

On Channel-Discontinuity-Constraint Routing in Wireless Networks

Swaminathan Sankararaman* Alon Efrat † Srinivasan Ramasubramanian ‡
Pankaj K. Agarwal §

Abstract

Multi-channel wireless networks are increasingly being employed as infrastructure networks, e.g. in metro areas. Nodes in these networks frequently employ directional antennas to improve spatial throughput. In such networks, given a source and destination, it is of interest to compute an optimal path and channel assignment on every link in the path such that the path bandwidth is the same as that of the link bandwidth and such a path satisfies the constraint that no two consecutive links on the path are assigned the same channel, referred to as “Channel Discontinuity Constraint” (CDC). CDC-paths are also quite useful for TDMA system, where preferably every consecutive links along a path are assigned different time slots.

This paper contains several contributions. We first present an $O(N^2)$ distributed algorithm for discovering the shortest CDC-path between given source and destination. For use in wireless networks, we explain how spatial properties can be used for dramatically expedite the algorithm. This improves the running time of the $O(N^3)$ centralized algorithm of Ahuja et al. for finding the minimum-weight CDC-path. Our second result is a generalized t -spanner for CDC-path; For any $\theta > 0$ we show how to construct a sub-network containing only $O(\frac{N}{\theta})$ edges, such that that length of shortest CDC-paths between arbitrary sources and destinations increases by only a factor of at most $(1 - 2 \sin \frac{\theta}{2})^{-2}$. We propose a novel algorithm to compute the spanner in a distributed manner using only $O(n \log n)$ messages. This scheme can be implemented in a distributed manner using the ideas of [4] with a message complexity of $O(n \log n)$ and it is highly dynamic, so addition/deletion of nodes are easily handled in a distributed manner. An important conclusion of this scheme is in the case of directional antennas are used. In this case, it is enough to consider only the two closest nodes in each cone.

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 [8], video streaming [19], and underground mining [15]. 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. The wireless interference constraint 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 antenna is employed, then there may be at most only one node transmitting on a channel in the vicinity of a node that is receiving on that channel. The interference constraints in wireless networks may be divided into two categories [1]: *inter-flow* and *intra-flow*. The inter-flow interference refers to the

*Department of Computer Science, University of Arizona, swami@email.arizona.edu.

†Department of Computer Science, University of Arizona, alon@cs.arizona.edu.

‡Department of Electrical and Computer Engineering, University of Arizona srini@ece.arizona.edu.

§Department of Computer Science, Duke University pankaj@cs.duke.edu.

scenario where two links belonging to 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.

Prior Work: The problem of routing and channel assignment in WINs refers to computing paths and channel assignment on the paths such that there are no inter-flow and intra-flow interferences. 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 hard when nodes employ omnidirectional antennas, hence is typically solved as two independent sub-problems. For a given set of calls where routing is known, the problem of channel assignment may be mapped to distance-2 vertex and edge coloring problems [7]. Using such a mapping, the objective is to compute channel assignment satisfying the limit on the number of transceivers at each node. Distance-2 vertex and edge coloring problems are well known NP-hard problems, hence the problem of channel assignment for networks employing omnidirectional antennas. 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 [5,6,13,17,21,30,34]. An approach based on balanced incomplete block design is developed in [14] to assign channels for each interface card such that the communication network is 2-edge-connected with minimum interference. In [27], 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, [10] develops a routing protocol to find an efficient path with low intra-flow interference, by taking into account link loss rate, link data rate, and channel diversity. In [29], shortest path with low interference is computed based on an extension to the AODV protocol to account for (inter- and intra-flow) interference and link data and loss rates. In the space of wireless network design for a given static traffic, centralized and distributed approaches for joint channel assignment and routing in the multi-interface WMN with the objective of maximizing throughput are developed in [3,20,23,24], while [16] considers the objective of achieving a given data rate.

It has been shown in [28,36,37] that the capacity of WINs may be further improved by increasing spatial reuse by employing directional antennas. The problem of channel assignment in networks employing directional antennas may be mapped to the edge-coloring problem [22]. In [9], 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 path and channel assignment such that all links can be active simultaneously is NP-complete when nodes employ omnidirectional antennas. Our algorithms are designed for the frequent cases for which the effect of interference can be ignored. For example, when directional antennas are used, as their prices and accuracy are rapidly improving, the nodes may be carefully placed such that any two independent links can use the same channel.

The first work addressing joint routing and channel assignment in WINs under such scenarios is the recent paper by Ahuja et. al. [1]. In this case, the path bandwidth will be the same as an individual link bandwidth if no two consecutive links on the path are assigned the same channel. We refer to the constraint on channel assignment as channel discontinuity constraint (CDC) and any path that satisfies the constraint as a CDC-path. In graph theory literature, CDC-paths are referred to as “properly edge-colored” paths, where channels correspond to colors. From now on, whenever it is mentioned that a CDC-path is found, it is not mentioned explicitly but it is implied that a channel assignment is also found.

We can overcome, to some extent, the difficulties that interference causes in the case of omnidirectional antennas using the known technique of *network coding*. Consider a streaming application, where node s_i forwards the message m_j to node s_{i+1} , and receives the message m_{j+1} at the same time from s_{i-1} . Meanwhile, node s_{i+1} forwards a message m_{j-1} received earlier from s_i to s_{i+2} . In the case where omnidirectional antennas are used, and we have applied the CDC-path protocol, it is quite possible that the frequency of transmission of both s_{i-1} and s_{i+1} is f_1 , as it is different than the frequency f_2 used by s_i (see Figure 1). However, as demonstrated in the paper by [38], since s_i “knows” m_{j-1} it can subtract it from the combined

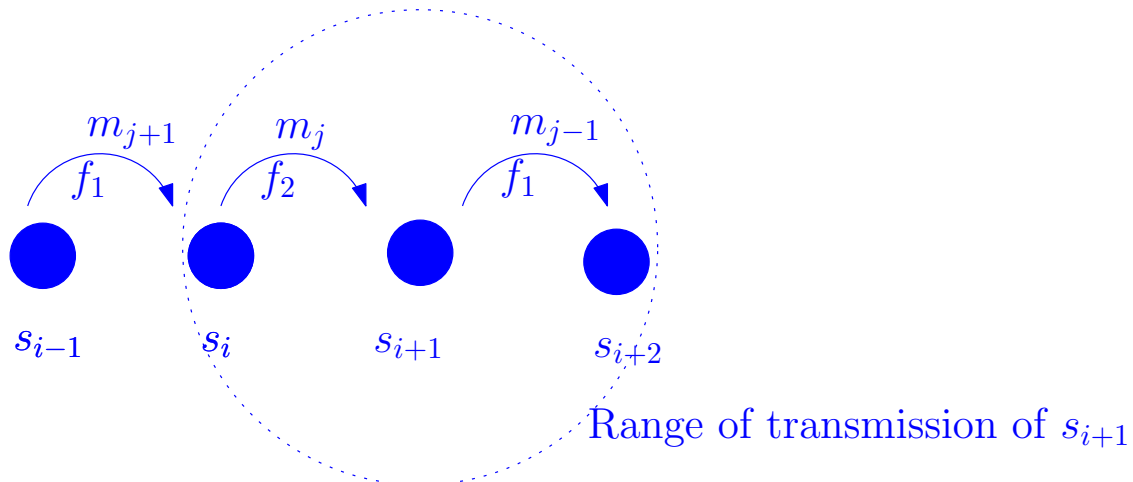


Figure 1: A streaming application where node s_i forwards message m_j to node s_{i+1} using frequency f_2 and receives message m_{j+1} at the same time from s_{i-1} using frequency f_1 . Node s_{i+1} uses frequency f_1 to send message m_{j-1} received earlier to s_{i+2} . This causes interference at s_i .

signal $m_{j-1} + m_{j+1}$. See [38] for details. However, this is not a trivial hardware modification, so we would concentrate in this paper mostly with other applications where CDC path are useful.

