# TOWARDS OPTIMAL DATA TRANSMISSION BY NETWORK CODING

# TOWARDS OPTIMAL DATA TRANSMISSION BY NETWORK CODING

Mohammad Ravanbakhsh

DISSERTATION FOR
THE DEGREE OF PHILOSOPHIAE DOCTOR


THE SELMER CENTER
DEPARTMENT OF INFORMATICS
UNIVERSITY OF BERGEN
NORWAY

DECEMBER 2009

**تقدیم به مادر و پدر عزیزم**

# ABSTRACT

Communication protocols based on coded schemes and, in particular, network coding promise increased efficiency for future networks. The field is still young. Nevertheless, many important results have been already obtained. In this thesis, network coding is studied in three directions:

In one direction, wireless network coding is studied. Wireless communication involves sharing of scarce resources, and special care is required to maximize the utilization of the medium. Network coding, in that sense, has the potential to improve the capacity. There are many issues involved in modeling a wireless network. Adapting all of these issues to network coding is a difficult task. We have considered a simplified model that takes into account the broadcast property and the transmission power. Based on this setting, we analyze some aspects of network coding in wireless networks. In this study, optimization algorithms were needed to compute special flow graphs for network coding. We propose special optimization algorithms for that purpose. We have studied the type of cycles that appear in wireless broadcast networks, and through simulation we observe the penalty of avoiding cycles in wireless network coding.

In another direction, the delay is studied from the queueing theoretic point of view, for code based communication protocols. We have improved previous results and propose a better measurement for delay.

Finally, we study secure network coding. For secure network coding, a new attack model is studied and the secrecy capacity is improved by a concatenated secret sharing scheme.

# ACKNOWLEDGEMENTS

During my study in Norway, I have received an invaluable amount of support and encouragement from many individuals. It is my duty to use this opportunity and express my gratitude to all of them.

My immense appreciation is to my supervisors, Øyvind Ytrehus (the Yoda), Ángela I. Barbero and Dag Haugland for their knowledge, inspiration, enthusiasm and patience. You were with me all the way. It is for sure that this thesis would not have been possible without you.

I am very thankful to Matthew G. Parker, since it was him, who first introduced me to Øyvind and I found the opportunity to join the Selmer center.

In my PhD studies, I had the chance to share office with Mikael Gidlund (Yoda), Joakim G. Knudsen and Sondre Rønjom. I should thank you for the great environment and all the helps with Norwegian language. Discussions with Mikael on wireless communications and feedback channels are unforgettable.

Life was so easy with all the help I received from Signe Knappskog, Ida Holen, Liljan Myhre and Inger Nilsen.

Special thanks to Mehdi M. Hassanzadeh for all the help in everything, and Lars Eirik Danielsen especially for helping me with the thesis format, and Mohammad Reza Sohizadeh for all the great discussions and delicious food, and Eirik Rosnes. I am also very grateful to the people at the Informatics Department for providing an excellent atmosphere for research.

Finally, my sincere acknowledgment is to my family, for being an irreplaceable source of support and guidance in every step of my life and my studies.

# CONTENTS

# LIST OF FIGURES

# List of Tables

# Overview

# 1 EXECUTIVE SUMMARY

"Towards" has many different meanings, e.g. "happening at the moment", "coming soon", or "the direction to". The reason behind selecting the very first word of the thesis's title, towards, is that the word describes the current status of the network coding concept in communication theory and information theory. That is, network coding is a very young field with promising potential, but it has yet to enter and find its place among the protocols that exist today.[1]

In communication networks, packets travel through the network to reach their destination. Network nodes or *routers* forward their received packets to the next router, and the packet delivery is shaped when packets reach their final destination according to correct forwarding via routers. This may seem complete, but from an information theoretic point of view, the question is how we can utilize the networks best or, in other words, what the capacity is and how it can be achieved. Since network coding theory was introduced by Ahlswede et al. in [2], this question has been answered up to some point, and some insight about the network's capacity has been gained.

In wired networks, the concept of network coding can already answer many of the questions related to the capacity. In wireless networks, however, things are different. One aim of this thesis was to look into some aspects of wireless network coding. Based on different setups for network coding in wireless networks, topological features that have influence on the required transmission power have been studied. Optimization algorithms for wireless network coding are also proposed.

In a communication system, improving one side, throughput for example, is often accompanied by a penalty on other sides. Network coding has many throughput advantages compared to routing based techniques. In the literature, less attention is given to the delay in network coding and packet coding schemes in general. The work by Shrader et al. in [6], took a step in that direction. The results, however, did not coincide with simulation results. As a part of this thesis an improved method for the computation of the delay is proposed, which very closely agrees with simulation results.

Another topic studied in this thesis is secure network coding. When network coding is used, the final receiver is capable of recovering the encoded packets only if a sufficient number of these packets is collected. This feature of network coding is a generalization of the the type II wire-

[1] There are a few exceptions, for example Microsoft Content Distribution.

tap channel model. The problem in the type II wiretap channel is the following: There are some communication paths between a sender and a receiver. Assume that some, but not all, of these paths are wiretapped. The goal is to build a coding scheme in such way that a wiretapper is not capable of recovering any information without capturing a predetermined minimum amount of packets, while the receiver on the other hand is capable of reproducing the complete secret message. Secure network coding is a multicast version of this problem, where instead of one receiver there are more than one receiver. Some techniques for improving the capacity of secure network coding are proposed, and also a new attack model is studied.

## 2 BACKGROUND

In studying a communication network, many concepts are used to model the system. In the following sections, a brief overview of these concepts is given. Readers who are already familiar with these concepts can skip the corresponding parts.

### 2.1 COMPLEXITY OF ALGORITHMS

An *algorithm* is a finite set of instructions that starts from an initial point and always terminates in finite time. In terms of execution time, some algorithms perform faster than others. There are many factors involved in the execution time of an algorithm. Many of these factors are related to the technology of the machine that runs the algorithm. The aim of studying *complexity for algorithms* is to evaluate the speed of an algorithm independently of technological factors, i.e. in terms of a *Turing Machine* that captures asymptotic properties of complexity.

