APPLICATION OF NETWORK CODING FOR VERY-LARGE-SCALE INTEGRATION (VLSI) ROUTING

A Thesis

by

NIKHIL PANDIT NEMADE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2008

Major Subject: Computer Engineering

APPLICATION OF NETWORK CODING FOR VERY-LARGE-SCALE INTEGRATION (VLSI) ROUTING

A Thesis

by

NIKHIL PANDIT NEMADE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Alex Sprintson
Committee Members,	Narasimha Annapareddy
	Jiang Hu
	Riccardo Bettati
Head of Department,	Costas N. Georghiades

May 2008

Major Subject: Computer Engineering

ABSTRACT

Application of Network Coding for Very-Large-Scale Integration (VLSI) Routing.

(May 2008)

Nikhil Pandit Nemade, B.E., Government College of Engineering, Pune, India

Chair of Advisory Committee: Dr. Alex Sprintson

This thesis studies the applications of the network coding technique for interconnect optimization and improving the routability of Very-large-scale integration (VLSI) designs. The goal of the routing process is to connect the required sets of sources and sinks while minimizing the total wirelength and reducing congestion. Typically, chip interconnects include multiple sinks and are routed through intermediate nodes. The main idea of the network coding technique is to enable the intermediate nodes to generate new signals by combining the signals received over their incoming wires. This is in contrast to the traditional approaches, in which an intermediate node can only forward the incoming signals. This thesis attempts to explore the possible benefits of the network coding technique for reducing the total wirelengh and mitigating congestion in VLSI designs.

The contribution of the thesis is three-fold. First, we extend the Hanan's theorem for multi-net rectilinear coding networks. Second, we propose several exact and heuristic solutions for finding near-optimal routing topologies that utilize network coding techniques. Next, we perform extensive simulation study to evaluate the advantage of network coding over the traditional approaches. The simulations help to identify routing instances where the network coding techniques are expected to be beneficial. Finally, we evaluate the potential benefits from network coding in practical settings by analyzing its performance on the International Symposium on Physical Design (ISPD) benchmarks.

Our results show that while network coding shows upto 2.43% improvement on unconstrained rectilinear grids, it shows upto 4.34% improvement in cases with constraints along the grid. In addition, it shows an improvement upto 8.4% in cases involving congestion reduction and also improves routing performance on ISPD routing benchmarks.

To my loving parents

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisor, Dr. Alex Sprintson, for giving me the opportunity to carry out research in the relatively new field of Network Coding. His constant guidance, patience and inspiration have proved to be invaluable towards the progress of the research and the completion of this thesis. He has always supported and encouraged me whenever I needed help and has answered all my questions willingly. In addition, I want to thank him for all the facilities he made available to me in order to facilitate the smooth progress of this research.

I would also like to thank Dr. Jiang Hu who provided valuable inputs with respect to VLSI design which made this research not just possible, but also more practical and meaningful. In addition, I wish to express my sincere gratitude to Dr. Reddy and Dr. Bettati whose co-operation, comments and suggestions played a huge part in successful completion of this work.

My friend, Karandeep Singh has provided valuable help by assisting me whenever I wanted to discuss my ideas and understanding and whenever I was held up while implementing the algorithms throughout the research.

Last but not the least, I would like to thank my family and other friends for their life-long co-operation, constant motivation and timely support.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Motivation 1
B. Previous Work 3
C. Objective 4
D. Organization of Thesis 4

II BACKGROUND 5

A. VLSI Physical Design 5
1. Global Routing 7
2. Routing Techniques Used in Global Routing 10
B. Network Coding 11
1. Algebraic Framework for Network Coding 13
2. Randomized Algorithm for Code Construction 16
3. Deterministic Algorithm for Code Construction 16
4. Coding Advantage 17
5. Encoding Complexity 17

III MULTI-NET ROUTING ON UNRESTRICTED RECTILINEAR PLANE 19

A. Problem Definition 19
B. Extension of Theorems to Network Coding 21
C. An Integer Problem Formulation 23
1. Finding Optimal Coding Network 23
2. Finding Optimal Steiner Tree Packing 25
D. Coding Advantage 27
E. Heuristic Solutions 29

IV MULTI-NET ROUTING ON CONSTRAINED AND CONGESTED TOPOLOGIES 34

A. Routing in the Presence of Hard Blockages 34
B. Routing in the Presence of Soft Blockages 36
C. Routing on a Congested Grid 38
D. Routing to Reduce Density 39

CHAPTER Page

E. Heuristic Solution 41

V PERFORMANCE OF NETWORK CODING IN ISPD BENCHMARKS 45

VI CONCLUSION AND FUTURE WORK 50

A. Conclusion 50
B. Future Work 51

REFERENCES 52

VITA 56

LIST OF TABLES

TABLE Page

I Performance evaluation of heuristic on ISPD benchmark ibm01 . 48

II Performance evaluation of heuristic on ISPD benchmark ibm03 . 48

LIST OF FIGURES

FIGURE		Page
1	VLSI design flow	6
2	Global routing and detailed routing	7
3	Grid graph representation of standard cells	9
4	Comparison of routing with and without network coding	11
5	Point to point connected topology	13
6	An example of a rectilinear coding network	20
7	A example of two collocated vertical segments with $n_l = 2$ and $n_r = 3$. Sliding both segments to the right (shown by horizontal lines) decreases the total wirelength.	22
8	Average wirelength reduction for unrestricted two-dimensional plane	28
9	Example of routing over an unconstrained plane	29
10	Average wirelength reduction in special cases	30
11	Illustration of heuristic solution for problem RCN.	32
12	The performance of the heuristic for problem RCN.	32
13	Example of escape graph	34
14	Wirelength reduction for hard blockages	35
15	Network coding in the presence of hard blockage	36
16	Wirelength reduction for soft blockages	38
17	Wirelength reduction for constrained grid	39

FIGURE Page

18 Density reduction in constrained topologies 40

19 Heuristic solution for constrained grid 42

20 Comparison of heuristic solution with Steiner tree packing . . . 43

21 Local density reduction in ISPD benchmark ibm01 46

22 Histogram of the density of edges before and after the process for ISPD benchmark ibm01 47

23 Histogram of the density of edges before and after the process for ISPD benchmark ibm03 47

CHAPTER I

INTRODUCTION

The goal of the routing process is to find high-quality paths and trees that connect the required sets of sources and sinks. Finding optimal (minimum cost) interconnect topologies is a fundamental problem that spans many areas in VLSI design, networking and transportation engineering. The objective, in most cases, is to find paths and trees that satisfy certain connectivity requirements at minimum cost. The cost of a routing topology typically depends on metrics such as the total length of the interconnects, local congestion and overflow. The related problems have received significant attention from the research community, and efficient routing algorithms have been developed for different settings. In particular, there exist efficient exact algorithms for finding shortest paths [27], minimum spanning trees [27], and approximate algorithms for finding minimum Steiner trees [5].

In this thesis, we extend the traditional routing techniques by applying the novel technique of *network coding* [1]. Network coding generalizes the traditional routing approach by enabling the intermediate nodes to generate new signals by combining the signals received over their incoming wires. In particular, we focus on the applications of the network coding technique for interconnect optimization and improving the routability of VLSI designs.

A. Motivation

Physical design is an important step in the VLSI design flow. The two major steps of physical design are placement and routing. The goal of the placement step is

---

The journal model is *IEEE Transactions on Networking*.

to determine the position of the cells in a way that avoids overlap, minimizes the estimated wirelength and facilitates efficient routing. The routing step includes global and detailed routing. Global routing aims at determining the position pattern of all the nets in the design and minimizing the critical path delay without specifying the actual detailed layout of wires. The detailed routing process determines the detailed wire layout for each net along the path chosen by global routing. The routing process is completed when all the required interconnections are routed and no overflows are present. If a router is unable to deliver a completed routing topology, either additional placement and routing iterations or, in some cases, logic changes are required to complete the routing.

The routing completion capability of a router can significantly affect the design turnaround time. In addition, global routing affects other VLSI design parameters such as power consumption, timing, delays and signal integrity. All these factors make it an important process with respect to the performance of the chip and thus an important target for further improvements.

With the traditional global routing approaches, interconnection of pins in every net is done primarily via rectilinear Steiner trees which typically include intermediate forwarding nodes, referred to as Steiner nodes. Global routing problems are known to be NP-complete [26]. In practice, the complexity of typical routing cases grows significantly with increasing chip size.

The advent of network coding [1] has opened up a wide range of opportunities in various fields of communication and networking. In VLSI designs, it can be employed in the global routing step to enhance the routing with respect to routability, wirelength and congestion. Also, in certain cases, it is possible to obtain a feasible solution only through network coding. These factors provide the motivation for studying the application of network coding for improving the routability of VLSI designs.

B. Previous Work

To the best of our knowledge, only one study considered application of network coding in VLSI design [3]. Specifically, [3] focused on improving the routability of buses in an integrated circuit. In contrast, in this work we study the application of network coding in the global routing stage of the physical design process.

Global routing is an established research area that has attracted a large body of research. A survey of the efforts taken to improve the performance of global routing is provided in [4]. The performance evaluation of several routers which were developed based on traditional routing schemes is available in [8], [9] and [10]. Recent progress in this field includes multi-commodity flow based algorithm [11], multi-level routing [12, 13] and progressive Integer Linear Programming (ILP) based method [8].

Network coding is a new research area, but it already attracted a significant interest from the research community. It has been shown in [1] and [14] that the capacity of a multicast network is equal to the size of the minimum cut that separates the source and each sink in the underlying communication graph. Li *et al.* [14] proved that linear network codes are sufficient for achieving the capacity of the network. In a subsequent work, Koetter and Medard [15] developed an algebraic framework for network coding. This framework was used by Ho *et al.* [16] to show that linear network codes can be efficiently constructed through a randomized algorithm. Jaggi *et al.* [21] proposed a deterministic polynomial-time algorithm for finding feasible network codes in multicast networks. Li *et al.* [2] established an upper bound on the coding advantage in undirected networks. In [18], Langberg *et al.* analyzed the encoding complexity and proved an upper bound on the number of encoding nodes required to achieve network capacity. In our work, we build on these methods and tools to develop develop exact and approximate algorithms for network coding.

C. Objective

The objectives of this research can be summarized as follows:

- Extension of the Hanan's theorem [22] for rectilinear network coding;
- Evaluation of the advantages of network coding over traditional approaches through an extensive simulation study in random topologies;
- Development of exact and heuristic solutions for the implementation of network coding;
- Performance evaluation of network coding technique on ISPD routing benchmarks [31].

D. Organization of Thesis

The thesis is organized as follows. Chapter II provides the introductory background on in VLSI Physical Design as well as on the field of network coding. Chapter III focuses on the problem of multi-net routing on an unrestricted rectilinear plane. Chapter IV addresses the problem of multi-net routing on a congested and constrained grid. In Chapter V we evaluate the performance of network coding technique on ISPD routing benchmarks. Finally, the conclusions and directions for future work appear in Chapter VI.

CHAPTER II

BACKGROUND

In this chapter we discuss several basic concepts of VLSI Physical design and network coding. We begin by a brief discussion of the VLSI design flow. Next, we present a detailed description of the routing process. Finally, we introduce the concept of network coding.

A. VLSI Physical Design

The standard cell design flow is shown in Figure 1. The design process starts with the implementation of the design either by Register Transfer Logic (RTL) coding or through a schematic capture tool. The RTL code is then synthesized to generate gate level netlists using target libraries. The gate level circuit is simulated to verify correct functionality of the design. After the circuit is simulated and verified, the process enters the physical design phase. We will specifically focus our discussion on this phase because it is in this phase that routing is performed on the chip. Physical design is an important step in the standard-cell VLSI design flow and includes there major components: partitioning, placement and routing.