Motivation: As stated, CDC-paths are particularly useful in streaming applications where the transmissions are continuous and any period of inactivity reduces bandwidth. CDC-paths are also quite useful in the context of a TDM or 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 along the path and this corresponds exactly to the Channel Discontinuity Constraint where each channel is a time slot. Thus, if (u, v) and (v, w) are adjacent links along the path, the throughput is maximized when they are using different time slots. Now, because of other “conversations” which have ended, certain time slots may become available and finding new CDC-paths may help maintain the maximum bandwidth.

Our Contributions: We present a distributed algorithm for finding the shortest CDC-path between two nodes in a graph. The algorithm requires the exchange of $O(N^2)$ fixed-size messages in total ($O(N)$ per node) where N is the number of nodes. This improves the previous algorithm presented in [1] which is centralized, and requires $O(N^3)$ running time.

The second contribution is the construction of a sparse graph \mathcal{G}' which is a t -Spanner for the problem of CDC routing, where t is a controlled parameter. The number of links in \mathcal{G}' is $O(N)$. By spanner, we mean that for any fixed t as close as one wishes to 1, and for any pair of nodes u and v in the network \mathcal{G} , if \mathcal{G} contains a route between them so does the spanner \mathcal{G}' , and its length is longer than the original route by only a factor of t . Our spanner is based on the Yao graph [35]. The basic idea is that at a node, we divide the region around the node into sectors and store only a few neighbors in each sector. The parameter t is dependent on the number of sectors. This is particularly applicable when using directional antennas. For example, if the antennas’ covering range can be abstracted as a sector of angle 30 degrees, then by storing for every node just the few neighbors in each sector, we retain connectivity and gain routes which are no longer than 4.3 times the theoretical bounds. This spanner is also highly dynamic, so insertion/deletion of links is carried out easily.

The rest of the paper is organized as follows. In Section 2, we discuss the network model and interference constraints. In Section 3, we review the technique developed in [1] that explains how the CDC problem can be expressed as a matching problem. In Section 4, we present a distributed algorithm for finding a single CDC-path between s and d which requires the sending of a total of $O(N^2)$ messages. In Section 5, we discuss a t -spanner for WINs containing $O(N)$ links. Finally, we present the conclusion and some future work.

2 System Model

Consider a multi-channel wireless network, where \mathcal{C} denotes the set of orthogonal channels each with bandwidth B . Let $C = |\mathcal{C}|$. Let \mathcal{V} denote the set of nodes each equipped with C transceivers that may be tuned to any of the orthogonal channels. Let R be the transmission range of a node. We refer to two nodes as *neighbors* if the Euclidean distance between them is not greater than R . Let $\mathcal{G}(\mathcal{V}, \mathcal{L})$ denote the connectivity graph, where \mathcal{L} denotes the set of links. A link connects two neighboring nodes.

WINs may be designed such that the number of links that can be active simultaneously can be maximized. A node z is said to be collinear with a transmission from x to y if node z cannot receive on the same channel as that used by the transmission from x to y . The interference at node z due to the transmission from x to y results in signal-to-noise ratio that is lower than the threshold for decoding the received signal. The collinearity constraint results in the dependence on channel assignment between links that are not adjacent to each other. (hence resulting in distance-2 coloring problems).

While the collinearity constraint is inherent in networks employing omnidirectional antennas, they can be eliminated completely by careful placement of nodes when directional antennas are employed. 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.

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 between neighboring nodes u and v is simply the set of common channels at the two 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 do not care if the communications required for the algorithm satisfy CDC, since, typically, a control channel is used to transmit network-related information rather than the actual data transmissions in the network and the control messages are also typically very short.

3 Problem Formulation

Problem Statement: *Given a multi-channel wireless network with no collinear interference, the set of available channels on every link, the cost of the links, and a node pair (s, d) , find the shortest path between s and d along with a channel assignment on every link of the path such that no two consecutive links in the path are assigned the same channel.*

Definition 1. *We call a path a **CDC-path** if we can assign channels c to each link (u, v) along this path and no two adjacent links have the same channel assigned.*

Ahuja et. al. [1] showed the equivalence of computing the minimum cost CDC-path to the computation of minimum cost perfect matching (MCPM) using Edmonds-Szeider (ES) expansion for nodes. For the sake of completeness, we give a description of their method here. In the ES-expansion graph, we expand each node x , except the source and destination nodes, into $2C + 2$ sub-nodes, denoted $(x_1, x'_1), (x_2, x'_2), \dots, (x_C, x'_C)$, each pair corresponding to a channel $c \in \mathcal{C}$. We also add sub-nodes (x_g, x'_g) . Each pair is connected to each other and we refer to these links as *channel link*. Every sub-node x'_c is also connected to x_g and x'_g . Assign cost 0 to all the above links in the expanded node. Ahuja et. al. [1] also noted that these expansions may be modified to reduce the number of links in the following manner. For links with two or less channels available on them, expand as before. For links with three or more channels, the additional links may be eliminated since, for any CDC-path which uses this link, we can always find a channel assignment from the three corresponding channels.

If two nodes x and y are connected in \mathcal{G} and have a set of channels available between them, then the sub-nodes x_c and y_c corresponding to the available channels are also connected. Connect s and d to the sub-nodes of its neighboring nodes corresponding to the available channels between them.

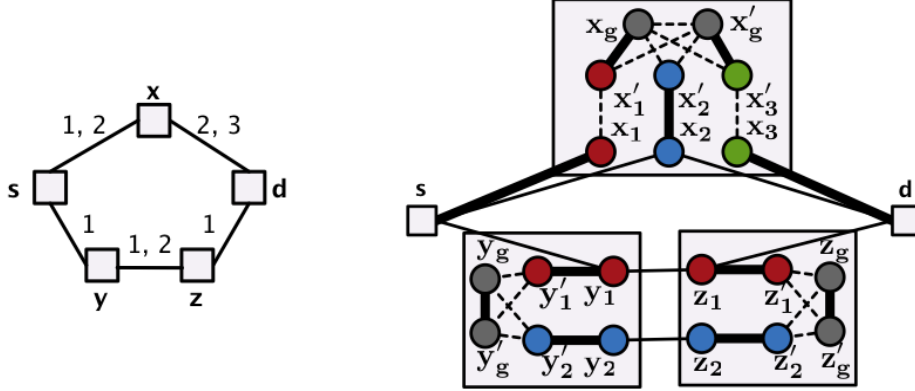


Figure 2: Example network with source node s and destination node d and its ES expansion. The bold links in the expansion graph show the links in the perfect matching set. Observe that the sequence of external links that are in the matching provide the CDC-path from s to d : $s-x-d$.