Each algorithm receives some parameters as input and performs the instructions over those parameters. The execution time of an algorithm naturally grows with the increase in the number of parameters it receives. One of the most common measures for complexity is based on studying how the execution time scales with respect to the size of the input. There are different notations to express this effect, but we will only describe the most commonly used notation, which is the big-$\mathcal{O}$ notation. Consider an algorithm with complexity function $f(n)$ where $n$ is the input size. The big-$\mathcal{O}$ notation states that $f(n)$ belongs to $\mathcal{O}(g(n))$ when $\lim_{n\to\infty} f(n)/g(n)$ is equal to a constant $k$. For example,

an algorithm whose time of execution is $f(n) = n^2 + 2n$ is in this form $\mathcal{O}(n^2)$.

Algorithms can be designed to solve a problem, or simply to verify if a given answer is a solution of a problem. The class of problems for which there exists a polynomial time algorithm for solving them are known as **P** and the class of problems for which there exists a polynomial time algorithm that can verify a solution are known as **NP**. It is easy to see that **P** $\subseteq$ **NP**.

**NP**-complete is a subset of **NP**. A problem $p$ in **NP** is also in **NP**-complete if and only if every other problem in **NP** can be transformed into $p$ in polynomial time. Since we do not know polynomial time complexity algorithms for solving **NP**-complete problems, a very difficult question in computer science that is still open is whether **P** = **NP**. Problems that can be reduced to an **NP**-complete problem by a polynomial factor are known as **NP**-hard. Figure 1 shows a simplified version of how these classes look as shown in [12].

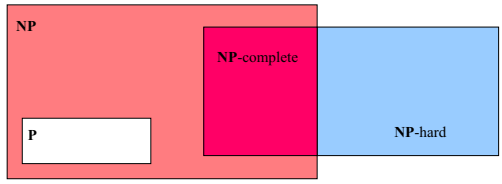


**Fig. 1:** *Complexity classes as shown in [12].*

## 2.2 Information Theory

In 1948, Claude E. Shannon introduced *information theory*. In his seminal paper [3], a measure for information was proposed. This measure is called *entropy*. For an *information source* $X$, which can be considered to be a random variable, information is represented by $M$ different symbols $i \in \{0, \ldots, M-1\}$ with the probability distribution $P(X)$, and the entropy $H(X)$ is computed as

$$H(X) = -\sum_{i=0}^{M-1} p_i \log_2 p_i$$

where $p_i = \Pr\{X = i\}$ for $i = 0 \ldots M-1$.

In a direct communication system, information is exchanged between two parties, which are known as a *transmitter* and a *receiver*. The physical

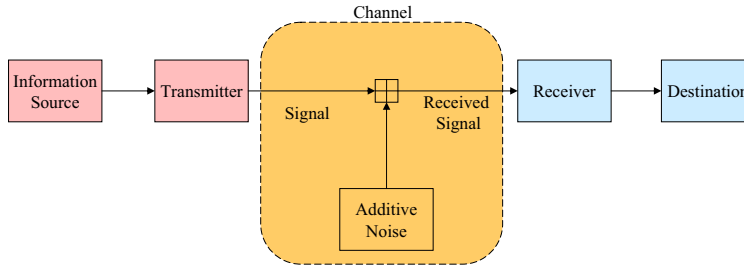


**Fig. 2:** *A typical communication system in Shannon's model.*

medium between these two parties is known as a *communication channel*, or channel for short. In Figure 2, a diagram of such a system is shown. In this configuration, symbols from information source $X$ travel through the channel, interact with the channel, and finally are received by the receiver. The received symbols $Y$ is a degraded version of the original information $X$. The amount of information we can learn about $X$ by having $Y$ is defined as *mutual information* and it is computable in the following way.

$$I(X;Y) = H(X) - H(X|Y). \tag{1}$$

In (1), the entropy function appears in two places. In the first place, $H(X)$ is the amount of information contained in the original message $X$. In the second place, $H(X|Y)$, is the conditional entropy and states that how much uncertainty will remain about $X$ by knowing $Y$. When $H(X|Y)$ is subtracted from $H(X)$, what is left is the information from source $X$ that is recovered at the destination. Shannon used this notation and computed the *capacity* of a channel in the following way.

$$C = \sup_{P(X)} I(X;Y). \tag{2}$$

One of the features of a signal is the *power*. A mechanical translation of power is the energy required for a signal to reach to the destination. For a channel with *Additive White Gaussian Noise* (AWGN), the capacity is a function of the signal power ($S_p$) to noise power ($N_p$) at the receiver:

$$\begin{aligned} C &= \log_2 \frac{S_p + N_p}{N_p} \\ &= \log_2(1 + \frac{S_p}{N_p}) \\ &= \log_2(1 + SNR). \end{aligned} \tag{3}$$

The interpretation of the capacity that was introduced by Shannon is that in a channel which is statistically defined, a part of the bandwidth should be given to redundant information in such a way that the receiver can use the extra information to retrieve the erroneously received information. This was the start of the research activity in *channel coding*, seeking to find codes that approach to the bound of (3). Although Shannon computed the capacity, he did not introduce methods for efficiently encoding and decoding for channels to reach that capacity. This task was essentially solved years later by other people. *Low Density Parity Check* codes (LDPC) by Gallager [9] and *turbo codes* [8] by Berrou et al. are among the famous capacity approaching codes. Theoretically, we can use turbo codes or other capacity approaching codes to design an almost error free communication between two parties.

The type of channel coding introduced above is known as *Forward-Error Correction* (FEC). There is another application of channel coding, which is based on error detection. This method is known as *Automatic-repeat ReQuest* (ARQ). In this method extra transmissions are required in case errors are detected thus coding is only used for detecting errors. In practice, hybrid approaches are also used.

## 2.3 Graph Theory

The origin of the concept of *graphs* dates back to the 18th century when Leonhard Euler first modeled the problem of the seven bridges of Königsberg with graphs [1].

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of two sets, $\mathcal{V}$ the set of *vertices* and $\mathcal{E}$ the set of *edges*. A vertex is simply a point in the space. Each edge is a line that connects two points. If an edge connects a point to itself then it is called a *loop*, otherwise the edge is a proper edge. In this thesis, all the edges will be proper edges. We denote the number of vertices of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, by $V = |\mathcal{V}|$. Similarly, the number of

edges is denoted by $E = |\mathcal{E}|$. In many problems related to graphs, the complexity is expressed a function of $V$ and $E$.