Partitioning aims at decomposing the whole design into smaller subsystems which can be developed independently. However, at the same time, the decomposition scheme needs to minimize interactions between the subsystems in order to reduce the number of interconnections that need to be routed between the systems. The input to the placement process is a set of blocks that need to interact with each other. Placement and floor-planning determine the cells to be used for these blocks. These processes also choose the terminals of a block to be used for interconnections. These terminals are called *pins*. The aim of this step is not only to minimize area utilization,

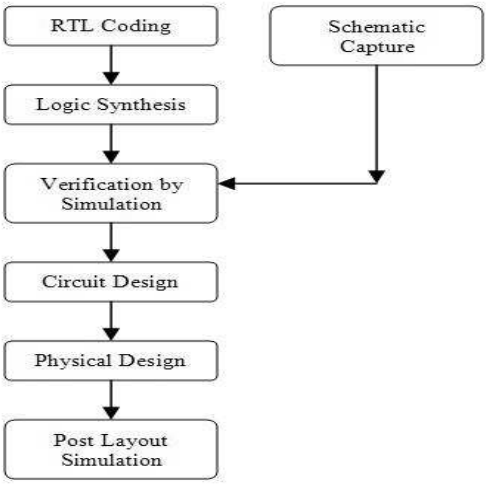


Fig. 1. VLSI design flow

but also to minimize the estimated wire-length required for interconnections and thus ease routing.

Once the placement phase is complete, the routing phase can be invoked. A set of pins which need to be connected to each other is called a *net* and a list of all nets to be routed is called a *netlist*. The space between cells or blocks on a chip is referred to as a *channel* and routing aims at establishing a set of wires in the channels so as to interconnect all nets in the netlist without any overflow. Routing needs to take into consideration factors such as channel capacities, wire widths and wire crossings. The routing stage is divided into global routing and detailed routing. Global routing chooses the channels to be used for routing of all the nets in the design. It aims at minimizing the cost of routing without specifying the actual detailed layout of wires. The cost of routing may depend on metrics such as wirelength, wire congestion or path delay to name a few. The detailed wire layout for each net along the path chosen by global routing is then determined in the detailed routing process. For example,

Figure 2(a) shows the outcome of the global routing process whereas Figure 2(b)shows the result after detailed routing has been completed. At the end of detailed routing, we expect the routing to be *completed*.

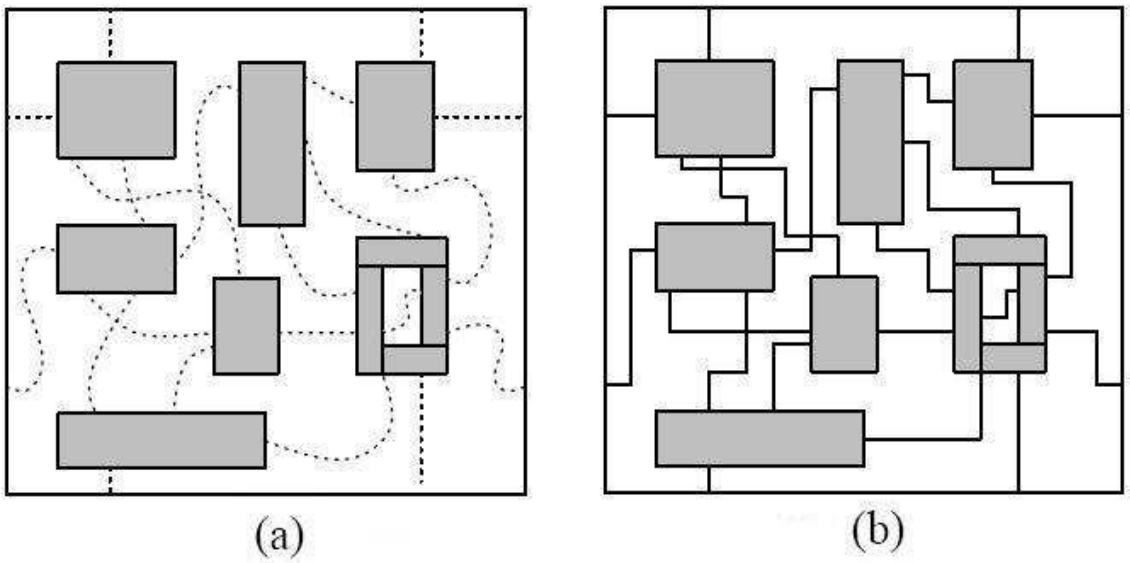


Fig. 2. Global routing and detailed routing

Completion of routing implies establishing all the required interconnects without causing any channel overflows or violation of physical or electrical properties. If a router is unable to deliver a completed routing topology, either additional placement and routing iterations or, in some cases, logic changes may be required after which routing is performed again. Since we intend to improve the outcome of global routing through the usage of network coding, we shall discuss this section is greater depth.

1. Global Routing

As discussed earlier, global routing aims at establishing an approximate routing topology to interconnect each net in the netlist. However, global routing is a well known

NP-complete problem [26] and the complexity of typical routing cases grows significantly with increasing chip size. To ease the process of global routing, the geometric information of a chip can be represented as a graph. Various models exist to represent the chip in this manner which include

1. Grid Graph model;
2. Checker Board Model;
3. Channel Intersection graph model.

We choose the Grid Graph model as it is convenient to use for standard cell, gate array and sea of gates design approach where the cell arrangement is regular. Thus we will continue the discussion specifically with respect to the Grid Graph model.

In the standard cell approach, each module is connected through a pin. Thus, each module is represented by a vertex in the graph. Two vertices are connected by an edge if their corresponding modules are adjacent to each other. Thus, an edge represents a boundary between two cells. Figure 3(a) shows the standard cell representation whereas Figure 3(b) shows the grid graph representation of a chip.

Thus, it can be seen from Figure 3 that the standard cell representation has been converted to a rectilinear graph and that global routing needs to be performed along this rectilinear graph. The problem now can be formulated as:

Problem RST (Rectilinear Steiner Tree). *Given a graph $G(V, E)$ where each vertex $v \in V$ represents a cell and each edge $e \in E$ represents a boundary between two adjacent cells, for each net $n \in N$ where N is the netlist, construct a tree of minimum cost which interconnects all the pins of the net.*

As these trees are constructed, a boundary between two cells may be crossed by edges of trees used for routing different nets. Thus, in the grid graph model, this

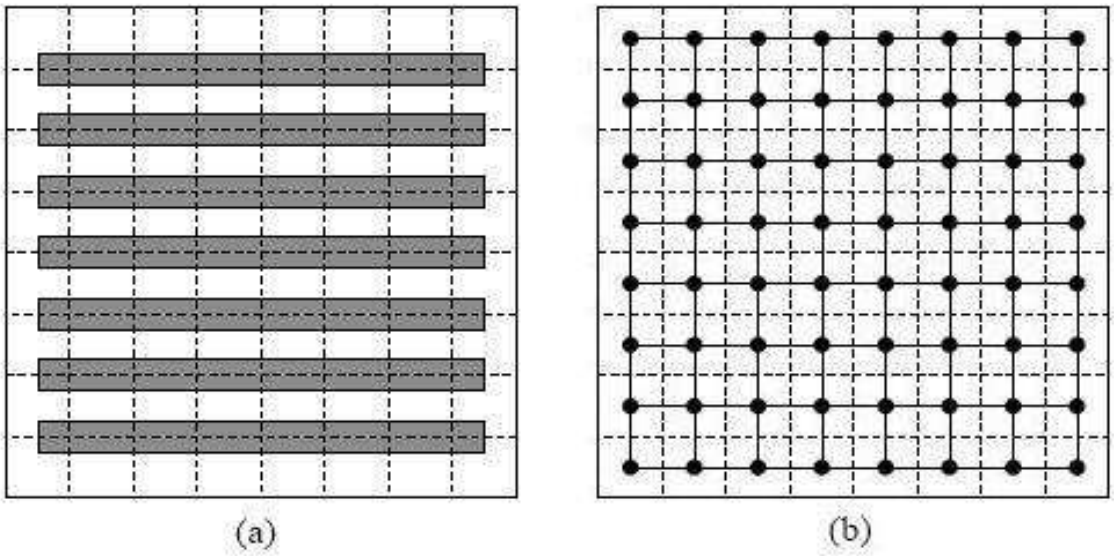


Fig. 3. Grid graph representation of standard cells

would be equivalent to an edge $e \in E$ being utilized multiple times. The number of times a boundary is crossed and thus an edge in the grid graph used is called the flow through the boundary or edge and is denoted as $f(e)$. However, each boundary (and thus an edge in the grid graph) has an upper bound on the number of wires it can accommodate. This upper bound is called the capacity of the boundary or capacity of an edge and is denoted as $u(e)$. The ratio $f(e)/u(e)$ gives the density of utilization of edge e. For any edge $e \in E$, if $f(e)$ exceeds $u(e)$, we say that an *overflow* occurs on that edge. The value of this overflow is given as $o(e) = f(e) - u(e)$. Thus, the fundamental requirement of global routing is to route all the nets in the netlist without causing overflow on any edge $e \in E$.

In addition to completing the routing without causing overflows, there may be other objectives depending on the employed design style. Congestion management is an important objective usually employed across all design styles. It aims at evenly distributing the density uniformly across all edges so as to simplify the detailed routing

step. In addition, total wirelength reduction is another important objective aimed at reducing both, delay and power consumption. Thus, the cost metrics are designed based on a linear combination of both these objectives. There are different techniques used to construct trees for routing nets. These include maze routing [4], Steiner tree packing [6], integer linear program (ILP) based methods [8] and network flow based methods [11].

2. Routing Techniques Used in Global Routing

Maze routing is the first technique which we will discuss. Each edge $e \in E$ has a cost associated with it depending on the cost metric. The costs of edges through blockages are set to infinity. This technique considers two pins at a time and uses Dijkstra's algorithm [23] to find the shortest path between the two pins. This technique guarantees to find the optimal shortest path if one exists. A net having multiple pins thus needs to be decomposed into pairs of pins over which Dijkstra's algorithm can be applied. However, since the size of the global routing problem can be large at times, maze routing can be very costly in terms of memory utilization and run time. Many efforts [28] have attempted to improve its speed and memory usage. Besides, the generated tree may not be optimum for nets having more than two pins.

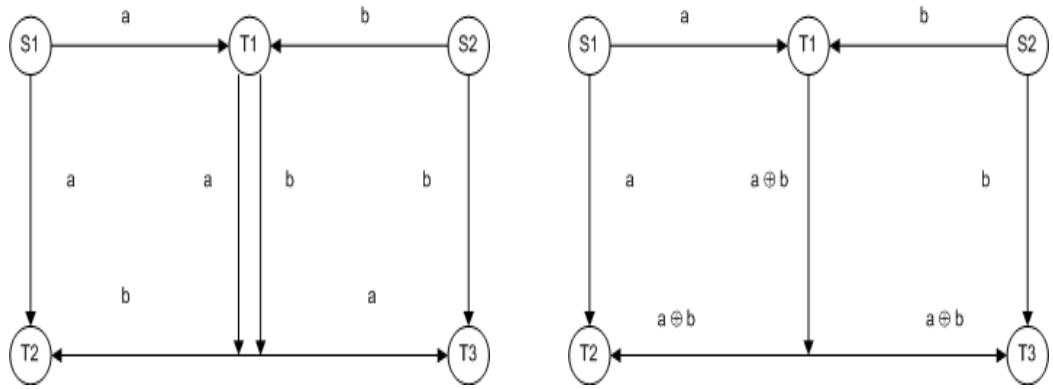
The next technique we will discuss is called Steiner tree construction. This technique can be applied to nets with more than two sinks. A Steiner tree is similar to a minimum spanning tree except for the fact that it needs to connect source nodes to a limited set of sinks. It can have extra intermediate nodes (other than sources and sinks) to minimize the total wirelength. The intermediate nodes of degree more than two are referred to as Steiner nodes. Our goal is to find a rectilinear minimum Steiner tree to interconnect all the pins of a net. This problem is known to be NP-complete [24]. However, Hanan [22] proved that a minimal rectilinear Steiner tree can

be constructed by using only a limited set of nodes obtained by drawing horizontal and vertical lines through the actual nodes to be interconnected. These nodes are referred to as Hanan points and they reduce the search set for potential Steiner nodes. However, considering that the problem is NP-complete, various heuristic solutions exist to obtain near-optimal Steiner trees.