Now, the problem of computing the shortest CDC-path is the same as computing the minimum-cost perfect matching in the expanded graph. 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.

We illustrate the equivalence of MCPM and CDC-path using an example. Consider the example network and its ES-expansion as shown in Figure 2. 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 shortest CDC-path between s and d in terms of hop length.

The best known sequential implementation of Edmonds' minimum cost perfect matching algorithm is by Gabow [11] with complexity $O(|\mathcal{V}||\mathcal{L}|)$ where $|\mathcal{V}|$ and $|\mathcal{L}|$ are the number of nodes and links respectively. Since the number of nodes in the expanded graph is $O(C|\mathcal{V}|)$ and the number of links is $O(C(|\mathcal{L}| + |\mathcal{V}|))$, the complexity of computing a solution to MCPM is $O(|\mathcal{V}||\mathcal{L}|C^3)$. 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 all the internal edges x_c and $x'_c \forall c$ at node x and x_g and x'_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 [31, 32] for exactly [31] and approximately [32] finding a minimum-weight matching in a geometric setting. In the following sections, we develop a distributed algorithm to compute the shortest alternating path with better bounds than Ahuja et. al. [1].

4 Finding the Shortest CDC Path

We are given the expanded graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ together with the matching \mathcal{M} of zero weight (all vertices except s and d internally matched). For any node u , let $MATE(u)$ denote the mate of u in the matching \mathcal{M} .

Definition 2. Between any two nodes s and d , we define an **alternating path** to be a path with alternating unmatched and matched edges.

Definition 3. 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 shares many ideas with Edmonds' algorithm. It works by finding minimum weight co-link paths similar to each phase of Edmonds' algorithm. Each node u maintains two distances: $d_T[u]$ and $d_S[u]$, corresponding to the minimum weight 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 distances d_S and d_T of its neighbors in S . During the algorithm, certain odd subsets of vertices are termed as *blossoms*.

Definition 4. 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 [18] 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 the corresponding ID of the parent of u , $P_S[u]$. 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 , B is now a subblossom of a new blossom B' . In this case, u stores the edge (w, x) which is the first unmatched edge in B' connected to B and not in B . For more details, see [18].

The algorithm works in three phases: FINDMIN, BLOSSOM and GROW. Before we proceed, let us define a few expressions. For a node u , let

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

Each node u maintains an "examined" state for each neighbor $v \in S, F$. Let $minval[u] = \min_v val[v]$ and $v_{min} = \underset{v \text{ not yet examined}}{\operatorname{arg\,min}} val[v]$. The steps of FINDMIN are as follows.

(Step 1) Each node u computes $minval[u]$ and the corresponding v_{min} .

(Step 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 $b[v] \neq b[u]$.

(Step 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 $v \in S, T$. Then, u finds the minimum $minval[v], v \in N(u)$ where $N(u) = \{v : v \in S, T \text{ and } (u, v) \in \mathcal{E}\}$. 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.

The second stage of the algorithm is the **BLOSSOM** phase which executes the blossom discovery process for an edge (u, v) corresponding to a new outermost blossom Q . This is done by sending messages from u and v towards s till the base of the blossom b is found. The way in which these messages are sent follows the description given by Shieber and Moran [26]. Once this is done, b sends messages throughout the blossom and informs all nodes $u \in Q$ of their membership by assigning $b[u] = b$. In addition, for all new nodes in the blossom, the corresponding parents and the corresponding alternate distance d_s or d_t are assigned.

The **GROW** phase extends the tree by adding one node to T and one to S . Let (u, v) be the edge used to add v to T and $MATE(v)$ to S . v is added to T and $MATE(v)$ to S . The F labels are removed and the

following values are assigned:

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

$MATE(v)$ 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 algorithm works in $O(n)$ iterations. In each one, it executes FINDMIN. Let $\langle u, v, minval \rangle = \arg \min_{u \in S} minval[u]$. If $u, v \in S$, we execute BLOSSOM. Otherwise, we execute GROW. The overall steps of the algorithm are outlined in Figure 3.

Algorithm SHORTEST-CDC

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

Figure 3: Algorithm SHORTEST-CDC for finding the shortest CDC path from s to d

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 [18]. 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 $\{S, T, F\}$. Initially, $s \in S$ and everything else is in F . An edge is termed as tight if $y(u) + y(v) + \sum_{u, v \in Q} z(Q) = 0$. At each step, Edmonds' algorithm searches for tight edges in order to close blossom or grow the tree. If there are no tight edges, it makes a dual change. We choose $\delta = \min(\delta_1, \delta_2, \delta_3)$, where

$$\delta_1 = \min_{\text{non-trivial blossom } Q \in T} \frac{-z(Q)}{2} \tag{1}$$

$$\delta_2 = \min_{u \in S, v \in F} w(u, v) - (y(u) + y(v)) \tag{2}$$

$$\delta_3 = \min_{u, v \in S} \frac{w(u, v) - (y(u) + y(v))}{2} \tag{3}$$

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. $z(Q) = z(Q) + 2\delta$). Now, the only case where a blossom $Q \in T$ is when it shrunk at the start of the algorithm. In our case, there are no shrunk blossoms at the beginning. Hence, we only need to worry about δ_2 and δ_3 .

Lemma 1. 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,

$$y(u) + y(v) + \sum_{u, v \in Q} z(Q) = w(u, v) \quad (4)$$

This, combined with Fact 1 gives the desired result. \square

Note that $d_T[u]$ is the weight of the shortest co-link path to u . This leads to the following lemma whose proof is in the appendix.

Lemma 2. For each node $u \in T$,

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

Using the above lemma, we can prove that our algorithm works in a similar manner to Edmonds' algorithm.

Lemma 3. 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 2,

$$y(s) + y(u') + \sum_{u' \in Q} z(Q) = d_T[u'] \quad (5)$$

The value of δ is given by $\min(\delta_1, \delta_2)$ where δ_1 and δ_2 are given by Equations 2 and 3. This is because, at the start of the algorithm, there are no blossoms. Hence, no non-trivial t -blossoms are found and Equation 1 never occurs. We may compute $\delta = y(s) + \min(\delta_1, \delta_2) = \min(y(s) + \delta_1, y(s) + \delta_2)$.

