure 7.

The case where \mathcal{P}' intersects itself at node v_j , in $\mathcal{P}'_{u_i, u_{i+1}}$, where j is odd. Path \mathcal{P}' is shown as a BLACK solid path and \mathcal{P}'' which does not intersect itself at any node in $\mathcal{P}'_{u_i, u_{i+1}}$ is shown as a BLUE dashed path.

(a) The removal of overlaps when $C_{out} \neq C_2$. The path

$\mathcal{P}'' : \{s, \dots, u_i, \dots, v_{j-1}, v_j, w, \dots, d\}$ (marked as BLUE dashed path) removes the portion of \mathcal{P}' from v_j back to itself.

(b) The removal of overlaps when $C_{out} = C_2$. The path

$\mathcal{P}'' : \{s, \dots, u_i, \dots, v_{j-1}, v_{j+1}, v_j, w, \dots, d\}$ (marked as a BLUE dashed path) consists of the portion of \mathcal{P}' from s to v_{j-1} , the links (v_{j-1}, v_{j+1}) and (v_{j+1}, v_j) followed by the portion of \mathcal{P}' from v_j to d .

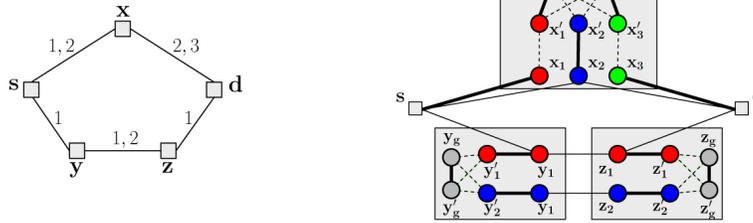


Figure 8. Example network with source node s and destination node d and its ES expansion. The bold links in the expansion graph show the links in the perfect matching set. Observe that the sequence of external links that are in the matching provide the CDC-path from s to d : $s-x-d$.

Algorithm SHORTEST-CDC

```
Set  $s \in S$  and  $\forall u \neq s, u \in F$   
Set  $d_S[s] = 0$  and  $d_S[u] = d_T[u] = \infty$  for all  $u \in F$   
Execute find_min. Let  $(u, v)$  be the edge for which the minimum was attained.  
if both  $u$  and  $v$  are in  $S$  then  
    Execute blossom  
else  
    Execute grow  
Once  $d$  is removed from  $F$ , the algorithm terminates.
```

Figure 9. Algorithm **SHORTEST-CDC** for finding the shortest CDC path from s to d

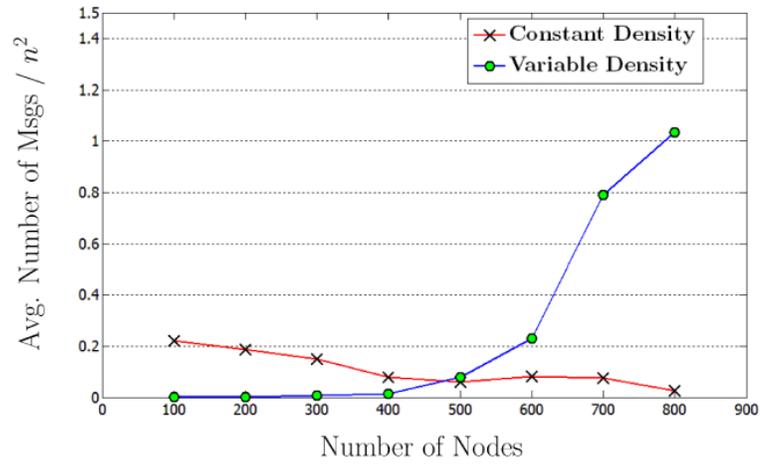


Figure 10.
Avg. Number of Messages/ n^2 vs Number of nodes in the network

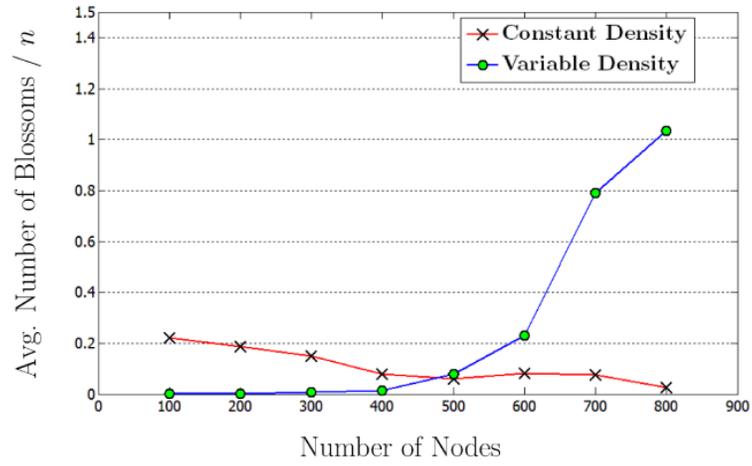


Figure 11.
Avg. number of blossoms/ n^2 vs Number of Nodes