In the next section, we will discuss the technique of *network coding* and present its basic concepts which will help us better understand how it can be applied in the area of global routing.

B. Network Coding

The advent of network coding [1] opens a wide range of opportunities in many areas of communication and networking. The basic idea of network coding is to enable the intermediate nodes to generate new signals by combining the signals arriving over their incoming wires. This is in contrast to the standard approach in which an intermediate node can only forward or duplicate incoming signals.



(a) Routing example without network coding

(b) Routing example using network coding

Fig. 4. Comparison of routing with and without network coding

Figure 4(a) shows the solution using traditional routing approaches whereas 4(b) shows the solution using network coding. We have two sources $S1$ and $S2$ which need to transmit signals a and b respectively to each of the three sinks, $T1$, $T2$ and $T3$. Note that this is an example of a routing problem for two multi-pin nets. Assume that each edge in the network has an unlimited capacity. Assume also that the cost of sending one signal across each edge is equal to d. Each edge is associated with a cost of d for using it. The routing topologies presented in the figure allow each source to transmit its symbols to all the 3 sinks. Note that with the traditional approach the total cost of $8d$, while with the network coding approach, the total cost is only $7d$. This is because, with the traditional routing approach the outgoing link from node T1 is used twice: once for signal a and once again for signal b. On the other hand, with network coding, node $T1$ generates a new signal $a \oplus b$ and transmits it only once along its outgoing edge and to nodes $T2$ and $T3$. The nodes $T2$ and $T3$ receive the encoded signal along with either signal a or b and thus can decode both signals. Thus, we pay only once for each link. As a result, the network coding technique allows us to reduce the total wirelength by 12.5

Again, consider the same example as shown in Figure 4. Assume this time that due to congestion, each edge in the network has a capacity of only 1 unit. Under these circumstances, in the absence of network coding, only one source will be able to successfully transmit its signal to all the three sinks. However, if network coding is used, both the sources will be able to send their signals to all 3 sinks. This is because sink $T1$ generates a new signal $a \oplus b$ which is delivered to sinks $T2$ and $T3$ whereas the signals a and b are delivered directly to these sinks. Thus, the sinks will be able to decode the two original symbols from the received signals. After having seen the benefits of employing network coding for routing, we will discuss it in further depth in the following sections.

1. Algebraic Framework for Network Coding

Koetter and Medard [15] have presented an algebraic framework for network coding. To understand the framework, consider the example shown in Figure 5. In this example, $X1$ and $X2$ are the original packets transmitted by the sources whereas $Z1$ and $Z2$ are the packets received at the terminals or sinks. A connection c is defined as $c = (s, v, X)$ where s is the source, v is the terminal or sink and X is the set of packets $(X1, X2)$ which need to be sent. The rate of the connection is denoted by $R(c)$. As these packets pass over the edges of the network, a linear combination of them is generated by multiplication with the edge coefficients. All operations are carried over a finite field of suitable size.

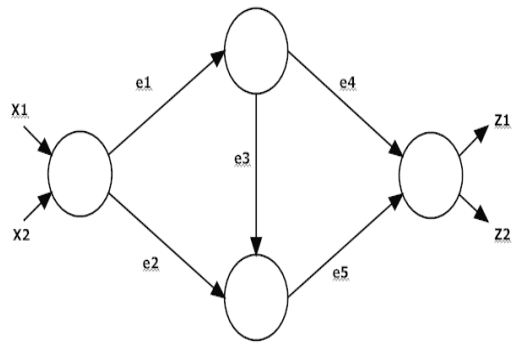


Fig. 5. Point to point connected topology

The linear equations for this topology may be written as

$$
\begin{aligned}
Y(e_1) &= \alpha_{1,e1}X1 + \alpha_{2,e1}X2 \\
Y(e_2) &= \alpha_{1,e2}X1 + \alpha_{2,e2}X2 \\
Y(e_3) &= \beta_{e1,e3}Y(e1) \\
Y(e_4) &= \beta_{e1,e4}Y(e1) \\
Y(e_5) &= \beta_{e2,e5}Y(e2) + \beta_{e3,e5}Y(e3) \\
Z1 &= \epsilon_{e4,1}Y(e4) + \epsilon_{e5,1}Y(e5) \\
Z1 &= \epsilon_{e4,2}Y(e4) + \epsilon_{e5,2}Y(e5)
\end{aligned}
\tag{2.1}
$$

The coefficients can be written in an array representation as follows :

$$
A = \begin{pmatrix} \alpha_{1,e1} & \alpha_{1,e2} & 0 & 0 & 0 \\ \alpha_{2,e1} & \alpha_{2,e2} & 0 & 0 & 0 \end{pmatrix}
$$

$$
B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \epsilon_{e4,1} & \epsilon_{e4,2} \\ \epsilon_{e5,1} & \epsilon_{e5,2} \end{pmatrix}
$$

$$
F = \begin{pmatrix} 0 & 0 & \beta_{e1,e3} & \beta_{e1,e4} & 0 \\ 0 & 0 & 0 & 0 & \beta_{e2,e5} \\ 0 & 0 & 0 & 0 & \beta_{e3,e5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}
$$

where F is the adjacency matrix which gives the relation between two edges of the

network. The relation between the original and received packets is given as

$$[Z] = [A][I + F + F^2][B][X] \tag{2.2}$$

However, as F is nilpotent,

$$I + F + F^2 + ... = (I - F)^{-1}$$

Thus, we can write $[Z] = [A][I - F]^{-1}[B][X]$. The transfer function of the network is given as $[M] = [A][I - F]^{-1}[B]$. The values of the coefficients that make up matrices A,B and $[I - F]^{-1}$ should be chosen such that the determinant of M is nonzero over the field. Usually, A and B are chosen to be unity matrices and the coefficients in $[I - F]^{-1}$ are chosen so that the determinant of M evaluates to a non-zero scalar value. A possible solution in the above example would be to assign $\beta_{e1,e4} = \beta_{e2,e5} = 1$ and $\beta_{e1,e3} = \beta_{e2,e5} = 0$. This is equivalent to finding two disjoint paths from the source to the sinks. Koetter and Medard [15] have also proved the following algebraic Max Flow Min Cut condition for a connection c with rate $R(c)$.

Theorem 1. *The following three statements are equivalent*

1. *A point-to-point connection $c = (s, v, X)$ is possible*
2. *The Min Cut Max Flow bound is satisfied for rate $R(c) = |X|$*
3. *The determinant of the transfer matrix is non-zero over the ring of polynomials $F_2[\xi]$*

The main theorem for multicast says that for a multicast network coding problem, there exists a linear network coding solution over finite field F_{2^m} for a large enough m if and only if there exists a feasible flow between the source and each sink individually.

The algebraic framework lays out a foundation for selecting feasible encoding coefficients, which can be chosen through a deterministic or a randomized algorithm.

2. Randomized Algorithm for Code Construction

Ho [19] has described a randomized coding approach to multicast networks where one or more sources require to send common information to more than one terminal or sink. In this approach, some or all of the coefficients are chosen randomly and independently as long as the fixed coefficients can maintain feasibility of the solution. The network code thus generated is valid if all terminals or sinks are able to reconstruct the source packets. Specifically, [19] shows that if all code coefficients are chosen uniformly and randomly over a finite field F_q, the probability that all terminals or sinks can decode the original packets is at least $(1 - d/q)^\eta$ where d is the number of terminals or sinks and η is the number of links with associated randomized coefficients.

3. Deterministic Algorithm for Code Construction

In [21], Jaggi *et al.* present a polynomial time algorithm for the design of maximum rate linear multicast network codes. This algorithm works in two stages. In the first stage, a set of h disjoint paths is found from the source to each sink $t \in T$. Only the edges in each of these disjoint paths are considered in the second stage.

In the second stage, a greedy algorithm is employed which visits each node in a topological order and selects the linear encoding coefficients for each edge leaving the node. The order of visiting the nodes ensures that the encoding coefficients for an edge e are selected after the encodings for all edges leading to it have been completed. The encoding coefficients for each edge are selected in a way that ensures that all sinks that lie downstream from edge e can obtain linearly independent combinations of the original source symbols.

4. Coding Advantage

The Coding advantage is the ratio of the throughput that can be obtained using network coding to the maximum throughput without achievable using traditional approaches. Li *et al.* [2] have studied the benefits of network coding for undirected networks. They have shown that the coding advantage for multicast communication in such networks is upper bounded by 2. In addition, they have show that network coding cannot provide an advantage for broadcast communication on undirected networks. Agarwal and Charikar [17] have shown that the maximum coding advantage for a given network is equal to the integrality gap of a certain linear programming (LP) formulation for the minimum Steiner tree problem. This holds for both directed as well as undirected networks. Specifically, in the case of directed networks, the coding advantage is equal to the integrality gap of the LP formulation for the directed Steiner tree problem whereas in the case of undirected networks, the coding advantage is equal to the integrality gap of the bi-directed cut LP formulation for the undirected Steiner tree problem.

5. Encoding Complexity

Though network coding is an important technique and can achieve network capacity, it requires a certain number of encoding nodes. The encoding nodes are more expensive than normal forwarding nodes as they must have the capability to generate linear combinations of the packets they receive. They may also cause delays. In [18], Langberg *et al.* proved that the number of encoding nodes required to achieve the capacity of the network is bounded by h^3k^2 where h is the number of packets to be delivered to k sinks, independently of the size of the network. They have also shown that in networks containing cycles, the number of encoding nodes is limited by the

size of the minimum feedback link set. Specifically, they showed that the number of encoding nodes is bounded by $(2B + 1)h^3k^2$ where B is the minimum size of a feedback link set.

In a VLSI design, implementation of network coding requires XOR gates to be added along the routing paths. This means additional logic needs to be added to the design which can increase logic cost and area consumption. However, any VLSI design has buffers along the routing paths to maintain signal strength. The buffers contain NOT gates inside them to drive the wire. These can be replaced with XOR gates which are only slightly costlier, but still cheaper than adding a totally new XOR gate. Thus, the increase in logic and area cost due to network coding can be significantly reduced by using this technique.

CHAPTER III

MULTI-NET ROUTING ON UNRESTRICTED RECTILINEAR PLANE

In this section we consider the problem of multi-net global routing on an unrestricted rectilinear (x, y) plane. We identify and apply the network coding techniques on the nets that have a significant number of shared sink nodes. Since global routing is normally carried out on a tile graph, two nodes can be treated as collocated if they belong to the same tile. Our goal is to route all nets in a way that minimizes the total length of the required wires.

A. Problem Definition

We proceed to formulate the problem of finding a minimum wirelength routing topology for multiple nets that share the same set of sink nodes. As we discuss below, our techniques can be extended to the case in which a large set of sink nodes is shared.

Problem RCN (Rectilinear Coding Network).