Taking the case of δ_1 , we have,

$$\delta_1 = \min_{u, v \in S} y(s) + \frac{w(u, v) - (y(u) + y(v))}{2} \quad (6)$$

$$= \min_{u, v \in S} \frac{w(u, v) + (y(s) - y(u)) + (y(s) - y(v))}{2} \quad (7)$$

$$= \min_{u, v \in S} \frac{w(u, v) + d_T[u'] + d_T[v']}{2} \quad (8)$$

The above equations follow from Equation 5 and Lemma 1. In the case of δ_2 , we may similarly compute it as

$$\delta_2 = \min_{u \in S, v \in F} w(u, v) - (y(u) + y(v)) \quad (9)$$

$$= \min_{u \in S, v \in F} w(u, v) - y(u) \quad (10)$$

$$= \min_{u \in S, v \in F} w(u, v) + y(s) - y(u) \quad (11)$$

$$= \min_{u \in S, v \in F} d_T[u] + w(u, v) \quad (12)$$

Since $d_T[u'] = d_S[u]$, Equations 8 and 12 are exactly the values computed by the algorithm SHORTEST-CDC during the FINDMIN phase. Hence, the lemma is proved. \square

Theorem 1. *The algorithm SHORTEST-CDC finds the shortest CDC path from s to t using $O(n^2)$ fixed-size messages.*

Proof. From Lemma 3, 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: (i) The FINDMIN 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 $O(n^2)$. (ii) The GROW phase. Clearly, at each GROW phase, $O(1)$ messages are sent leading to a total message complexity of $O(n)$. (iii) The BLOSSOM phase. Using the approach of [26], for each edge, we can achieve the backward and forward processes using $O(n)$ messages. 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 [26] is for unweighted graphs, since we do not have blossom expansions, their method can be used exactly since the rest of the blossom discovery is identical to unweighted graphs [18]. Each message sent is of fixed size. Hence, the total number of fixed-size messages sent during the algorithm is $O(n^2)$. \square

It is worth noting that if the graph is unweighted or if the edge weights are integers, then we have several advantages. The complexity of FINDMIN is reduced since there is no need to send messages back to s . The blossom discovery process becomes simpler by discovering all blossom edges at the same time. We can find a path using $O(n\frac{\pi}{R} + n \log n)$ messages where π is the length of the shortest $(s - t)$ CDC path.

5 Spanner for CDC Routing

In this section, we discuss a spanner for Channel-Discontinuity-Constraint routing in wireless networks. We are given a graph $\mathcal{G}(\mathcal{V}, \mathcal{L})$ where \mathcal{V} is the set of nodes in \mathbb{R}^2 and \mathcal{L} is the set of links between nodes, and the set of channels $\mathcal{C}(u)$ available at each node u . Let $|u - v|$ denote the euclidean distance between u and v . The weight of every link (u, v) is given to be $|u - v|$. The cost of a path \mathcal{P} is the sum of the weights of the links along \mathcal{P} and is denoted by $|\mathcal{P}|$.

Definition 5. *Let $d_{\mathcal{G}}(u, v)$ denote the cost of the shortest CDC-path from u to v in \mathcal{G} . If there is no such path, then the cost is infinity. We say that a graph $\mathcal{G}'(\mathcal{V}, \mathcal{L}')$ where $\mathcal{L}' \subseteq \mathcal{L}$ is a **CDC t -Spanner** of \mathcal{V} if and only if for every $u, v \in \mathcal{V}$, $d_{\mathcal{G}'}(u, v) \leq t \cdot d_{\mathcal{G}}(u, v)$ where t is a constant.*

Definition 6. *Let $\mathcal{T} = \{T_1, \dots, T_p\}$ denote a partition of nodes in \mathcal{G} 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 . If C is the number of channels available in the network, then, $p \leq 2^C$.*

The above definition of CDC spanners is based on the definition of spanners for Unit Disk Graphs in [33] but is equally relevant in the case of networks using directional antennas. The CDC t -Spanner $CDCYG_k(\mathcal{V}, \mathcal{L}', \mathcal{C})$, which is based on the Yao graph [35], is created as follows. The set of links in $CDCYG_k$ is obtained in the following manner. For each node $v \in \mathcal{V}$, divide the region around v into k interior-disjoint sectors centered at v with opening angle θ . In each sector, connect v to its two nearest neighbors of each type $T \in \mathcal{T}$ and connect these neighbors with a link. 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 $t = (1 - 2 \sin \frac{\theta}{2})^{-2}$. This is particularly applicable when directional antennas are used. 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.

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 4), all edges from A to B may be required. However, no routing algorithm is needed here since s can check to see if d is within range and transmit to it. When not considering the degenerate case, we obtain the following theorem.

Theorem 2. $CDCYG_k$ is a CDC t -Spanner of \mathcal{G} and $|\mathcal{L}'| = O(|V|)$ where $t = (1 - 2 \sin(\frac{\theta}{2}))^{-2}$.

The following lemma was proven by Ruppert and Seidel [25] as a part of their proof of the stretch factor of the Yao Graph.

Lemma 4 (From [25]). *If there are two nodes q and r in a sector whose apex is p and $|p - q| \leq |p - r|$, then $|q - r| \leq |p - r| - (1 - 2 \sin(\frac{\theta}{2}))|p - q|$.*

Let $\mathcal{P} : \{u_0, u_1, \dots, u_{m-1}, u_m\}$ be an (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 $CDCYG_k$ such that $|\mathcal{P}''| \leq t \cdot |\mathcal{P}|$. Note that this is not the algorithm used in practice but is described just to show the existence of such a path. From now on, for every path we create we also assign channels to its links.

The algorithm has two major components:

Link Replacement: Each link (u_i, u_{i+1}) in \mathcal{P} is replaced by a path. The resulting path \mathcal{P}' , obtained by replacing all the links in \mathcal{P} , may not be simple. However, $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$.

Untangling: A path \mathcal{P}'' is obtained from \mathcal{P}' such that $|\mathcal{P}''| \leq |\mathcal{P}'|$ and \mathcal{P}'' is simple.

We divide the links in \mathcal{P} into three cases - (i) the intermediate links (u_i, u_{i+1}) for $i = 1 \dots m - 2$, (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 not the first or last 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 manner. If, for even j , (v_j, u_{i+1}) is a link in \mathcal{G} , the next node is u_{i+1} . Otherwise, the next vertex v_{j+1} of \mathcal{P} is the nearest neighbor of type either $T(u_i)$ or $T(u_{i+1})$ in the sector Ψ of v_j where Ψ is the sector containing u_{i+1} . From v_{j+1} , connect to the next nearest neighbor v_{j+2} to v_j of type $T(v_{j+1})$ the sector Ψ . Repeat for v_{j+2} . This process is demonstrated in Figure 6. Let C_1, C_2, C_3 denote the channel assignment in \mathcal{P} to the links (u_{i-1}, u_i) , (u_i, u_{i+1}) and (u_{i+1}, u_{i+2}) respectively. We may assign channels satisfying CDC to $\mathcal{P}'_{u_i, u_{i+1}}$ as follows (i) C_2 to (v_j, v_{j+1}) and C_1 or C_3 to (v_{j+1}, v_{j+2}) for $j = 0 \dots r - 3$ and (ii) C_2 to (v_{r-1}, u_{i+1}) .