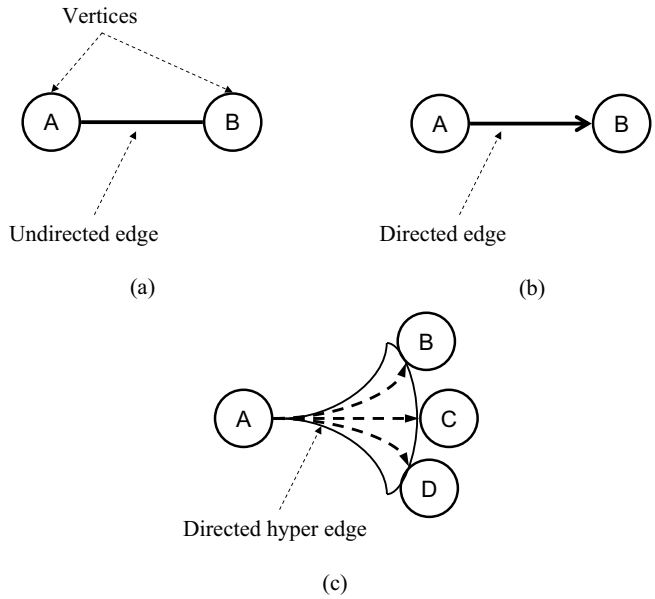


**Fig. 3:** *Different types of edges. (a), an undirected edge (b), a directed edge (c), a hyperedge.*

### 2.3.1 EDGE

The presence of an edge between two vertices is the sign of connection between those vertices. An edge between two vertices $i$ and $j$ is represented by the pair $(i, j)$ or $(j, i)$. This type of edge is known as *undirected* edge, since there is no ordering involved. There is another type of edge which is known as a *directed* edge. In a graph with directed edges, the connection from one vertex to another is not only described by the presence of an edge, but also by the direction of that edge. For a directed edge $e$, $start(e)$ and $end(e)$ denote the starting and finishing vertex of the edge $e$, respectively. The pair $(start(e), end(e))$, represents the edge $e$ and is not equivalent to $(end(e), start(e))$. The direction in a directed edge imposes an ordering relation. We show this relation by $start(e) \rightarrow end(e)$.

In Figure 3, different types of edges are shown. In Figure 3(a), both vertices A and B are connected and in Figure 3(b), only A is connected

to B. In this thesis, all the edges are directed. A hyperedge, as shown in Figure 3(c), is an edge that simultaneously connects one vertex to many other vertices. In theory, if we move over this edge we will reach multiple destinations at the same time. For a hyperedge, $end(e)$ is a set of vertices with at least one element.

### 2.3.2 VERTEX

For directed graphs, edges connected to a vertex are either incoming or outgoing edges. As a result of this, for each vertex we can consider two sets, one for the incoming and one for the outgoing edges. The set of incoming edges for vertex $i \in \mathcal{V}$ will be denoted by $\Gamma_I(i)$, while $\Gamma_O(i)$ will represent the outgoing edges.

In directed graphs, two edges $e$ and $e'$ are adjacent if there exists a vertex $i$ such that $e \in \Gamma_I(i)$ and $e' \in \Gamma_O(i)$. We describe this relation in the form $(e \prec e')$.

In modeling real life applications by graphs, vertices often represent places or objects with a special *location*. To preserve the location in the model, we assign a tuple of coordinates to each vertex. When we draw a graph, we can use these tuples to place the vertices relative to each other and with respect to a point of origin.

### 2.3.3 SUBGRAPH

Some of the common terminology that is used in graph theory is defined below.

**Definition 1.** *A* subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ *of a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *is a graph where* $\mathcal{V}' \subseteq \mathcal{V}$ *and* $\mathcal{E}' \subseteq \mathcal{E}$.

**Definition 2.** *A* path *in a directed graph is a subgraph that is the result of visiting vertices by moving along directed edges, such that the directions of the edges are not violated and each edge is visited at most once. In other words, consecutive edges e and* $e'$ *have the* $(e \prec e')$ *relation. For example,* $e_1 \prec e_2 \prec \cdots e_n$ *is a path.*

**Definition 3.** *A path in a directed graph that starts and ends in the same vertex, is called a* directed cycle[2]. *For example,* $e_1 \prec e_2 \prec \cdots e_n \prec e_1$ *is a cycle.*

[2]Through this thesis, for convenience we use *cycle* instead of *directed cycle*.

## 2.4 Queueing Theory

We can see lines or *queues* in almost every place where a server is handling customers. In a server system, customers arrive in the system and are served by the server. The arrival pattern can be deterministic (regular) or random. A deterministic arrival pattern has a rate $\lambda$ while for a random pattern besides $\lambda$, usually a multi-parameter probability distribution for the arrival pattern is also considered. The same is true for the service time. The service rate [3] is denoted by $q$. If both arrival pattern and service time are deterministic, the problem is not interesting, and in queueing theory at least one of the parts has a randomness. When new customers arrive to the server, they may find it busy. Therefore, new customers need to wait ,queue up, until other customers in the queue have been served.

There are interesting questions around a queueing system, e.g. waiting time in the queue, service time, number of the customers in the queue and etc. Since the arrival pattern of customers or service time are random, quantities related to the queue will be random too and we can assign a random variable to each of them with a specific distribution.

The Kendall notation [10] has been developed to describe queueing systems and has the form $A/B/c/K/m/Z$. $A$ is the distribution of the interarrival time of customers, $B$ is the service time distribution, $c$ is the number of servers, $K$ the system capacity (the maximum number of customers allowed in the system), $m$ the number of all potential customers and $Z$ the queue discipline. Usually the shorter version $A/B/c$ is used and it is assumed that customers are serviced according to a *first come first services* (FCFS) policy. Systems with exponential distribution for interarrival time and service time are denoted by $M/M/1$.

With queueing theory techniques, it is possible to compute average values for different quantities in a queue. The average waiting time in the system (including the service time), the average waiting time in the system (excluding the service time), the average number of customers in the system (including those in the service), and the average number of customers waiting in the queue are denoted by $W$, $W_q$, $L$ and $L_q$ , respectively. For the average service time ($W_s$) we have:

$$W_q = W - W_s.$$

[3] In a stable system, where the length of the queue does not grow to infinity, we have $\lambda < q$.

Little [10] has shown that for a steady state queueing system, under very general conditions, the following holds:

$$L = \lambda W$$

and

$$L_q = \lambda W_q.$$

Assuming that we know $\lambda$ and the average service time, knowledge about any of the four mentioned quantities ($W$, $W_q$, $L$ and $L_q$), will enable us to compute the other three. This can be achieved by solving the steady-state probability corresponding to that quantity. For example if $S_t$ is the random variable that shows the status of the quantity under study at time $t$ then the steady state probability $P_k$ is computed as:

$$P_k = \lim_{t \to \infty} \Pr\{S_t = k\}.$$

Then the average value can be computed from the expected value of that quantity with probability distribution $P_k$.