Input: *A set $\{n_1, \ldots, n_h\}$ of multi-sink nets, each net $n_i, 1 \leq i \leq h$, needs to deliver a signal a_i from the source node s_i to the common set of sink nodes $T = \{t_1, \ldots, t_k\}$. For each source and sink node, we know the horizontal and the vertical coordinates.*

Find: *A set of vertical and horizontal wires of minimum total length and an encoding scheme that allows delivering each signal $a_i, 1 \leq i \leq h$ from s_i to all sinks in T.*

The standard way to solve Problem RCN is to find a collection of *rectilinear* Steiner trees that connect each $s_i \in S$ to all sinks in T. A *Rectilinear Steiner Tree* (RST) is an acyclic set of segments that connects source $s_i \in S$ to the set of sinks T, such that all segments in the tree intersect only on their endpoints. The length of the tree is equal to the sum of the lengths of its vertical and horizontal segments. A

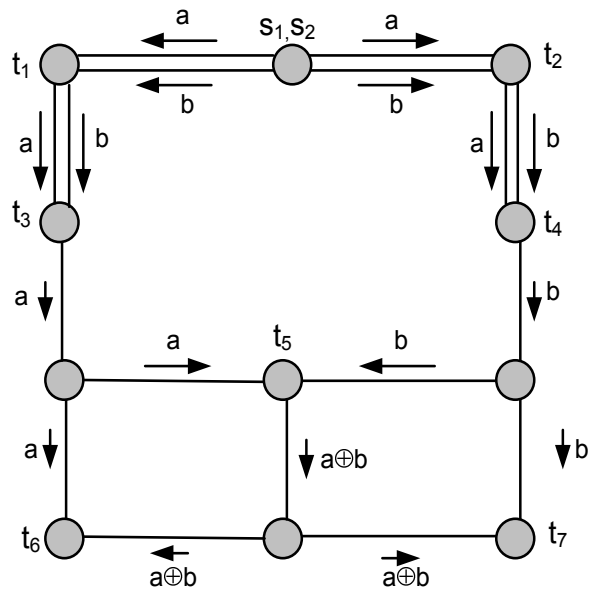


Fig. 6. An example of a rectilinear coding network

Rectilinear Steiner Minimum Tree (RSMT) is a rectilinear Steiner tree of minimum length. As shown earlier, a collection of RSMTs is not necessarily the optimal solution to the problem. Without loss of generality, we assume that any point incident to two collinear segments is a sink. We refer to any point incident to three or more segments which is not a sink as a *Steiner* point. A *corner point* is a point of degree two with non-collinear segments which is not a sink.

The network coding solution generalizes the standard Steiner tree solution by allowing the Steiner nodes to generate new signals. With network coding, a solution to Problem RCN is a set of segments, which is not necessary a union of disjoint Steiner trees. Rather, as shown in [15], a feasible collection A of segments must satisfy the following condition: For each sink $t_i \in T$, the collection A contains h segment-disjoint paths $P_1^i, \ldots, P_h^i$, such that path P_j^i connects source s_j to sink t_i. Note that the paths P_m^i and P_g^j can share common segments if $i \neq j$. Note that the different collinear segments in A can be (partially) collocated. Figure 6 depicts an

example of a rectilinear coding network with two co-located source nodes s_1, s_2 and seven sinks $t_1, \ldots, t_7$.

B. Extension of Theorems to Network Coding

Hanan [22] analyzed the properties of minimum RST trees and showed that for a instance of Problem RST with a source node s and a set T of k sinks, there exists a set of $(k+1)^2$ points, referred to as *Hanan* points, such that there exists a minimum RST tree that connects s and T and all of its Steiner points belong to that set. Specifically, let (x_s, y_s) be coordinates of source s and let (x_t^j, y_t^j) be coordinates of sink $t_j \in T$. A point (x', y') is said to be a Hanan point if $x' \in \{x_s\} \cup \{x_t^j \mid t_j \in T\}$ and $y' \in \{y_s\} \cup \{y_t^j \mid t_j \in T\}$. The Hanan's theorem implies that there exists a minimum RST in which each Steiner point is a Hanan point.

In this section we prove an extension of Hanan's theorem for network coding.

Theorem 2. *For any instance of Problem RCN there exists a minimum rectilinear coding network A such that for any Steiner point (x', y') of A then the following two conditions hold:*

$$\begin{aligned} x' &\in \{x_s^i \mid s_i \in S\} \cup \{x_t^j \mid t_j \in T\} \\ y' &\in \{y_s^i \mid s_i \in S\} \cup \{y_t^j \mid t_j \in T\}, \end{aligned} \tag{3.1}$$

where x_s^i and y_s^i are x and y coordinates of the source s_i and y_t^i and y_t^i are x and y coordinates of sink t_i.

Proof. Suppose, by way of contradiction, that there exists an instance I of Problem RCN such that each minimum rectilinear coding network for I contains at least one vertical segment $e = ((x', y'), (x', y''))$ such that $x' \notin \{x_s^i \mid s_i \in S\} \cup \{x_t^j \mid t_j \in T\}$ or a horizontal segment $e' = ((x', y'), (x'', y'))$ such that $y' \notin \{y_s^i \mid s_i \in S\} \cup \{y_t^j \mid t_j \in T\}$.

For each minimum rectilinear coding network A we associate values $\theta_x(A)$ and

$\theta_y(A)$, defined as follows:

1. $\theta_x(A)$ is equal to the sum of all x coordinates of the vertical segments of A;
2. $\theta_y(A)$ is equal to the sum of all y coordinates of the horizontal segments of A.

Let $\hat{A}$ be a minimum rectilinear coding network for instance I that has a minimum value of $\theta_x(\hat{A}) + \theta_y(\hat{A})$. Suppose that $\hat{A}$ contains a vertical segment $e = ((x', y'), (x', y''))$ such that $x' \notin \{x_s^i \mid s_i \in S\} \cup \{x_t^j \mid t_j \in T\}$. Note that e is not incident to the source or a sink and it does not have adjacent vertical segments. We denote by n_l and n_r the number of horizontal segments incident to e from the left and right, respectively.

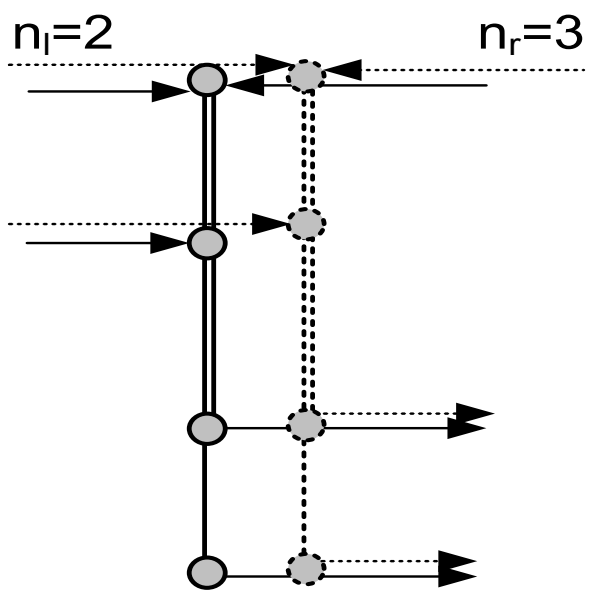


Fig. 7. A example of two collocated vertical segments with $n_l = 2$ and $n_r = 3$. Sliding both segments to the right (shown by horizontal lines) decreases the total wirelength.

First, we assume that e is not collocated with any other vertical segment of $\hat{A}$. We consider the following three cases:

- Case I: $n_l > n_r$. In this case we can "slide" segment e to the left, resulting is a coding network whose wirelength is less than that of $\hat{A}$, which contradicts the optimality of $\hat{A}$.
- Case II: $n_l < n_r$. In this case we can "slide" segment e to the right, resulting is a coding network whose wirelength is less than that of $\hat{A}$, which contradicts the optimality of $\hat{A}$.
- Case III: $n_l = n_r$. In this case we can "slide" segment e to the left, without decreasing the total wire-length of A. We note that the resulting rectilinear coding network satisfies the following conditions: (i) $\theta_y(A') \leq \theta_y(\hat{A})$ (ii) $\theta_x(A') = \theta_x(\hat{A})$. This, however contradicts the fact that $\hat{A}$ has a minimum value of $\theta_x(\hat{A})+\theta_y(\hat{A})$.

The case of multiple collocated vertical segments can be dealt with by using similar arguments. In this case we define by n_l and n_r the total number of horizontal segments incident to at least of the co-located vertical segments from the left and right, respectively and "sliding" all co-located segments together, as shown in Figure 7.

Using similar arguments, we can also show that $\hat{A}$ does not contain a horizontal segment $e' = ((x', y'), (x'', y'))$ such that $y' \notin \{y_s^i \mid s_i \in S\} \cup \{y_t^j \mid t_j \in T\}$. □

C. An Integer Problem Formulation

In this section we present integer programs that allow us to find both the optimal coding network for for Problem RCN and the optimal Steiner tree packing topology.

1. Finding Optimal Coding Network

In this section we present an integer program that allows us to find an optimal coding network for for Problem RCN. In the previous section, we showed that it is sufficient to restrict our search to the graph imposed by Hanan points. A *Hanan grid* is a graph

that includes all horizontal and vertical segments that connect adjacent Hanan points. The Hanan grid has a relatively small size, which allows us to use computationally intensive tools, such as IP solvers.

Let G be a Hanan grid that corresponds to the problem instance. Let $\overrightarrow{G}$ be the graph formed from G by replacing each vertical and horizontal segment $e \in G$ by two bi-directed arcs a_e^1 and a_e^2 that connect the end points of e.

The main theorem of network coding implies that the coding network contains, for each sink $t_i \in T$, h disjoint paths between from all source nodes $s_i, 1 \leq i \leq h$ and t_i. We denote the flow on h disjoint paths between the sources and sink t_i by F_i. For each Hanan point v we define $N_i(v)$ to be amount of flow F_i generated by this point. The value $N_i(v)$, $1 \leq i \leq k$ is defined as follows. Let N_v the number of source nodes $s_i, 1 \leq i \leq h$ located at point v. Then, if the sink t_i is not located at v then $N_i(v) = N_v$. Otherwise, $N_i(v) = N_v - k$.

We define the following variables:

- For each arc $a \in \overrightarrow{G}$: c_a- the capacity of edge a;
- For each arc $a \in \overrightarrow{G}$ and for each sink $t_i \in T$: f_a^i- the amount of flow of type i that goes through arc a;

The integer program is then defined as follows:

Minimize $\sum_{a \in \overrightarrow{G}} c_a$

Subject to:

- For each $v_i \in \overrightarrow{G}$ and for each $t_m, 1 \leq m \leq k$:

$$\sum_{v_j:(v_i,v_j) \in \overrightarrow{G}} f^m_{(v_i,v_j)} - \sum_{v_j:(v_j,v_i) \in \overrightarrow{G}} f^m_{(v_j,v_i)} = v^m(v_i)$$

– For each $(v_i, v_j) \in \overrightarrow{G}$ and for each $t_m, 1 \leq m \leq k$:

$$f^m(v_i, v_j) \leq c_{(v_i,v_j)}$$

$$f^m(v_i, v_j) \in \{0, 1, \ldots, h\}$$

– For each $(v_i, v_j) \in \overrightarrow{G}$:

$$c_{(v_i,v_j)} \in \{0, 1, \ldots, h\}$$

The first constraint enforces the flow conservation at each node $v_i \in \overrightarrow{G}$. The second constraint ensures that the capacity of each edge is more or equal to the amount of flow to each sink. Finally, all other constrains ensure the integrality of all variables.

2. Finding Optimal Steiner Tree Packing

Let G be a Hanan grid that corresponds to the problem instance. Let $\overrightarrow{G}$ be the graph formed from G by replacing each edge $e \in G$ by two arcs a_e^1 and a_e^2. For each source $s \in S$ there exists a corresponding Steiner tree that connects s and the sinks $t_1, \ldots, t_k$.

We define the following variables:

- For each arc $a \in \overrightarrow{G}$: c_a- the amount of bandwidth reserved on a;
- For each arc $a \in \overrightarrow{G}$ and for each source $s \in S$: f_a^s the reservation on arc a for the tree that originates at s;
- For each arc $a \in \overrightarrow{G}$, for each source $s \in S$, and for each destination $t \in$: $f_a^{s,t}$ the reservation on arc a for the path that between s and t (that uses the edges

that belong to the tree that originates at s).