Now, for the second case, if s has exactly one channel available to it, we may only construct the above path through nodes of type $T(u_1)$. In the third case, if d has exactly one channel available to it, then a path from u_{m-1} to d may not exist. However, we may construct a path 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 5, that \mathcal{P}' always exists. We also prove, in Lemma 6, that $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ and we can assign channels to the links in \mathcal{P}' satisfying CDC. However, \mathcal{P}' might not be simple, since when constructing $\mathcal{P}'_{u_i, u_{i+1}}$, no caution is given for not using nodes of $\mathcal{P}'_{u_j, u_{j+1}}$ where $j \neq i$. This problem is resolved in Lemma 7, which shows how to create an (s, d) CDC-path from \mathcal{P}' , without increasing its length. The proofs of these lemmas are in the appendix.

Lemma 5. *If there is a path \mathcal{P} from s to d in \mathcal{G} , then the path \mathcal{P}' from s to d in $CDCYG_k$ constructed by the Link Replacement Procedure exists.*

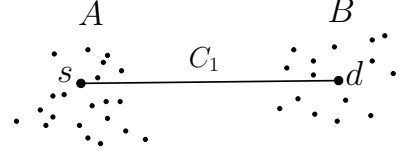


Figure 4: s and d are located in clusters A and B respectively. The only path from s to d is the edge (s, d) which is not present in $CDCYG_k$.

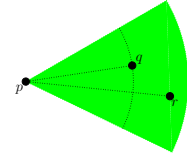


Figure 5: Nodes q and r are in a sector of p and $|p - q| < |p - r|$.

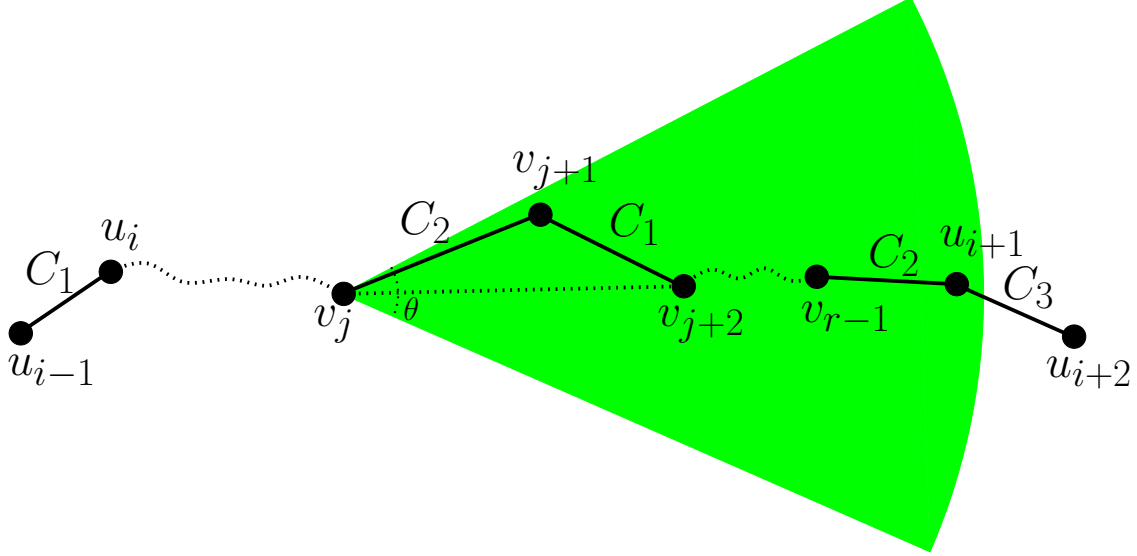


Figure 6: A CDC-path from u_i to u_{i+1} constructed by repeatedly adding the links (v_j, v_{j+1}) and (v_{j+1}, v_{j+2}) from v_j until u_{i+1} is a direct neighbor. v_{j+1} and v_{j+2} are the two nearest neighbors of v_j in the sector of v_j containing u_{i+1} whose set of channels is the same as either u_i or u_{i+1} .

Lemma 6. $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ for $t = (1 - 2 \sin \frac{\theta}{2})^{-2}$ and we can assign channels to the links in \mathcal{P}' satisfying the Channel Discontinuity Constraint.

Lemma 7. 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 (i) $|\mathcal{P}''| \leq |\mathcal{P}'|$, (ii) has fewer such overlaps, and (iii) no two consecutive links are assigned the same channel.

Proof of Theorem 2. 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 $CDCYG_k$ as described in the link replacement procedure, which may not be simple. From \mathcal{P}' , we may apply Lemma 7 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 $CDCYG_k$ from s to d . Note that \mathcal{P}'_{s, u_i} and $\mathcal{P}'_{u_{i+1}, d}$ 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 6 and 7, we know that $|\mathcal{P}''| \leq |\mathcal{P}'| \leq t \cdot |\mathcal{P}|$. Hence, $CDCYG_k$ is a CDC t -Spanner for $t = (1 - 2 \sin \frac{\theta}{2})^{-2}$. Now, for each node v in $CDCYG_k$, we have k sectors. In each sector, there are links from v to two nodes of every type. Hence, there are $O(k|\mathcal{T}|)$ links for each node v . In total, there are $O(k|\mathcal{T}||V|)$ links. \square

Our results are 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 by the directional antennas, will yield a network with almost same properties as the original network. Moreover, a close look of the proof of Theorem 2 reveals that, excluding some convoluted cases, the bounds also hold if the cost of transmitting a message between nodes u, v is proportional to $|u - v|^\beta$, for any $\beta > 1$.

The total number of links in $CDCYG_k$ seems to be $O(k \cdot 2^C \cdot |V|)$ since the total number of types is $O(2^C)$. We now argue that the number of links can actually be bounded by $O(k \cdot C^2 \cdot |V|)$ 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 a CDC t -spanner $CDCYG'_k$. The proof of the following lemma is in the appendix.

Lemma 8. $CDCYG'_k$ is a CDC t -Spanner where $t = \frac{1}{(1 - 2 \sin \frac{\theta}{2})^2}$ and the number of links in $CDCYG'_k$ is $O(k \cdot C^2 \cdot |V|)$.

We now outline the algorithm to compute the spanner in a distributed manner using $O(n \log n)$ messages. Let p be a node, and let $\{\Psi_1 \dots \Psi_{\frac{\pi}{k}}\}$ 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 2 such nodes w , then q broadcasts its distance $|q - p|$. For details on the message complexity of this approach, refer to [4].

6 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 number of channels available in the network is either 10 or 20. 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 $O(C^2 n^2)$ 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 7 shows the ratio of the average number of messages sent to n^2 as a function of the number of nodes in the network. Here, the total number of channels is 10 and each node picks 4 channels. 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 limited. 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 $C^2 n^2$. Figure 8 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 7. As can be seen, it closely follows the graph of Figure 7. This leads us to conclude that the number of blossoms is the major factor in the number of messages.