## 3 More on Graphs: Edge Labels

When modeling real life problems with graphs, in order to preserve properties that are of importance to us, we need to consider some other properties of the edges. Here, in this section, some of these properties, which we call *edge labels*, will be defined. Each edge is assigned a collection of numbers called labels.

### 3.1 Cost

A *cost*[4] is a basic edge attribute. In maps, for example, the travel price, travel time, distance, etc, between two cities can be regarded as costs. In a particular problem, where we look for a subgraph that is a potential solution to our problem, one of the important factors we are interested in is the cost of that subgraph and how low that cost can be.

We denote the cost of the edge $(i, j)$ by $a_{ij}$. For the subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, the cost is:

$$A(\mathcal{E}') = \sum_{(i,j) \in \mathcal{E}'} a_{ij}.$$

[4]Cost can be a scalar or a vector of real numbers. In this thesis, we assume different costs related to an edge are compressed in and represented by a single real number.

The *Dijkstra algorithm* finds the minimum cost (shortest) path between two vertices. This algorithm is also known as the single-source shortest path algorithm. Dijkstra's algorithm searches through the graph, labels a special vertex as the *source*, and finds the shortest path between the source vertex and all the other vertices of the graph. So the output of Dijkstra's algorithm is a *tree* that is rooted at the source vertex and includes all the other vertices of the graph. The running time of the Dijkstra algorithm has complexity $\mathcal{O}((E+V)\log(V))$ for a graph with $E$ edges and $V$ vertices [12].

Another problem in graph theory that is related to the cost metric is the *minimum spanning tree*. A minimum spanning tree is a subgraph which is a tree that covers all the vertices and has minimum cost. It is not necessarily unique. There are efficient algorithms to find a minimum spanning tree. Dijkstra's algorithm is very similar to the minimum spanning tree algorithms.

An interesting phenomenon is that a simple modification of the problem will convert it into a very difficult problem. The modification is the following. Instead of searching for the minimal tree that covers all the vertices, find the tree that contains a specified subset of vertices, but not necessarily all the vertices, and has minimum cost. This problem is known as the *Steiner tree* problem, and belongs to the **NP**-hard class of problems. A modified version of Dijkstra's algorithm is a heuristic for the Steiner tree problem. Later we will explain more about this algorithm.

## 3.2 Capacity

Another label for edges is the *capacity*. We use capacity when we want to model networks with transportable objects or matter that can be quantified. In an edge $e$, the capacity is the maximum rate, measured in e.g. "quantity per use" or "quantity per time unit", by which objects can move from $start(e)$ vertex to $end(e)$ vertex. We can generalize the case for the vertices of edge $e$, to any two distinct vertices in the graph by applying the *flow* concept. In the following we will introduce the definition of a flow.

A *transportation network* is a directed labeled graph in which there is one vertex $s$ with no incoming edges (*source*) and one vertex $t$ with no outgoing edges (*sink*)[5]. In this graph, the label of each edge $(i,j)$ is the capacity of that edge and is denoted by $c_{ij}$. Flow in a transportation

[5]Other vertices are called *relays*.

network is a map $F : \mathcal{E} \rightarrow \mathbb{R}$, where an edge $(i, j)$ is mapped to a real number $z_{ij}$ with the following properties:

1. $0 \leq z_{ij} \leq c_{ij}\ \forall (i,j) \in \mathcal{E}$
2. $\sum_{k \in \Gamma_O(i)} z_{ik} = \sum_{j \in \Gamma_I(i)} z_{ji}\ \forall i \in \mathcal{V} \setminus \{s, t\}$.

The value of a flow is $|F| = \sum_{k \in \Gamma_O(s)} z_{sk} = \sum_{j \in \Gamma_I(t)} z_{jt}$, and the *max-flow* is a flow with maximum value. In [4, 5], more details about max-flow can be found. The Ford-Fulkerson algorithm introduced in [4], can efficiently compute the max-flow.

In a transportation network with several sources or sinks, a super source and a super sink can be defined in order to transform it into a transportation network with one source and one sink. Nevertheless, in communication networks, in case of several sinks, a flow $F^t$ is considered for each sink $t$. The union of the edges with $z^t_{ij} > 0$ form a subgraph which is known as the *flow graph* to sink $t$. Sometimes, abusing the notation, we will simply call it the flow to sink $t$. It is easy to see that flow graphs to different sinks can overlap and share edges. Based on the properties of the problem, the capacity of an edge can be either used independently or be divided among different flows overlapping on that edge.

## 3.3 Linear Programming and Flow Problems

The idea behind the Ford-Fulkerson algorithm is to find *augmenting* paths between a source and a sink until no more augmenting paths can be found. Each augmenting path allows an increase in the flow value. The final result of the Ford-Fulkerson algorithm is a max-flow. While the value of a max-flow is obviously unique, a max-flow is not necessarily unique.

If both capacity and cost are known for all the edges, one problem is to find the max-flow with the lowest cost, for wich the cost is defined as $\sum_{(i,j) \in \mathcal{E}} a_{ij} z_{ij}$. This problem is known as the *minimum cost flow* problem. A useful heuristic approach to the Ford-Fulkerson problem uses a modified version of Dijkstra's algorithm to search for augmenting paths. In addition to graph theoretic approaches to flow problems, there are numerical methods based on *linear programming* (LP) to solve the max-flow. The following LP program (4)-(6) defines the minimum cost flow

problem for the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with source vertex $s$ and sink vertex $t$:

$$\min \sum_{(i,j)\in\mathcal{E}} a_{ij} z_{ij} \tag{4}$$

$$0 \leq z_{ij} \leq c_{ij}, \forall (i,j) \in \mathcal{E} \tag{5}$$

$$\sum_{k\in\Gamma_O(i)} z_{ik} - \sum_{j\in\Gamma_I(i)} z_{ji} = \Delta(i), \forall i \in \mathcal{V} \tag{6}$$

where

$$\Delta(i) = \begin{cases} R & i = s \\ -R & i = t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

and $R$ is the max-flow value for sink vertex $t$ from source vertex $s$.