The integer program is defined as follows:

Minimize $\sum_{a \in \overrightarrow{G}} c_a$

Subject to

- For for each $s \in S$, $t \in T$:

$$\sum_{v_j:(s,v_j)\in\overrightarrow{G}} f^{s,t}_{(s,v_j)} - \sum_{v_j:(v_j,s)\in\overrightarrow{G}} f^{s,t}_{(v_j,s)} = 1;$$

$$\sum_{v_j:(v_j,t)\in\overrightarrow{G}} f^{s,t}_{(v_j,t)} - \sum_{v_j:(s,v_j)\in\overrightarrow{G}} f^{s,t}_{(s,v_j)} = 1;$$

- For for each $s \in S$, $t \in T$, and $v_i \in \overrightarrow{G}$, $v_i \neq s$,$v_i \neq t$:

$$\sum_{v_j:(v_i,v_j)\in\overrightarrow{G}} f^{s,t}_{(v_i,v_j)} = \sum_{v_j:(v_j,v_i)\in\overrightarrow{G}} f^{s,t}_{(v_j,v_i)}$$

- For each $s \in S$, $t \in T$, and for each $(v_i, v_j) \in \overrightarrow{G}$:

$$f^{s,t}_{(v_i,v_j)} \leq f^{s}_{(v_i,v_j)}$$

- For each $(v_i, v_j) \in \overrightarrow{G}$:

$$\sum_{s\in S}^{k} f^{s}_{(v_i,v_j)} \leq c_{(v_i,v_j)}$$

- For each $(v_i, v_j) \in \overrightarrow{G}$ and for each $s \in S$

$$f^{s}(v_i, v_j) \in \{0, 1\}$$

D. Coding Advantage

As discussed below, the network coding technique can result in a significant reduction in the total wirelength. In this section we evaluate the value of coding advantage in random settings.

Definition 3. *Let I be an instance of problem Problem RCN, let OPT the cost of the optimal solution to I with network coding and let OPT^* be the cost of the solution I using the standard methods (Steiner tree packing). Then, the coding advantage C is defined as follows:*

$$C = \frac{OPT^* - OPT}{OPT}. \tag{3.2}$$

To evaluate the coding advantage of the network coding in a two-dimensional plane we conducted a series of experiments. In all experiments, the location of source and destination nodes were chosen at random. The cost of an optimum solution with network coding was computed by solving the the integer program presented in Section C. This cost was compared to the total cost of optimal Steiner trees for each net. We used ILOG CPLEX [29] for finding solving integer program and the GeoSteiner software [30] for finding optimal rectilinear Steiner trees.

Specifically, we created 200 random instances, each instance has 2-6 nets, each net includes 6-20 sinks. Figure 8 shows the values of coding advantage obtained in different experiments. Our results show that the network coding has an advantage in 19.7% of the cases. Figure 8 shows the histogram of coding advantage for the cases in which the network coding results in a reduction in the total wirelength. The average coding advantage for such instances is 0.18%.The maximum gain is 2.43%.

Figure 9 shows an example of routing on an unconstrained plane. This example has 2 sources 6 sinks. The coding node is marked and the dashed lines indicate the path of the encoded signal. The thick lines denote two co-located segments. In

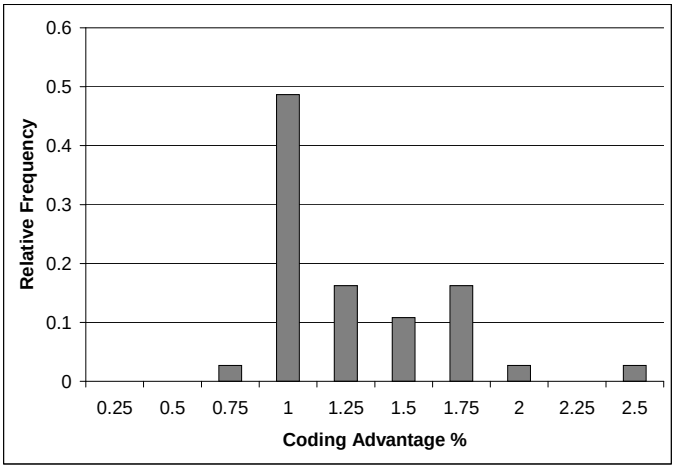


Fig. 8. Average wirelength reduction for unrestricted two-dimensional plane

this example, network coding based routing offers a gain of 2.22% over Steiner tree packing employed over the same arrangement of nodes.

We observed that the network coding technique is most beneficial in the cases in which the instance of Problem RCN is similar to that of depicted in Figure 4. In order to confirm our observation we conducted an experiment with two nets whose sources s_1 and s_2 are located on horizontal line with a sink node t_1 located in the middle. We selected two other sinks, t_2 and t_3 in a random manner. We conducted a 216 experiments which out of which 14.5% showed a non-zero advantage of network coding. Figure 10 shows the histogram of coding advantage for the cases in which the network coding results in a reduction in the total wirelength. The average coding advantage in such instances is 4.9%, with some instances having the advantage as high as 12%.

We conclude that in general the network coding can result in a reduction of total wirelength for a significant portion of the random instances. The instances which

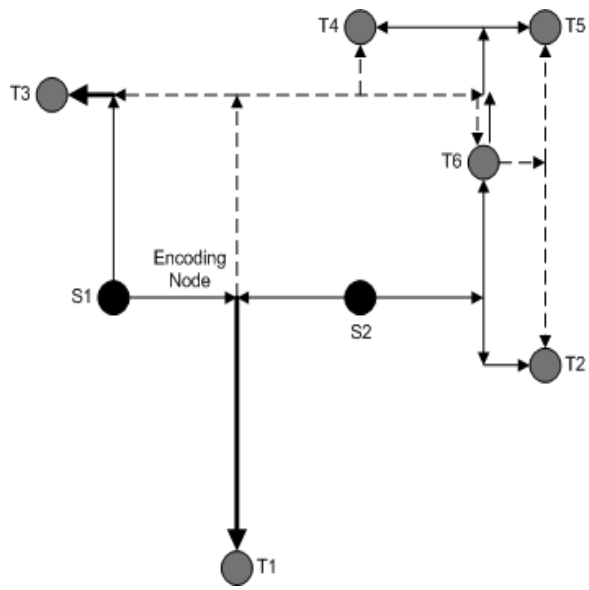


Fig. 9. Example of routing over an unconstrained plane

are most likely to benefit from the network coding are those similar to the instance presented in Figure 4.

E. Heuristic Solutions

The optimum algorithm, described above, has a satisfactory running time for instances with a small number of nets that share a limited number of sinks. However, in the case of a large number of nets with many common sinks, the computational complexity of the optimal algorithm is prohibitively large. Accordingly, in this section we present a heuristic solution which achieves a near-optimal performance and whose running time is much smaller than that of the optimum solution.

Our heuristic works for the instances I of Problem RCN which includes two nets, n_1 and n_2. We denote by s_i the source node of net $i, i \in \{1, 2\}$, and by x_{s_i} and y_{s_i} the x and y coordinate of s_i, respectively. Let R be a region that include all points

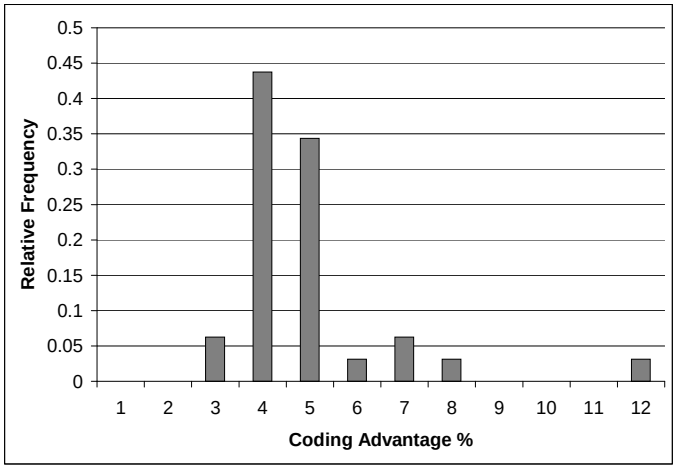


Fig. 10. Average wirelength reduction in special cases

(x, y) for which it holds that

$$\begin{aligned} \min\{x_{s_1}, x_{s_2}\} \leq x \leq \max\{x_{s_1}, x_{s_2}\} \\ \min\{y_{s_1}, y_{s_2}\} \leq y \leq \max\{y_{s_1}, y_{s_2}\} \end{aligned} \tag{3.3}$$

We also denote by $\hat{s}$ the point that located in the middle of the line that connects s_1 and s_2, i.e., the point with x coordinate $\frac{x_{s_1}+x_{s_2}}{2}$ and y coordinate $\frac{y_{s_1}+y_{s_2}}{2}$.

Our heuristic is motivated by the observation that most of coding networks that have an advantage over the standard solutions are similar to the network depicted in Figure 4. Such coding networks have a single encoding node v. The set of sinks T of such networks can be decomposed into two subsets, T_1 and T_2, while the set of edges can be decomposed into three Steiner trees, Y_1, Y_2, and Y_3, such that Y_1 connects s_1, v, and T_1, the second tree connects s_2, v, and T_2, where the third tree, Y_3 connects v, T_1, and T_2. The search for a node v can be performed efficiently by observing that v can only be located at a Steiner point, and likely to be located between s_1 and s_2.

Specifically, our heuristic algorithms includes the following steps:

1. Construct the Hanan grid $G(V, E)$ of the instance I of Problem RCN.

2. Find a minimum cost rectilinear Steiner tree $\hat{Y}_1$ that connects s_1 and T.

3. Find a minimum cost rectilinear Steiner tree $\hat{Y}_2$ that connects s_2 and T.

4. For each Hanan point point v of G which lies within the region R do:

   (a) Find a near optimal coding network with the coding point located in v through the following steps:

      i. Divide the sinks in $T \setminus \{v\}$ into two groups T_1 and T_2, such that T_1 includes the sinks that are closer to s_1 than to s_2 and T_2 includes the sinks that are closer to s_2 than to s_1.[1].

      ii. Find a Rectilinear Minimum Steiner tree Y_1 that connects nodes in T_1, s_1, and v.

      iii. Find a Rectilinear Minimum Steiner tree Y_2 that connects nodes in T_2, s_2, and v.

      iv. Find a Rectilinear Minimum Steiner tree Y_3 that connects nodes in $T_1 \cup T_2$ and v.

5. If region R is empty, we select a Steiner point which is closest to $\hat{s}$

6. Select the point v that minimizes the total wirelength of trees Y_1, Y_2, and Y_3.

7. If the total cost of Y_1, Y_2, and Y_3 is less than that of $\hat{Y}_1$ and $\hat{Y}_2$, output Y_1, Y_2, and Y_3; otherwise output $\hat{Y}_1$ and $\hat{Y}_2$.

As shown in Figure 11, node $T1$ is chosen as the encoding node since it lies between the two sources $S1$ and $S2$. Steiner tree 1 connects source $S1$ with the

[1]We consider Manhattan distances

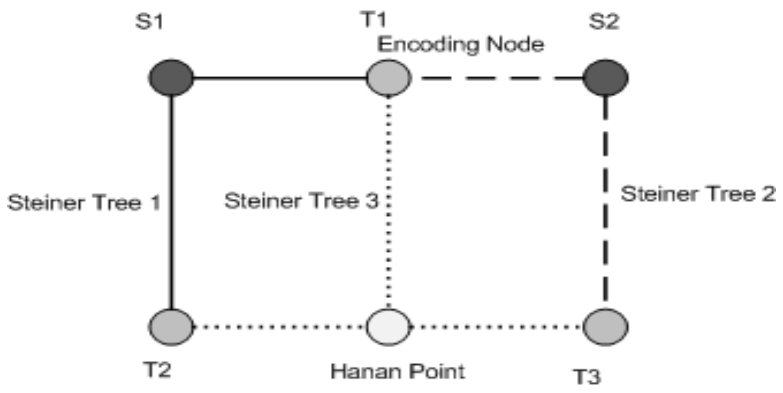


Fig. 11. Illustration of heuristic solution for problem RCN.

encoding node and sink $T2$ which is near $S1$. Steiner tree 2 connects source $S2$ with the encoding node and sink $T3$ which is near $S2$. Steiner tree 3 connects the encoding node with all sinks. The three Steiner trees together form the network coding topology required to interconnect the sources and sinks.