Figure 9 shows the ratio of Average number of messages to n^2 as a function of the number of channels at a given node. In this case, n is fixed at 300 nodes and the number of channels is varied from 4 to 10. The function appears to be close to linear. This leads us to conjecture that, the number of messages is, in practice, not quadratic in C and could probably be linear.

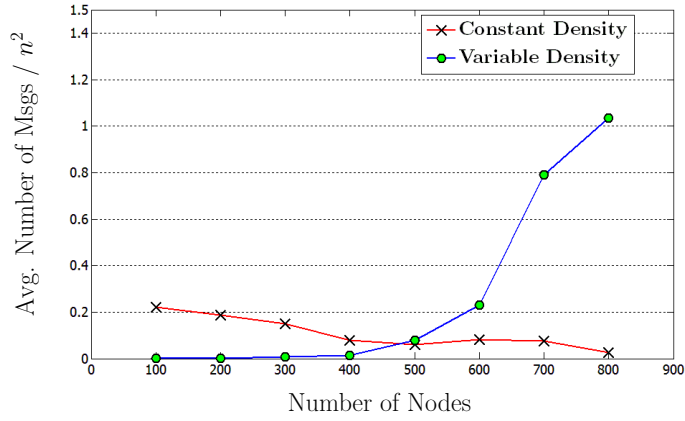


Figure 7: Avg. Number of Messages/ n^2 vs Number of nodes in the network

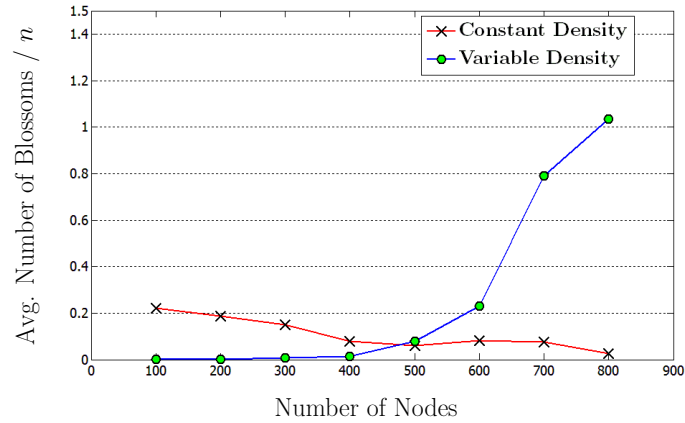


Figure 8: Avg. number of blossoms/ n^2 vs Number of Nodes

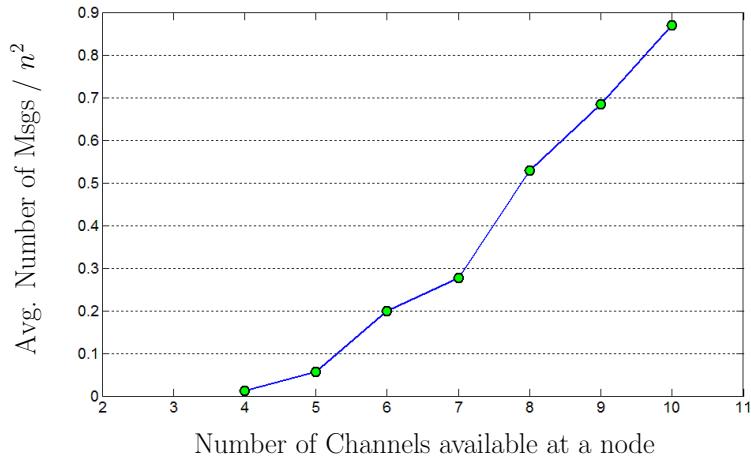


Figure 9: Average number of Messages/ n^2 vs Number of Channels available at a node

7 Conclusion

In this paper, we studied the problem of path selection and channel assignment with channel discontinuity constraint (CDC) in a wireless infrastructure network. Given that any two independent links may share a channel, we showed that the problem of computing a CDC path is equivalent of computing the minimum weight alternating path. We developed a distributed algorithm to compute the minimum weight 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 exchanged per path computation as the network size and number of channels increases. In addition, we develop a t -spanner for CDC routing, where the number of links employed is significantly reduced yet guaranteeing that the minimum-weight CDC path in the spanner is no worse than t times the minimum-weight CDC-path in the original network.

8 Acknowledgements

The work in this paper was partially supported by an NSF CAREER Grant 0348000.

References

- [1] Sandeep K. Ahuja, Abishek Gopalan, and Srinivasan Ramasubramanian. Joint routing and channel assignment in multi-channel wireless infrastructure networks. In *Proceedings of BROADNETS 2008*, 2008.
- [2] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. In *Computer Networks*, 2004.
- [3] M. Alicherry, R. Bhatia, and L. E. Li. Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. In *JSAC*, 2006.
- [4] Jesus Arango, Alon Efrat, Srinivasan Ramasubramanian, Stephen Pink, and Marwan Krunz. Retransmission and backoff strategies for wireless broadcasting. *Ad Hoc Networks*, 8(1):77–95, 2010.
- [5] C. L. Barrett, G. Istrate, V. S. A. Kumar, M. V. Marathe, S. Thite, and S. Thulasidasan. Strong edge coloring for channel assignment in wireless radio networks. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2006.
- [6] A. A. Bertossi, M. C. Pinotti, and R. Rizzi. Channel assignment on strongly-simplicial graphs. In *International Parallel and Distributed Processing Symposium*, 2003.
- [7] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. V. Leeuwen. λ -coloring of graphs. In *The 17th Annual Symposium on Theoretical Aspects of Computer Science*, 2000.
- [8] R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. In *IEEE Communications Magazine*, 2005.
- [9] S. M. Das, H. Pucha, D. Koutsonikolas, Y. C. Hu, and D. Peroulis. Dmesh: Incorporating practical directional antennas in multichannel wireless mesh networks. In *JSAC*, 2006.
- [10] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *International Conference on Mobile Computing and Networking*, 2004.
- [11] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms*, 1990.
- [12] Harold Neil Gabow. *Implementation of algorithms for maximum matching on nonbipartite graphs*. PhD thesis, Stanford University, 1974.