The *simplex* algorithm [7] can solve this system of equations and find the correct values for $z_{ij}$ (weakly in polynomial time). The set of edges with $z_{ij} > 0$ form a subgraph that is the flow graph with max-flow value. In general the $z_{ij}$ values are real numbers. However, if capacities are integers, then the Simplex algorithm applied to this program will produce an optimal solution in which $z_{ij}$ is integer for all $(i,j) \in \mathcal{E}$ [12].

Actually, whether the capacities are integers or not, requiring $z_{ij}$ to be integers can be an extra constraint in the program. In (4)-(6), by default we consider $z_{ij} \in \mathbb{R}$. A solution with this assumption is known as a *relaxed solution*. However, in (7), if $R$ has an integer value and we have the extra condition that $z_{ij} \in \mathbb{N}$ then the problem in (4)-(6) is no longer an LP program but an *integer program*. In general, problems of this form belong to the class of **NP**-hard problems.

## 4 Network Information Theory

Here, we will combine the concepts defined in the previous sections. A data *network* consists of computers or any digital communication equipment that are linked together via communication channels. This network can be modeled by a graph in such way that vertices represent the computers or other digital equipments, and edges represent the communication channels between any two communicating computers. From this point, we will call vertices as *nodes* and edges as *links*.

In Figure 4, a data network, from a transmitter and receiver point of view, is shown. The transmitter on the left is the source node where

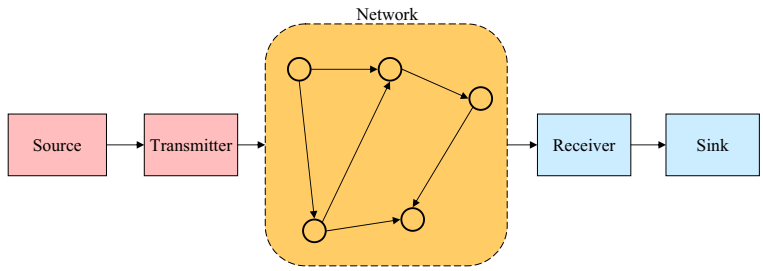


**Fig. 4:** *Network information theory is an extension of Shannon's model for networks.*

information is generated and starts its travel through the network to the receiver.

Network information theory can be viewed as an extension to Shannon's information theory of channels for networks. Typical questions from network information theory are

- What is the capacity of a network?
- How much flow can be injected into a network from source nodes, and received at sink nodes?

The single source single sink case is a classic problem in computer networks and was briefly presented in section 3.2. There are different settings for a network to operate, and these models are explained in more detail in section 4.1.

## 4.1 Communication models: a Non-coding Perspective

In a non-coding communication network, delivery of *packets* of information is based on copying and forwarding. In the following, these two strategies are explained in more detail for two main operating modes of a network. These modes are *unicast* and *multicast*.

In a unicast, a source node communicates with a sink node. In this scenario, packets travel from the source node to the sink node by hopping from one node to another. From an information theoretic point of view, the capacity of a unicast is the max-flow value, where the capacity of each link is interpreted in the information theoretic sense. Then it is assumed that each link is used with a capacity approaching channel code. This approach is valid under the assumption that the channel capacity can be attained independently in each link, and there

**Table 1:** *Optimal Flow Optimization Algorithms for Non-coding Packet Delivery Scenarios*

| Operating Mode | Main Feature | Complexity | |
|---|---|---|---|
| | | Integer | Relaxed |
| Unicast | Peer-to-Peer | **P** | **P** |
| Multicommodity | Multiple Unicasts | **NP**-hard | **P** |
| Multicast | Steiner tree | **NP**-hard | **P** |
| Tree Packing | Multiple Multicast tree | **NP**-hard | Not Available |
| Broadcast | Special Multicast | **P** | **P** |

is no bottleneck in the processing of packets in the nodes. The first of these assumptions is satisfied e.g in a wired point-to-point network. In such a network, it is possible to use the Ford-Fulkerson algorithm or any other optimization algorithm, to compute the capacity of a unicast and the required flowgraph. If link capacities are integers, then an integer solution can be found in polynomial time.

Another setting based on unicast is *muticommodity*. A multicommodity is simply multiple and simultaneous unicasts.

The other mode is multicast. In a multicast scenario, the same information from one source node is sent to a set of sink nodes. This seems like a multicommodity in which a single source node is communicating by many sink nodes. But, for the multicommodity scenario, the strategy for packet delivery is only based on forwarding of packets. In a multicast, using this strategy is not an optimum approach. Besides forwarding, the copying strategy can also be used. In copying, different copies of a received packet can be sent on different outgoing links. When capacities are integers and we are interested in an integer solution to this problem, the problem is to find a multicast tree. The optimal solution to this problem is a Steiner tree. In the case of multiple multicasts, where we look for multiple [disjoint] multicast trees from a source node to sink nodes, the problem is known as tree packing [11]. The importance of the tree packing problem is that if we are able to find a solution for this problem, then we have the maximum capacity of a network based on copying-forwarding policy.

Broadcast is a special case of multicast, where a single source is sending the same information to all the nodes of the network. In Table 1, the above mentioned modes of operation are tabulated with their computational complexity.

# 5 WIRELESS COMMUNICATION

In wired communication, the capacity of each link is fixed and predefined. This makes it easier to compute flows in wired networks. In *wireless communication*, things are different. In the following sections, we will view wireless communication from different perspectives and study the factors that influence the capacity of wireless communication.

## 5.1 SINGLE USER WIRELESS LINK

In a wireless communication between a transmitter and a receiver, the channel is simply the environment. The signal that is generated at the transmitter propagates through the environment, and in one point it reaches the receiver. Figure 5 depicts this process. As it is shown in the figure, each ray of the transmitted signal can experience different conditions. Three possible scenarios are total absorption, reflection and direct reception (line of sight). In direct reception, some part of the signal will be absorbed by the environment and we can consider this as a generalization of the absorption scenario (partial absorption). For these conditions four interesting phenomena can be described. The first one is related to the reflection and the rest are common for all.