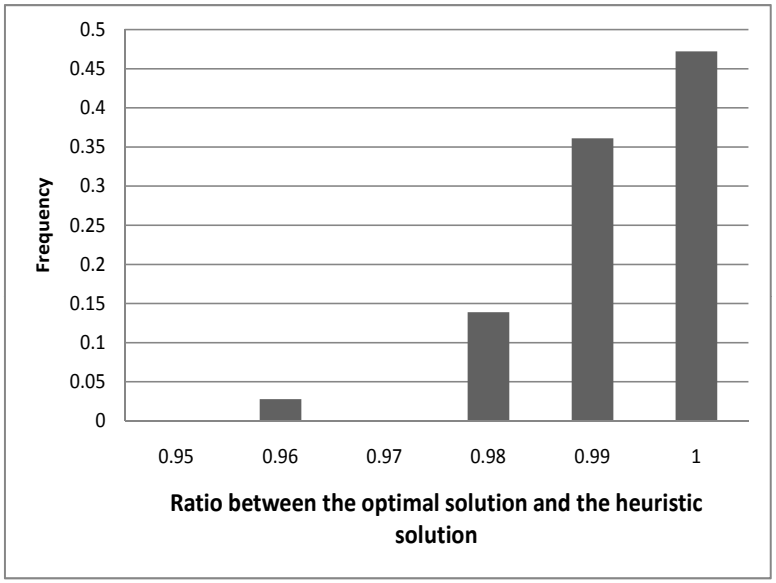


Fig. 12. The performance of the heuristic for problem RCN.

We can use any heuristic or optimal solution for the finding minimum cost or near-minimum cost rectilinear Steiner trees. Figure 12 shows the performance of the heuristic for the cases in which the network coding has an advantage over the standard solution. The average ratio between the optimal wirelength and that of heuristic solution is 99%.

CHAPTER IV

MULTI-NET ROUTING ON CONSTRAINED AND CONGESTED TOPOLOGIES

In the previous chapter, we compared the performance of network coding based routing algorithm with the traditional Steiner tree packing approach on the unrestricted rectilinear plane. However, in the practical scenarios involving routing on a chip, we will frequently come across topologies which are constrained or congested. In this section, we discuss how network coding can be applied to such topologies and compare the results of simulations with those obtained from traditional approaches.

A. Routing in the Presence of Hard Blockages

It has been shown in [25] that the Hanan grid can be extended to accommodate hard blockages. The modified grid is referred to as the modified Hanan grid, or *escape* graph. An example of an escape graph is shown in Figure 13.

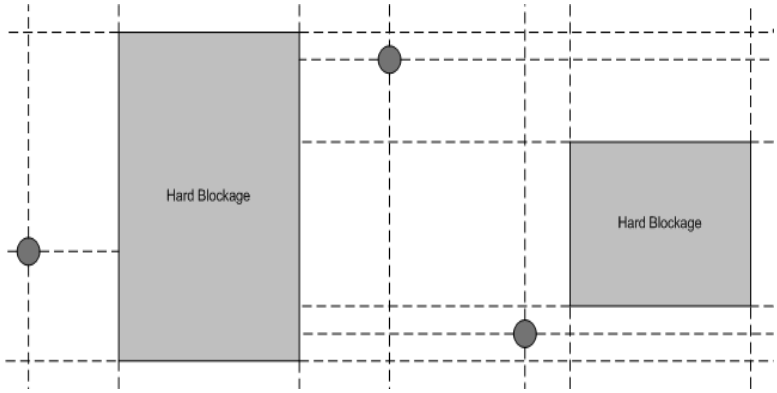


Fig. 13. Example of escape graph

The escape graph includes some extra edges which correspond to the boundaries

of the blockages in the plane. It can be shown, using the arguments similar to that used in Section B that the escape graph contains an optimal coding network. That is, it is sufficient to restrict the search for optimal coding networks to the escape graphs.

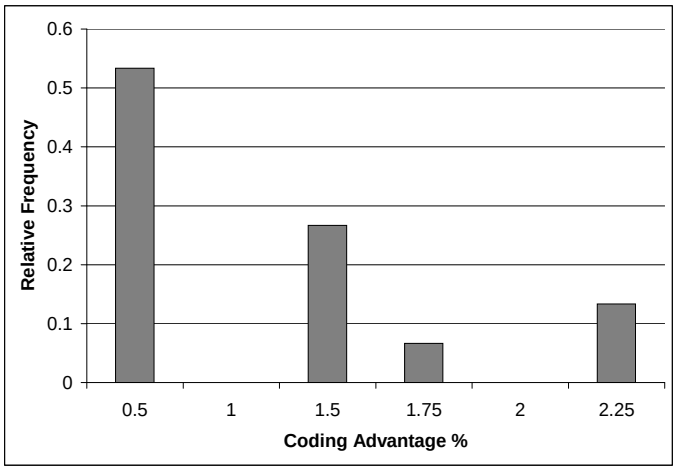


Fig. 14. Wirelength reduction for hard blockages

Figure 14 shows the histogram of coding advantage for 175 experiments with hard blockages. In these experiments we selected a random number of source and sink nodes, which are placed in several locations, which are also selected at random. All obstacles have rectilinear shapes, their size and position are are also randomly selected. We observed a positive coding advantage in 46.1% of the cases, which indicate that a positive coding advantage is more likely to occur in the presence of hard blockages. The average coding advantage in such cases is 0.866%.

Figure 15 shows an example in which network coding has been applied to carry out routing in the presence of two hard blockages on the grid. As can be seen, the signals from the two sources enter the encoding node which computes their linear combination and transmits it along the path marked by the dashed line. The topology

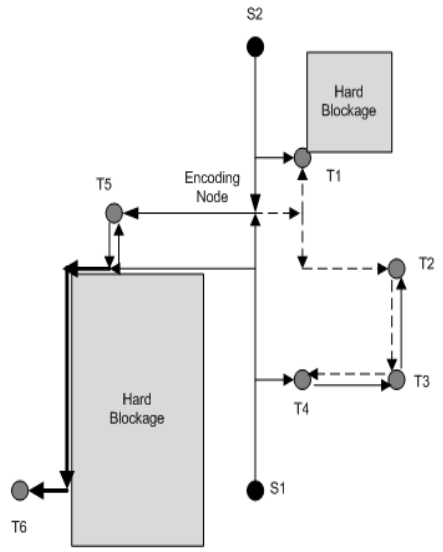


Fig. 15. Network coding in the presence of hard blockage

is similar to the butterfly network. The thick lines are paths of width two where both signals travel without coding. This topology has a coding advantage of 2.22%.

B. Routing in the Presence of Soft Blockages

In this case, rectilinear network coding is carried out on a plane which contains soft blockages. Soft blockages are those through which a signal can pass, but the capacity of edges that run through a soft blockage is limited. Thus, only limited number of interconnects can run can be done through a soft blockage. For example, a congested region can be treated as a soft blockage. In this case, routing is done in a rectilinear fashion using the entire plane. We refer to this grid as the regular grid.

The experimental setup for the simulations is described as follows. The number of sources and sinks and their locations on the plane are randomly chosen. Soft blockages are simulated by randomly choosing the available capacity on the edges of the regular grid subject to a maximum of the number of signals to be sent. The

optimal topology such that each source connects to each sink is then determined along the regular grid. The cost metric used in this case is the length of the routing path. Steiner tree packing is done via a integer program formulation. The solution of this program gives the cost of the optimum path using standard methods.

In the case of soft blockages, the network is modeled as a rectilinear grid. Each segment e in the grid is associated with the cost c_e of using this segment in the routing topology. The cost depends on the particular situation with the congestion in local area of e, with more severe congestion result in higher cost values. Unfortunately, with soft blockages, the Hanan's theorem is no longer applicable, so we need to search the entire grid, resulting is a relatively large solution space.

We run the experiment to evaluate the coding advantage in this setting. We used the integer program discussed in Section C to find the optimal solution using both the Standard approach and the network coding approach.

The values of the coding advantage are presented in Figure 16 which shows the results for 210 simulations. These were run over a 10 x 10 grid. Up to 10 nodes were randomly chosen in each simulation with random choice of sources and sinks among the chosen nodes. In 16.9% of the simulations there is a gain when network coding is used for routing instead of the standard routing techniques. The maximum gain in these 200 simulations is 3.44% with an average gain of 0.258%.

Soft blockages on a grid cause some edges to get constrained while some may not even be constrained. However, if every edge on the grid is constrained, the gain obtained is higher than with soft-blockages. This is evident from the results in the following section.

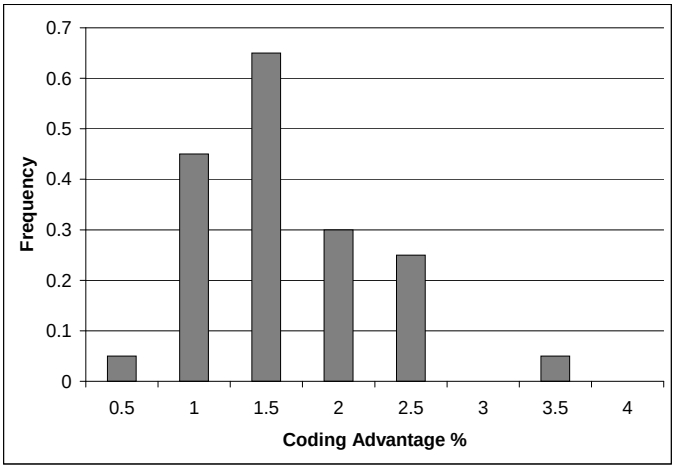


Fig. 16. Wirelength reduction for soft blockages

C. Routing on a Congested Grid

In this section we evaluate the benefits of network coding in highly congested topologies. We run our experiments on a rectilinear grid with randomly selected source and destination nodes. Our goal is to minimize the total wirelength, as the longer wires cause congestion in more areas.

This is a special case of soft blockages where each edge of the regular grid has only unit capacity. This simulates a network which is congested and constrained for capacity. The cost metric is again the length of the path.

Figure 17 shows the coding advantage of network coding over the traditional approaches over a constrained grid. The results are from 200 simulations. Each simulation was run over a 10 x 10 grid with up to 15 nodes. The number and location of the nodes was randomly chosen and so was the number and location of the sources among these nodes. As can be seen, in 13% of the cases, network coding gave an advantage over the traditional approaches. The maximum gain observed in these

simulations was 4.34% while the average gain across the instances that yield positive gain is 0.38%.

The coding advantage is higher for routing instances with 2 sources, both of which are in horizontal or vertical alignment and there is a sink between them. An experiment was carried out to evaluate this situation and the results are shown in Figure 17.