- [13] C.-C. Hsu, P. Liu, D.-W. Wang, and J.-J. Wu. Generalized edge coloring for channel assignment in wireless networks. In *International Conference on Parallel Processing*, 2006.
- [14] H.-J. Huang, X.-L. Cao, X.-H. Jia, and X.-L. Wang. A bibd-based channel assignment algorithm for multi-radio wireless mesh networks. In *International Conference on Machine Learning and Cybernetics*, 2006.
- [15] G. A. Kennedy and P. J. Foster. High resilience networks and microwave propagation in underground mines. In *The 9th European Conference on Wireless Technology*, 2006.
- [16] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *Proceedings of the ACM Mobicom*, 2003.
- [17] S. O. Krumke, M. V. Marathe, and S. S. Ravi. Models and approximation algorithms for channel assignment in radio networks. In *Wireless Networks*, 2001.
- [18] E. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, Winston, 1976.
- [19] N. Mastrorade, Y. Andreopoulos, M. V. D. Schaar, D. Krishnaswamy, and J. Vicente. Cross-layer video streaming over 802.11e-enabled wireless mesh networks. In *ICASSP*, 2006.
- [20] X. Meng, K. Tan, and Q. Zhang. Joint routing and channel assignment in multi-radio wireless mesh networks. In *ICC*, 2006.
- [21] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *INFOCOM*, 2006.
- [22] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad hoc networking with directional antennas: A complete system solution. *JSAC*, 23:496–506, 2005.
- [23] A. Raniwala and T.-C. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM*, 2005.
- [24] A. Raniwala, K. Gopalan, and T.-C. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In *ACM SIGMOBILE Mobile Computing and Communications Review*, 2004.
- [25] J. Ruppert and R. Seidel. Approximating the d -dimensional complete Euclidean graph. In *Proc. 3rd Canad. Conf. Comput. Geom.*, 1991.
- [26] Baruch Shieber and Shlomo Moran. Slowing sequential algorithms for obtaining fast distributed and parallel algorithms: maximum matchings. In *PODC '86: Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, pages 282–292, New York, NY, USA, 1986. ACM.
- [27] M. Shin, S. Lee, and Y.-A. Kim. Distributed channel assignment for multi-radio wireless networks. In *MASS*, 2006.
- [28] A. Spyropoulos and C. S. Raghavendra. Capacity bounds for ad-hoc networks using directional antennas. In *IEEE International Conference on Communications*, 2003.
- [29] A. P. Subramanian, M. M. Buddhikot, and S. Miller. Interference aware routing in multi-radio wireless mesh networks. In *2nd IEEE Workshop on Wireless Mesh Networks*, 2006.
- [30] A. P. Subramanian, H. Gupta, and S. R. Das. Minimum interference channel assignment in multi-radio wireless mesh networks. In *SECON*, 2007.
- [31] Kasturi R. Varadarajan. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In *Proceedings of FOCS '98*, 1998.

- [32] Kasturi R. Varadarajan and Pankaj K. Agarwal. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *Proceedings of SODA '99*, 1999.
- [33] Yu Wang and Xiang-Yang Li. Distributed spanner with bounded degree for wireless ad hoc networks. In *IPDPS '02: Proceedings of the 16th International Parallel and Distributed Processing Symposium*, 2002.
- [34] L. Xu, Y. Xiang, and M. Shi. A novel channel assignment algorithm based on topology simplification in multi-radio wireless mesh networks. In *25th IEEE International Performance, Computing, and Communications Conference*, 2006.
- [35] Andrew C Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. Technical report, Stanford University, 1977.
- [36] S. Yi, Y. Pei, and S. Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *ACM international symposium on Mobile ad hoc networking and computing*, 2003.
- [37] J. Zhang and S. C. Liew. Capacity improvement of wireless ad hoc networks with directional antennae. In *IEEE Vehicular Technology Conference*, 2006.
- [38] Shengli Zhang, Soung Chang Liew, and Patrick P. Lam. Hot topic: physical-layer network coding. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006.

A Proof of Lemma 2

Lemma 2. 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 [12]. 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 weight of this path $W(s, u)$ is given by,

$$W(s, u) = \sum_{i=0}^{k-1} w(u_i, u_{i+1}) \tag{13}$$

$$= (w(s, u_1) + w(u_2, u_3) + \dots + w(u_{k-1}, u)) + (w(u_1, u_2) + w(u_3, u_4) + \dots + w(u_{k-2}, u_{k-1})) \tag{14}$$

$$= (w(s, u_1) + w(u_2, u_3) + \dots + w(u_{k-1}, u)) - (w(u_1, u_2) + w(u_3, u_4) + \dots + w(u_{k-2}, u_{k-1})) \tag{15}$$

by Fact 1, since the weights 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 15, we get the result of the lemma, i.e.,

$$d_T[u] = y(s) + y(u) + \sum_{u \in Q} z(Q) \quad (16)$$

since s is not part of any blossoms(it can only be the base of some blossom). \square

B Proof of Lemmas from Section 5

Lemma 5. *If there is a path \mathcal{P} from s to d in \mathcal{G} , then the path \mathcal{P}' from s to d in $CDCY G_k$ 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 neighbors 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 , $|v_j - v| \leq |v_{j-2} - v|$, we have the fact that v is a neighbor of v_j . Hence, there cannot be a case that v_j does not have any neighbors of type $T(v)$. So, we will always be able to reach v and hence, there is a path from u to v in $CDCY G_k$.

In the case where (u, v) is the first link, we may be restricted to paths through nodes of type $T(v)$ and the above reasoning holds good in this case also. In the case where (u, v) is the last link and when v has only one channel, say C_1 , we may not find a path from u to v . If we try to find a path from u to v through nodes of type $T(u)$, then we may get stuck if u has no neighbors of type $T(u)$. On the other hand, if we find a path through nodes of type $T(v)$, we can only assign channel C_1 to links along this path and hence, it will not satisfy CDC. However, we know that $T(u)$ has more than one channel since we are assuming that $m \geq 3$ and hence, there are two or more edges in \mathcal{P} . Therefore, we may find a path from v to u through nodes of type $T(u)$. \square

Lemma 6. *$|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$ for $t = (1 - 2 \sin \frac{\theta}{2})^{-2}$ and we can assign channels to 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 path from u_i to u_{i+1} $\pi : \{u = v_0, v_2, \dots, v_{r-3}, v_{r-1}, v_r = u_{i+1}\}$ consisting of nodes v_j , where j is even, in $\mathcal{P}'_{u_i, u_{i+1}}$ together with the node u_{i+1} . Note that π is not necessarily a CDC-path. The links (v_j, v_{j+2}) along with the link (v_{r-1}, u_{i+1}) as shown in Figure 6 constitute the links in π . For each node v_j on this path, from Lemma 4, we know that

$$|v_{j+2} - u_{i+1}| \leq |v_j - u_{i+1}| - (1 - 2 \sin \frac{\theta}{2})|v_j - v_{j+2}| \quad (17)$$

See Figure 6. Now, by summation over all even j , we get

$$\begin{aligned} & \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_{j+2} - u_{i+1}| \\ & \leq \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - u_{i+1}| - \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} (1 - 2 \sin \frac{\theta}{2})|v_j - v_{j+2}| \end{aligned} \quad (18)$$

Rearranging (details omitted), we get

$$\sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - v_{j+2}| \leq c \cdot |u_i - u_{i+1}| \quad (19)$$