- **Multi-path:** Reflected beams will arrive at the receiver with different delays with respect to the line of sight, and they add together. This will cause the final signal to be degraded in places where the phase of the received signals is opposite. The effect is known as multi-path *fading*.
- **Doppler Effect:** One property of wireless communication is the potential mobility of transmitter or receiver. When transmitter and receiver move with respect to each other, the frequency of the received signal can be different from what it is supposed to be.
- **Inter-Symbol Interference:** Ideally, a receiver should be synchronized with the sender in order to correctly sample the transmitted symbols. In an out of sync symbol reception, symbols interfere with each other. This effect is known as Inter-Symbol Interference (ISI). In wireless communication, mobility of the wireless device along with the multi-path and Doppler effects, cause an ISI effect.
- **Path Loss:** The signal from the transmitter travels in concentric spheres. The transmission power at the transmitter spreads over

the surface of the spheres as the signal moves. As the radius of the sphere gets bigger and bigger the power density on the surface starts to decrease. The result of this reduction is a weaker signal at the receiver. Besides this reduction in power, some part of the signal will be absorbed by the environment. Therefore, for the distance $d$, the reduction can be considered proportional to $d^{-\alpha}$ for some path loss constant $\alpha$ with $2 \leq \alpha \leq 4$.

Among the phenomena we discussed above, the three first ones are usually addressed with coding and special techniques, and we only need to take care of the path loss by correctly assigning the power.

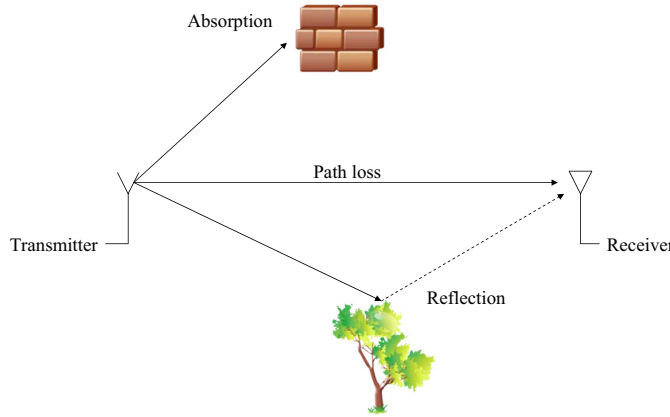


**Fig. 5:** *Wireless medium between a transmitter and a receiver.*

## 5.2 Multiple User Wireless Links

In a wireless network with many users, the medium is shared among all the users. Due to this sharing, the communication is subject to two phenomena which do not occur in a point-to-point wired link:

- **Interference:** In wireless networks with multiple users, simultaneous communications can interfere with each other. There are techniques that deal with this problem. Multiplexing approaches like *Time Division Multiple Access* (TDMA) and *Frequency Division Multiple Access* (FDMA) are well known strategies. In these multiplexing approaches, the interference is avoided at the price of reduced efficiency. From the queuing theoretical point of view, some transmitters in such a static multiplexing approach have to

work at lower rates because less time or frequency is assigned to them. Therefore, the overall capacity is lowered. In the case where interference is allowed, one of the well known techniques is *Coded Division Multiple Access* (CDMA). In CDMA based systems, the capacity of each link is calculated in the following way. The capacity for the link $(i, j)$ is $C_{ij} = log_2(1 + SINR_j)$ where

$$SINR_j = \frac{P_{ij}}{\sigma^2 + \sum_{k \neq i} P_{kj}}.$$

The $P_{mn}$ is the received power from node $m$ at node $n$. We can see that in a CDMA system, the capacity of one link is not independent of other links, since the activity of one link influences the capacity of other links. Many algorithms that are used in wired cases require substantial changes to adapt to this feature. In this thesis we have not explicitly considered interference, in order to simplify the discussion.

- **Broadcast Channel:** In a wireless communication, when a transmitter sends a signal, all the receivers within its hearing range can listen to it. Since each receiver has a different position with respect to the transmitter, the receivers experience different channel qualities and therefore the capacity of each of those is not the same. Based on coding for the degraded broadcast channel [21], it is possible to transmit simultaneously at the capacity of each receiver, which means that receivers with higher capacity will not be limited by receivers with lower capacity. A broadcast limitation still remains in the sense that the sender can not transmit different messages to different receivers.

  In principle, each network node might apply broadcast coding in order to transmit information to all of its neighbors within some predefined or locally optimized vicinity [28]. In practice, combined broadcast and network coding is poorly understood and is the topic of future research. A simple case, is the *relay network*, where a relay node is placed between a transmitter and a receiver. The capacity of this network is not known in general. We therefore restrict the discussion to a model where nodes broadcast the same information (in content and amount) to all neighbors within a hearing range that is defined by the transmitters signal power and the parameter $\alpha$ (related to path loss).

### 5.2.1 WIRELESS NETWORKS

Consider a graph that represents a fixed data network. It has nodes and links, and a capacity associated to each link. In particular, nodes can not move, links are fixed, no change to the set of links is possible, and capacity is constant. Data networks of this type are typically *wired networks*. In this section we will define another type of network that is not constrained in this way. These networks are known as *wireless networks*.

#### 5.2.1.1 CELLULAR NETWORKS

In these networks, there are two types of nodes: *Fixed nodes* or *base stations* (BS) that are connected together via a wired network, and *mobile nodes* that can move. In these networks, each mobile node communicates directly with one BS in its range, and through this link the network of BSs can connect this mobile node to any other mobile node. We will not consider cellular networks in this thesis.

#### 5.2.1.2 DIRECTED WIRELESS LINKS

These networks are very similar to wired networks. Each link, similar to a wired link, has a fixed capacity, and the capacity of each link is not affected by the other links. There is a line of sight between each transmitter-receiver pair. Fixed microwave tower to tower links is an example of these type of wireless networks.

#### 5.2.1.3 WIRELESS BROADCAST NETWORK

In a wireless network of this kind, nodes are fixed and each node is equipped with a single transmitting antenna that often has a tunable transmitting power. In order to model a wireless network of this kind with graphs, there are two steps to follow.

- **Step One:** Based on the location of the wireless node, a vertex is considered for each node with respect to an origin point in a Cartesian coordinate system.
- **Step Two:** Each node has a single antenna with a predefined power setting $p$. According to the power $p$, a coverage radius $d$ is considered for each node, where the receivers located in this coverage radius will receive the least required $\text{SNR}_c$ to achieve the capacity defined in (2). This coverage radius is proportional to

$\sqrt[\alpha]{p/\mathrm{SNR}_c}$ where $\alpha : 2 \leq \alpha \leq 4$ is the path loss constant. If noise power is known and constant, it suffices to say that the coverage radius is proportional to $k \cdot \sqrt[\alpha]{p}$. In a wireless network, when a wireless node transmits a signal, all the nodes in the range will be able to listen to that signal. This type of channel is known as a *broadcast channel*. A hyperedge from that node is considered to all the nodes in the coverage range. To model this effect in a graph model without hyperedges, we replace each node with two co-located nodes. The first node receives all the incoming links to the original node. The second node has the outgoing links of the original node. A middle link from the first node to the second one is also considered. This middle link collects all the streams from incoming links and then a single stream is presented at outgoing links in the second node. For simplification we consider that, despite different distances and different interfering conditions, all the outgoing links of the second node have the same capacity and the capacity of the middle link that connect two nodes is also the same.