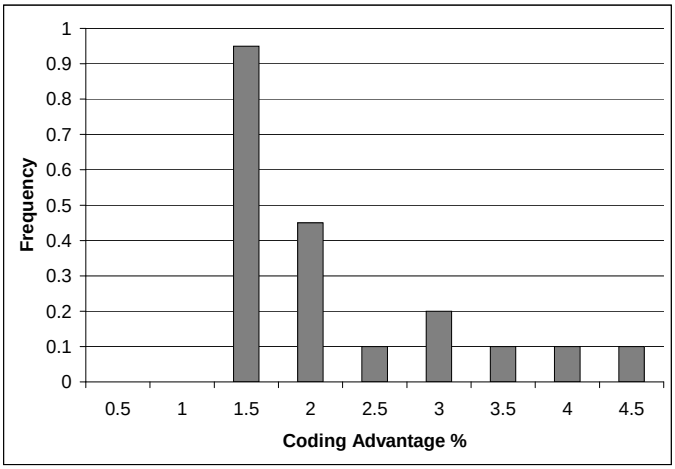


Fig. 17. Wirelength reduction for constrained grid

D. Routing to Reduce Density

This case simulates the situation in a congested network where some capacity of the edges may already be used and the routing needs to be done using only the available capacity. For each edge, the density of the edge is the ratio of the number of wires routed through this edge to the total capacity of the edge. Thus, a higher density means that a limited residual capacity is available for further use. It is desirable to have an even density distribution throughout the grid. This is achieved

by encouraging usage of low density edges by giving them lower costs and assigning higher costs to edges with higher density. Thus, in this case, our goal is to route the signals such that the density-based cost of routing is the minimum.

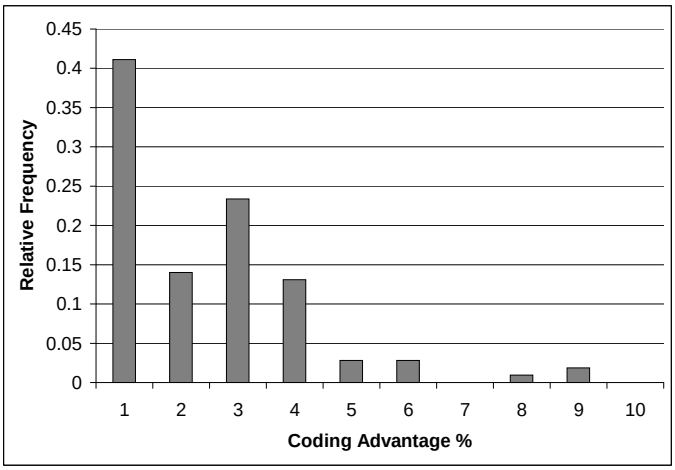


Fig. 18. Density reduction in constrained topologies

Figure 18 shows the histogram of coding advantage for 200 experiments. We considered a grid routing topology in which the density of each edge is chosen at random. The cost of a routing topology (a coding network or a Steiner tree) is estimated by the sum of the costs of its edges, where the cost of each edge is equal to its density. The locations of the source and destination nodes were chosen at random as well. We observed a positive coding advantage in 53% of the cases. In such cases, the average coding advantage is 1.01%. The coding advantage in some cases can be as high as 8.4%.

E. Heuristic Solution

The optimum algorithm, for all the cases discussed in this chapter, has a satisfactory running time only for instances on a small grid size and which share a limited number of sinks. However, if the size of the routing grid and the sink is very large, the computational complexity of the optimal algorithm becomes prohibitively large. Accordingly, in this section we present a heuristic solution which achieves a near-optimal solution and whose running time is much smaller than that of the optimum solution.

The heuristic solution which we propose to use on a constrained grid is based on the main theorem of network coding. For every terminal, it finds disjoint paths of minimum cost from each source to that terminal, starting with that terminal which has lowest total distance from both sources. While constructing the minimum cost disjoint paths, it encourages sharing of links already used for connecting previous terminals.

The heuristic solution includes the following steps :

Given an undirected network $G(V, E)$ where each edge $e \in E$ has a (residual) capacity and cost that has two source nodes s_1 and s_2 and a set of destination nodes $T = t_1, t_k$,

1. Let d_t^i be the distance between terminal t and source i and let $d_t = d_t^1 + d_t^2$
2. Let $A \longrightarrow \phi$
3. For each terminal $t \in T$ in order of d_t (beginning with the smallest) do:
   (a) Find two disjoint paths P_1 and P_2, such that P_1 is a path between s_1 and t, and P_2 is a path between s_2 and t, of minimum cost.
   (b) For each edge in P_1 and P_2 select an orientation according to the direction of the edge in P_1 or P_2. Assign the cost ϵ_i to the resulting directed arcs,

where ϵ_i is a small positive number and i is the number of the current iteration.

(c) Add the directed arcs that correspond to edges of P_1 and P_2 to A.

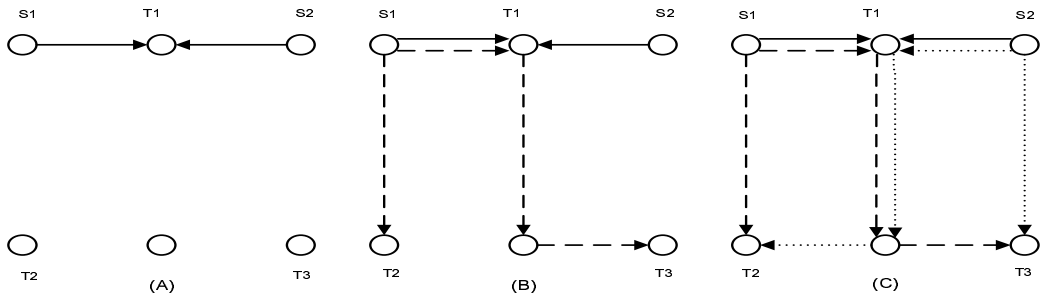


Fig. 19. Heuristic solution for constrained grid

As shown in Figure 19(A), since $T1$ is the terminal closest to both sources, disjoint paths are constructed along the initially undirected graph from each sources to node $T1$. The orientation of edges used along these disjoint is fixed and they are assigned a very small positive cost. Next, as seen in Figure 19(B), disjoint paths are constructed from both sources to $T2$. Edges with two arrows indicate that the edge is used in two different flows. However, the edge is used and paid for only once. Finally, as seen in Figure 19(C), the third pair of disjoint paths are constructed from the two sources to terminal $T3$. Again, some edges previously used are preferred and used in this pair. The edges used after all the pairs of disjoint paths have been determined constitute the network coding topology for interconnecting sources and sinks on the constrained grid.

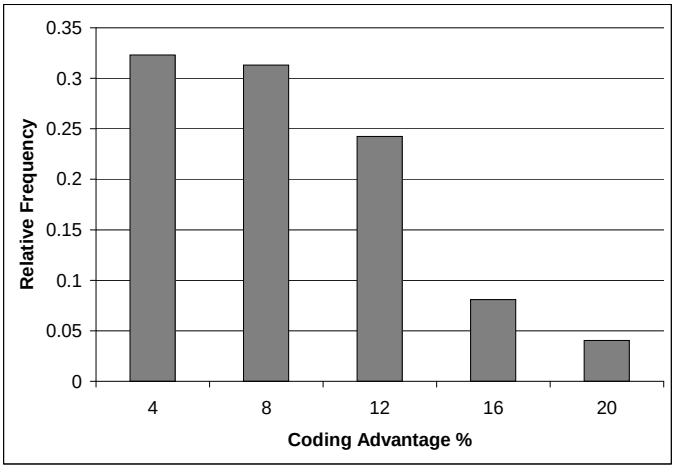


Fig. 20. Comparison of heuristic solution with Steiner tree packing

To evaluate the performance of this heuristic, we compare its performance with the Steiner tree heuristic. We considered a grid routing topology in which the capacity available on each edge is 1. Thus, this is the case of the constrained network. The cost of a routing topology (a coding network or a Steiner tree) is estimated by the sum of the costs of using the edges, where the cost of each edge is equal to its wirelength. The number of nodes was randomly selected. Each case always had two sources. The locations of the source and destination nodes were chosen at random on a 10x10 grid. The Steiner tree heuristic employed was as follows.

1. Initially, the nodes are connected by undirected edges of the grid;

2. Find the shortest distance of each sink from the source;

3. Sort the sinks in ascending order of their distance from the source;

4. For each sink considered in the sorted order, find the shortest path to it from the source. After each sink is processed, set the orientation of the edges depending

on how it is used and the cost of edges utilized along its path to a very small positive value to encourage its use for other sinks;

5. Add the edges used along each path to a tree T which is the required Steiner tree.

6. Update costs of all edges before the next Steiner tree is constructed.

The above heuristic is used to find a Steiner tree connecting all the sinks to each source. The total cost using the Steiner tree heuristic is thus the total routign cost of the two Steiner trees.

Figure 20 shows the histogram of coding advantage for 200 experiments. We observed a positive coding advantage in 50% of the cases. In such cases, the average coding advantage is 3.45%. The coding advantage in some cases can be as high as 20%.

CHAPTER V

PERFORMANCE OF NETWORK CODING IN ISPD BENCHMARKS

In order to evaluate the advantage of the network coding methods in the practical settings, we applied our techniques on the instances generated by the ISPD benchmarks. The network coding was used for reducing the density of the interconnect networks, using rip-up-and-reroute methods.

In the rip-up-and-reroute methods, each net in the netlist is individually routed without considering congestion by constructing a Steiner minimum tree. After all nets are routed, the congestion on the chip is analyzed and nets passing through highly congested regions are ripped out and rerouted through areas with lesser congestion. This method is commonly used in global routing.

Specifically, we used the benchmark `ibm01` of the ISPD98 benchmarks. We first applied the BoxRouter [8] over the benchmark for $K = 15$, and calculated the resulting density map. Density is defined as the ratio of number of times a boundary is crossed to the capacity of the boundary. We also computed the overflow for each edge, which is defined as the number of times the boundary is crossed in excess of the capacity of the boundary. The cost c_e of each edge $e \in E$ is defined to be the square of the sum of density and overflow:

$$c_e = (\frac{p_e}{g_e} + overflow)^2, \tag{5.1}$$

where p_e is the number of wires that uses the edge, g_e is the total capacity of an edge and $overflow$ denotes the number of wires routed on an edge in excess of its capacity. The goal of this cost function is to avoid edges of high density and with an overflow.

Next, we find pairs of nets which have at least 4 pins in common. These are pairs of nets which may offer a coding opportunity. We found 1803 such pairs in the `ibm01`

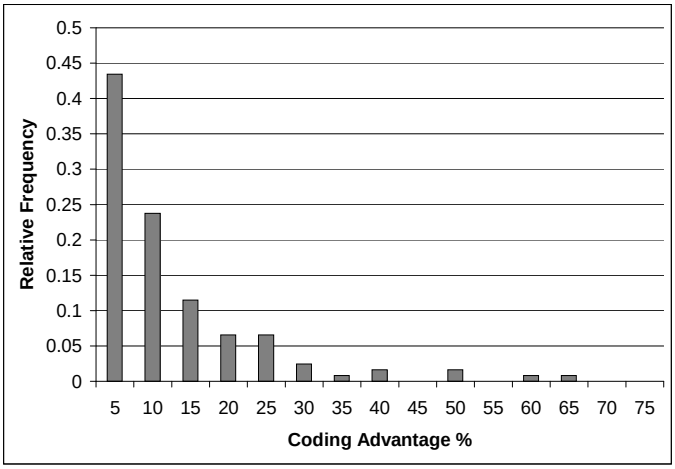


Fig. 21. Local density reduction in ISPD benchmark ibm01

benchmark. For every pair of such nets, we compare the cost of using the network coding to the cost the the routing used by BoxRouter. We used the heuristic solution presented in Section E to find an optimal coding network.

Our experiments show that the network coding technique can be successfully incorporated in the rip-up-and-reroute algorithms. Specifically, the network coding technique was beneficial in 7.27% of the cases in which we had a coding opportunity, where the coding opportunity is the case in which we have two nets with a large number of shared sinks. Figure 21 shows the histogram of local density improvement for the cases in which the network coding has an advantage over the standard methods. The local density improvement is defined by the ratio of the cost of the replaced Steiner trees to the cost of network coding topology that was added to substitute them. The average local density reduction is 0.712%, but in some cases, the reduction can be as high as 62.31%.