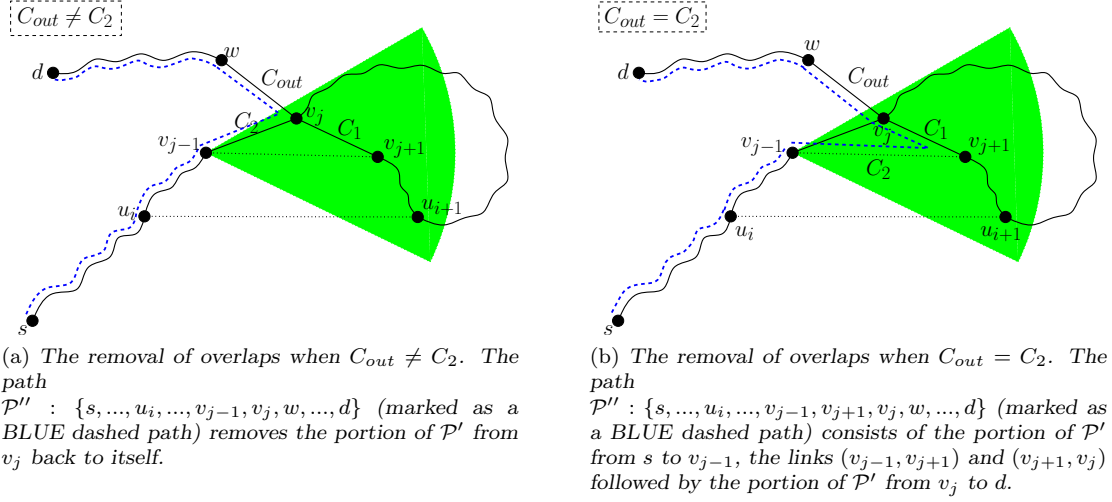


Figure 10: The case where \mathcal{P}' intersects itself at node v_j , in $\mathcal{P}'_{u_i, u_{i+1}}$, where j is odd. Path \mathcal{P}' is shown as a BLACK solid path and \mathcal{P}'' which does not intersect itself at any node in $\mathcal{P}'_{u_i, u_{i+1}}$ is shown as a BLUE dashed path.

where $c = (1 - 2 \sin \frac{\theta}{2})^{-2}$. Now, we have in π , for each v_j, v_{j+2} , where j is even and $0 \leq j \leq r - 3$, from Lemma 4,

$$\begin{aligned} |v_{j+1} - v_{j+2}| &\leq |v_j - v_{j+2}| - \frac{1}{c} \cdot |v_j - v_{j+1}| \\ \implies c \cdot |v_j - v_{j+2}| &\geq |v_j - v_{j+1}| + |v_{j+1} - v_{j+2}| \end{aligned} \quad (20)$$

since $\frac{1}{c} \leq 1$. Combining equations 19 and 20, we get . Hence,

$$\begin{aligned} \sum_{j=0}^{r-2} |v_j - v_{j+1}| &\leq c \cdot \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - v_{j+2}| \\ \sum_{j=0}^{r-1} |v_j - v_{j+1}| &\leq c \cdot |v_{r-1} - u_{i+1}| + \sum_{\substack{j=0 \\ j \text{ is even}}}^{r-3} |v_j - v_{j+2}| \\ &\leq c^2 \cdot |u_i - u_{i+1}| \end{aligned} \quad (21)$$

Hence, $|\mathcal{P}'_{u_i, u_{i+1}}| \leq t \cdot |u_i - u_{i+1}|$ for $t = (1 - 2 \sin \frac{\theta}{2})^{-2}$. Consider that 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, u_{i+1}) are assigned the channels C_1 and C_3 respectively. We now have a channel assignment for the path π because for every set of nodes v_j, v_{j+1}, v_{j+2} for $i = 0 \dots r - 3$ along this path when i is even, we can assign channels C_2, C_1 or C_2, C_3 depending on whether v_{j+1} and v_{j+2} are of type $T(u_i)$ or $T(u_{i+1})$. Finally, we assign the channel C_2 to the link v_{r-1}, u_{i+1} . This channel assignment is shown in Figure 6. Hence, repeating this over all links in \mathcal{P}' , we have a channel assignment for links in \mathcal{P}' and $|\mathcal{P}'| \leq t \cdot |\mathcal{P}|$, for $t = (1 - 2 \sin \frac{\theta}{2})^{-2}$. \square

Lemma 7. *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 (i) $|\mathcal{P}''| \leq |\mathcal{P}'|$, (ii) has fewer such overlaps, and (iii) 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 10. There are three cases for C_{out} - (i) $C_{out} \neq C_1$ and $C_{out} \neq C_2$, (ii) $C_{out} = C_1$ and (iii) $C_{out} = C_2$. Now, we obtain a new path (not necessarily simple) \mathcal{P}'' satisfying CDC 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, v_j and v_{j+1} are in the cone 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 cone of v_{j-1} containing u_{i+1} . If j is odd, the two cases where $C_{out} \neq C_2$ and $C_{out} = C_2$ are depicted in Figure 10.

If $C_{out} \neq C_2$, we obtain \mathcal{P}''_{u_i,v_j} by going 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 10a.

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 10b. 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}, u_{i+1}) , we have $|v_{j-1} - v_{j+1}| \leq |v_{j-1} - v_j| + |v_j - v_{j+1}|$ and $|v_j - v_{j+1}| \leq |v_{j+1} - u_{i+1}| + |u_{i+1} - v_j|$. Adding, we get,

$$\begin{aligned} |v_{j-1} - v_{j+1}| + |v_j - v_{j+1}| &\leq (|v_{j-1} - v_j| + |v_j - v_{j+1}|) + \\ &\quad (|v_{j+1} - u_{i+1}| + |u_{i+1} - v_j|) \end{aligned} \tag{22}$$

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

$$\begin{aligned} |\mathcal{P}''| &\leq |\mathcal{P}'_{s,v_{j-1}}| + (|v_{j-1} - v_j| + |v_j - v_{j+1}|) \\ &\quad + (|v_{j+1} - u_{i+1}| + |u_{i+1} - v_j|) + d_{\mathcal{P}'_{v_j,d}} \end{aligned} \tag{23}$$

Since 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 . \square

Lemma 8. $CDCY'_k$ is a CDC t -Spanner where $t = \frac{1}{(1 - 2 \sin \frac{\theta}{2})^2}$ and the number of links in $CDCY'_k$ is $O(k \cdot C^2 \cdot |V|)$.

Proof. Consider an edge (u, v) which is part of some CDC-path \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 other than the first and last link, (ii) (u, v) is the first link in \mathcal{P} and (iii) (u, v) is the last link in \mathcal{P} . Consider the first case. 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 through nodes which have channels $\{C_a, C_b\}$. This will satisfy CDC because we may assign channels C_a and C_b alternately to edges in this path. If the number of channels shared is two or more, then, by our construction, we may still construct a path as before between u and v through nodes which have any pair of channels from the shared channels. In the remaining cases 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 5, 6 and 7 still hold for these paths. Hence, $CDCY'_k$ is a CDC

t -Spanner for \mathcal{G} . The number of links in $CDCYG'_k$ 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 $CDCYG'_k$ is $O(k \cdot C^2 \cdot |V|)$. □