In these networks, connectivity is a big issue. In Section 5.3, this problem is explained.

From the point of view of network coding (Section 6), wireless broadcast networks with interference are the most interesting and challenging.

### 5.3 Unit rate Capacity Links

In wireless networks, in order to provide the required connectivity, transmitters should set their transmitting power. This procedure can be done in two fashions, symmetric and asymmetric. For the asymmetric fashion, we do not know any efficient algorithm to find the transmission powers.

The transmission power of wireless nodes and the optimal value for transmission power can be computed efficiently. In [31] an algorithm is introduced for this purpose.

With the knowledge of this optimal transmission power, the parameter $\alpha$, SNR and other constants related to path loss, it is possible to compute the radius of coverage around each wireless node. Then the wireless nodes can establish links to their neighbors in their coverage radius. In Figure 6(a), the coverage radius is shown, and in Figure 6(b) the wireless node is connected to its neighbors.

The wireless links in Figure 6(b) is a hyperedge because the broadcast property of wireless links. It is possible to model this network with a graph that preserves the broadcast property by replacing each wireless node by two co-located nodes, the receiver node and the transmitter node. All the incoming links of the original wireless node will be connected to the receiver node, and similarly al the outgoing links will be reestablished from the transmitter node. A link should connect the receiver node to its corresponding transmitter node. In Figure 6(c) this process is shown.

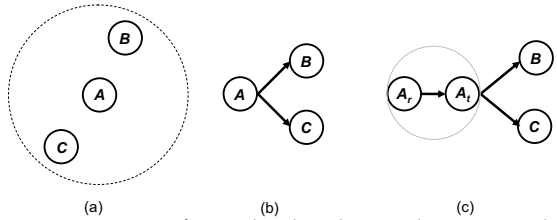


**Fig. 6:** *The conversion process of a wireless broadcast node into a graph node with unit rate link capacities.*

If no broadcast coding is considered then the capacity of the hyperedge is bounded by the coverage radius. Since all the transmission powers are equal, all the links can be considered to have unit rate capacity.

## 5.4 Random Geometric Graphs

In studying wireless networks, we consider an ensemble of random graphs known as *random geometric graphs* [14]. In a finite random geometric graph, a finite number of nodes are randomly placed in a $n$-dimensional space $\mathbb{R}^n$ according to a probability distribution for each coordinate. If the set $\mathcal{X}$ of nodes is finite, then the set of locations in the random geometric graph is a finite subset of $\mathbb{R}^n$. The edge set of such a graph is defined according to a distance function. Let $\|.\|$ be some norm on $\mathbb{R}^n$, for example the Euclidian norm and let $d$ be the coverage radius[6]. Two nodes $X, Y \in \mathcal{X}$ are connected by a link if $\|Y - X\| \leq d$. For wireless networks, $n$ is usually 2 or 3, and the norm function is the Euclidian distance. If we ignore broadcast constraints, then the resulting graph of this type is an undirected graph because the existence of an edge $(X, Y)$ implies the existence of an edge $(Y, X)$.

[6] $d = k \cdot \sqrt[n]{p}$ defined in Section 5.2.1.3.

In another model, the connectivity is based on $k$-nearest neighbors, and each node has a link to its $k$-nearest neighbors. This model is sometimes known as *directed proximity graphs*. Connectivity in random wireless graphs are studied in [15, 16].

## 6 Network Coding

As compared to traditional packet-switching networks where packets are virtually unmodified during transmission to the final receiver, coded-packet schemes [27] and specifically *network coding* [2] allow network nodes or routers to modify packets by performing *mathematical operations* over their received packets to form new packets; trusting destinations to be able to reconstruct the original information. This new concept in information theory literature has shown many interesting properties and potentials for future networks. There is still much research going on and the concept has found its way in many areas of information theory and communication theory.

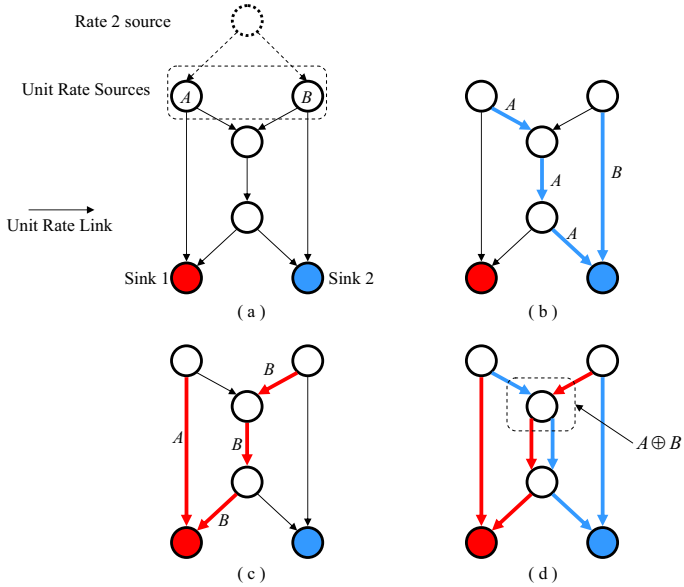


**Fig. 7:** *The butterfly example. (a) shows a directed graph for representation of a network with two unit rate source nodes and two sink nodes. (b) shows the flow that carries the max-flow to sink 2. (c) is flow for sink 1 with max-flow rate. (d) is the flow graph which is the union of the two flows and shows where the coding takes place.*

In a unicast scenario, *original packets* at the source node travel through the network until they reach to the sink node. In this mode, the maximum rate for transferring packets from the source node to the sink node is possible by only using a forwarding strategy. In a multicast, a source node is communicating the same information to a set of sink nodes. Since they all receive the same information and with the same rate, the capacity is bounded by the minimum max-flow among all the sink nodes. In this scenario a flow for each sink is considered. Flows to different sinks may overlap. This is not always possible with the store-and-forward technique. The famous butterfly network, is a classic example that describes this situation. The butterfly network is shown in Figure 7.