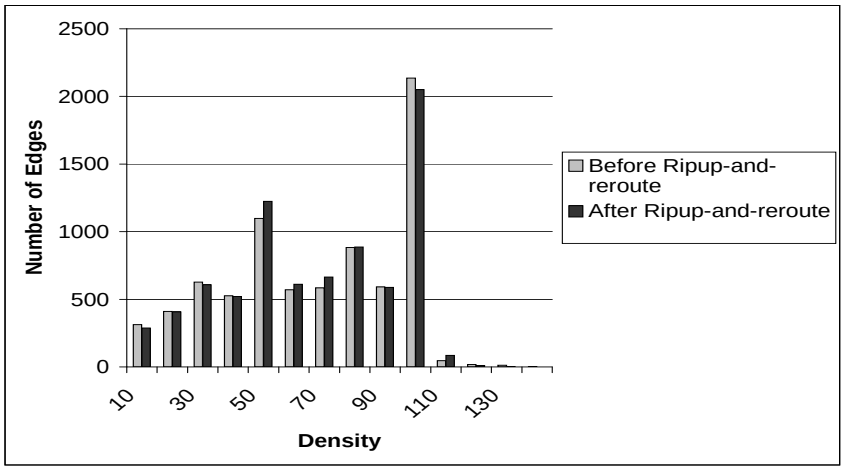


Fig. 22. Histogram of the density of edges before and after the process for ISPD benchmark ibm01

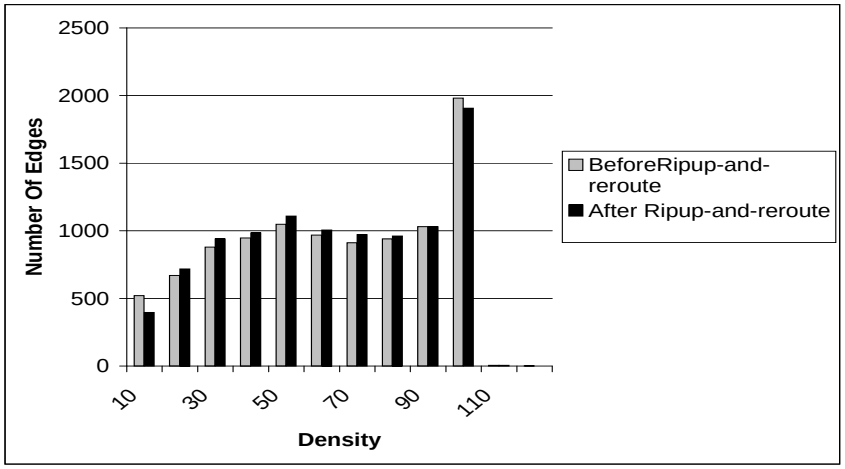


Fig. 23. Histogram of the density of edges before and after the process for ISPD benchmark ibm03

Table I. Performance evaluation of heuristic on ISPD benchmark ibm01

Parameter	Value Before Network Coding	Value After Network Coding	Percentage Saving
Total Density	43570768	42179426	3.19%
Maximum Density	133	125	6.01%
Total Overflow	126	115	8.73%
Maximum Overflow	4	3	25%
Overflow Count	78	98	-25.64%

Table II. Performance evaluation of heuristic on ISPD benchmark ibm03

Parameter	Value Before Network Coding	Value After Network Coding	Percentage Saving
Total Density	44525085	44259506	0.59%
Maximum Density	110	105	4.54%
Total Overflow	9	6	33.33%
Maximum Overflow	2	1	50%
Overflow Count	7	6	14.28%

Figures 22 and 23 plot the histograms of densities before and after rip-up-and-reroute has been performed for the `ibm01` and `ibm03` benchmarks respectively. As seen in the figure, network coding reduces the number of edges in the high density region of the histogram by utilizing edges having lower or medium densities. This explains the increase in the count of edges in the medium density region. At the same time, it also reduces the number of edges with high overflows by using other edges either with lower overflows or no overflows. Thus, there is a slight increase in the number of edges with lower overflow values, but the count of edges with high overflows and also the total overflow of the design have been reduced.

Table I compares the parameters of the routing benchmark before and after rip-up-and-reroute has been carried out on the benchmark `ibm01`. It can be seen that the process helps in reducing the total density, the maximum density, the maximum overflow and the total overflow on the chip. Thus, the congestion across the chip is reduced.

Table II compares the same parameters for benchmark `ibm03`. Again, there is a significant reduction in total cost, maximum density, total and maximum overflows and overflow count.

CHAPTER VI

CONCLUSION AND FUTURE WORK

A. Conclusion

This thesis investigates the applications of network coding for improving the routability, density, overflow and wirelength of VLSI designs. We have performed a detailed analysis of cases including routing over a Hanan grid, routing in the presence of soft blockages, routing in the presence of hard blockages and routing over constrained networks. We presented both the optimal solutions based on integer programming and heuristic solutions.

We have performed an extensive simulation study for the above mentioned cases and compared the performance of the network coding technique with that of traditional routing approaches. In the random cases, on the unconstrained rectilinear grid, network coding was especially beneficial in topologies which were roughly similar to the rectilinear butterfly network as shown in Figure 4. The same held true even in the presence of hard blockages where routing had to be carried out along an escape graph. On the other hand, on congested graphs having either soft blockages or constrained edges, network coding was seen to be useful in more general cases and also yielded higher values of gain. Besides, network coding also offered significant gain in random cases where the aim was to reduce the density of edges. Thus, our results indicate that though network coding does provide improvement in all cases, it is most beneficial in cases involving congested and constrained topologies.

We also evaluated the performance of the network coding heuristic on the ISPD98 routing benchmarks and observed that network coding not only helped in improving the density map of the entire design, but also reduced the maximum density, maxi-

mum overflow and the total overflow of the design.

Thus, it can be seen that network coding proves to have significant potential to reduce congestion and density of VLSI designs.

B. Future Work

This thesis mainly focuses on VLSI design styles such as standard cell, sea of gates or gate array where the cell arrangement is regular giving rise to a rectilinear grid. It would be interesting to investigate how this technique performs in the custom design technique where the cell arrangement is not regular and thus the routing is along an irregular rectilinear grid.

In addition, this thesis restricts itself to the study of routing for two dimensional grids. However, in multi-layer VLSI designs routing may take place in three dimensions and thus, the performance of the network coding technique needs to be evaluated in this case.

Since network coding yields good results when applied in the rip-up-and-reroute process, it would be interesting to incorporate this technique into a router and study its performance therein. Moreover, since network coding does yield some improvement in the case of routing on an unconstrained grid, it may also be employed in a router to lay out the initial routing trees over which rip-up-and-reroute is performed.

Thus, it can be seen that the research involving application of network coding for VLSI routing is still in its infancy and there is much more scope for studying, applying and further enhancing it.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[2] Z. Li and B. Li, "Network Coding in Undirected Networks," in *Proceedings of the 38th Annual Conference on Information Sciences and Systems (CISS)*, pp. 257–262, 2004.

[3] N. Jayakumar, K. Gulati, S. Khatri, and A. Sprintson, "Network Coding for Routability Improvement in VLSI," in *Proceedings of of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 820–823, November 2006.

[4] J. Hu and S. Sapatnekar, "A Survey on Multi-net Global Routing for integrated Circuits," *Integration: the VLSI Journal*, vol. 31, pp. 1–49, Nov. 2001.

[5] G. Robins and A. Zelikovsky, "Improved Steiner tree approximation in graphs," in *Symposium on Discrete Algorithms*, pp. 770–779, 2000.

[6] S. Areibi, M. Xie and A. Vannelli, "An efficient rectilinear Steiner tree algorithm for VLSI Global Routing," in *Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 1067–1072, May 2001.

[7] H. Sun and S. Areibi, "Global routing for VLSI standard cells," in *Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 485–488, May 2004.

[8] M. Cho and D. Pan, "BoxRouter: A New Global Router Based on Box Expansion and Progressive ILP," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 373–378, 2006.

[9] M. Pan and C. Chu, "FastRoute: a Step to Integrate Global Routing into Placement," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 464–471, Nov. 2006.

[10] R. Hadsell and P. Madden, "Improved Global Routing through Congestion Estimation," in *Proceedings of the Design Automation Conference*, pp. 28 - 31, June 2003.

[11] C. Albrecht, "Provably Good Global Routing by a New Approximation Algorithm for Multicommodity Flow," in *Proceedings of the ACM International Symposium on Physical Design*, pp. 19–25, April 2000.

[12] J. Cong, J. Fang, and Y. Zhang, "Multilevel Approach to Full-Chip Gridless Routing," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 396–403, Nov. 2001.

[13] Y. Chang and S. Lin, "MR: a New Framework for Multilevel Full-Chip Routing," *IEEE Transactions on Computer-Aided Design*, vol. 23, pp. 793–800, May 2004.

[14] S. Li, R. Yeung, and N. Cai, "Linear Network Coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, Feb. 2003.

[15] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, Oct. 2003.

[16] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting," in *Proceedings of the IEEE International Symposium on Information Theory*, July 2003.

[17] A. Agarwal and M. Chariker, "On the Advantage of Network Coding for Improving Network Throughput," *Information Theory Workshop, IEEE*, pp. 247–249,

24–29, Oct. 2004.

[18] M. Langberg, A. Sprintson and J. Bruck, "The Encoding Complexity of Network Coding," *IEEE Transactions on Information Theory*, vol. 52, pp. 2386–2397, June 2006.

[19] T. Ho, "Networking from a network coding perspective," Ph.D. Dissertation, Massachusetts Institute of Technology, EECS Department, May 2004.

[20] Z. Li, B. Li, D. Jiang and L. Lau, "On Achieving Optimal Throughputwith Network Coding," *INFOCOM 2005, Miami*, vol. 3, pp. 2184–2194, March 2005.

[21] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Transactions on Information Theory*, vol. 51, pp. 1973–1982, June 2005.

[22] M. Hanan, "On Steiner's Problem with Rectilinear Distance," *SIAM Journal of Applied Mathematics*, vol. 14, pp. 255–265, March 1966.

[23] E. Dijkstra, "A note on two problems in connection with graphs," in *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[24] M. Garey, D. Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete," *SIAM Journal of Applied Mathematics*, vol. 32, pp. 826–834, 1977.

[25] J. Ganley and J. Cohoon, "Routing a Multi-Terminal Critical Net: Steiner Tree Construction in the Presence of Obstacles," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, pp. 113–116, June 1994.

[26] J. Leeuwen M. Kramer, "The Complexity of Wire-routing and Finding Minimum area Layouts for Arbitrary VLSI Circuits," *Advances in computing research*, vol. 2: VLSI theory, JAI Press, London, 1984.

[27] T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms*, chapter Single-Source Shortest Paths, pp. 580–620, second edition, Boston, MA: McGraw-Hill, 2001.

[28] N. Sherwani, *Algorithms for VLSI physical design automation*, chapter Global Routing, pp. 247–291, third edition, Norwell, MA: Kluwer Academic Publishers, 1999.

[29] ILOG CPLEX, in http://www.ilog.com/products/cplex, Jan. 2007.

[30] GeoSteiner Homepage, in http://www.diku.dk/geosteiner, Sept. 2006.

[31] ISPD98/IBM benchmarks, in http://www.ece.ucsb.edu/∼kastner/labyrinth/benchmarks, April 2007.

VITA

Nikhil Pandit Nemade received his Bachelor of Engineering degree in Electronics and Telecommunication Engineering from the Government College of Engineering, Pune (COEP), India in June 2005. In August 2005, he joined Texas A&M University to pursue his Master's degree in computer engineering. His research at Texas A&M is focused on exploring the opportunities of applying network coding for routing on a VLSI chip. He has worked as an intern at Samsung Telecom America, Dallas and at Dell Inc., Austin during the summers of 2006 and 2007 respectively and received his Master of Science degree in Computer Engineering in May 2008.

Permanent Address:

Nikhil Nemade

C-18 Kalpanamati Society,

Aundh,

Pune - 411007

India

e-mail : nikhil.nemade@gmail.com

The typist for this thesis was Nikhil Nemade.