In this network the source node has two unite rate streams, that can be considered as two unit rate sources namely $A$ and $B$, which generate a bit length data packet, and want to communicate with the sink nodes. All the links in this network have unit capacity. Although the max-flow value to each of the sinks is equal to two, without coding it is not possible to setup a multicast with rate two. The bottleneck in this problem is the middle link in which the two flows overlap.

In Figure 7(b) and (c) flows to each sink node are shown. In Figure 7(d), the overlapping of the two flows is shown. In a non-coding packet delivery, the middle link has to decide which of the flows to pass. The coding solution to this problem is the simple *exclusive-or* operation that takes the packets from different flows and combine them together. In this setting, each sink node receives an original packet and a coded packet. In order to recover the other original packet at each sink node, a simple exclusive-or operation is needed. The coding procedure is shown in Figure 7(d).

In [2], it is proved that only linear operations over a finite field are needed to achieve the minimum max-flow capacity. In general, conditions for a feasible encoding are studied in [13]. In Figure 7(d), the middle link receives two flows. For this network, a feasible network coding can be found in a finite field of size 2, i. e. GF(2). Therefore, the original packets, $A$ and $B$ at the source node can be viewed as binary vectors. In the coding procedure, a mathematical operation takes place over the corresponding bits of these vectors. The mathematical operation in here is a modulo 2 addition. In Figure 7(d), the sink node with red color receives two packets, one which is original and the other which is a coded packet. Since the sink node is aware of the encoding

procedure, it can set up a linear system of equations in the following way:

$$\mathbf{A}X = Y$$

where $Y$ is a vector that contains the received packets, $X$ represents the vector of the original packets and $\mathbf{A}$ is the encoding matrix. For the red sink node in the above example we have:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} A \\ A+B \end{pmatrix}.$$

Recovery of the original information $X$ can be done by using methods to solve systems of linear equations, e.g. *Gaussian Elimination*.

There are two different methods for setting up a network coding session over a network: *random network coding* and *deterministic network coding*. In random network coding there is no encoding phase involved and coding takes place when network nodes combine their received packets, using random combinations, and transmit them to other nodes. This procedure continues until all sink nodes receive enough coded packets to recover the original packets. The sinks will solve systems of equations with different matrices each time. In a deterministic network coding, there are two steps involved. In the first step, flows to each sink are constructed and in the second step, the way to encode at each link is designed. The deterministic encoding will then be used in each multicast session. Observe that the sinks will then solve systems of equations with the same matrix in all the sessions. Deterministic encoding algorithms are described in[13, 17–19, 25, 26].

The reason why network coding helps to achieve optimal throughput can be sketched as follows: when multiple flows are streaming in a network, we can define different flow variables for each link and represent them as a vector. In this vector the first element is the overall flow rate passing through that link. We denote this value for link $(i, j)$ by $z_{ij}$. Then, we have a flow variable for each sink. The amount of flow that is passing through link $(i, j)$ related to sink $t$ is denoted by $x_{ij}^{(t)}$.

In non-coding packet delivery systems, the flows are treated like physical objects that occupy space. Therefore, flows with common links share the capacity of the links where they overlap. This can be written in mathematical language as:

$$z_{ij} = \sum_{t \in \mathcal{T}} x_{ij}^{(t)} \tag{8}$$

whereas for network coding it can be is expressed as:

$$z_{ij} = \max_{t \in \mathcal{T}} x_{ij}^{(t)}. \tag{9}$$

The mathematical expressions in (8) and (9) are shown in Figure 8. This may also explain the complexity for finding a feasible solution when non-coding techniques are considered, e.g. Steiner tree packing, while a feasible solution in network coding is very easily obtainable.

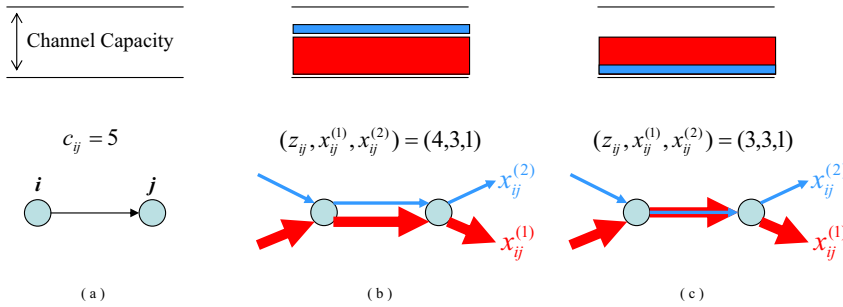


**Fig. 8:** *Overlapping flows. (a) shows a link with capacity value 5. (b) shows two overlapping flows passing through the link in a traditional (non-coding) packet delivery system. (c) shows two overlapping flows passing through the link in a network coding based packet delivery system.*

### 6.1 Random Network Coding

Random network coding [20] is a practical approach for setting up a multicast. In random network coding, it is assumed that $k$ original packets are at the source node. These packets start traveling through the network until they reach to sink nodes. Relay nodes will combine incoming packets using random linear combinations over a finite field, and then new coded packets will continue the trip to the sink nodes. The coding coefficients are stored in the header of the packet and this makes possible to see which combination of the original packets is contained in a coded packet. When a sink node receives enough number of these coded packets it can perform a Gaussian elimination operation based on the header information and recover the original packets. When original packets are recovered, the sink node sends an acknowledgement to the source node. When the source node receives

acknowledgements from all the sink nodes, a mechanism is used to stop the stream of coded packets in the system.

In random network coding, field size should be selected large enough to guarantee a maximum rank with high probability. If the field size is too small, then the there is a large probability that coded packets arriving at the sink node will be linearly dependent on previously arrived packets, and thus will not contribute to the information transfer. Therefore a too small field can lead to excessively slow effective transmission. The other source of delay is the sink node that sends the last acknowledgement. This sink node slows the whole process. Since coded packets are received at the sink node, it is not possible to access the original packets before $k$ linearly independent packets have been received.

## 6.2 Cyclic Networks

The flow graph of a network can have cycles. These cycles are called *link cycles* in [26].

The flow to each sink $t$ induces a partial order on the edges that are involved in that flow. Given two directed edges $(A, B)$ and $(B, C)$ in the flow, we say that $(A, B)$ precedes $(B, C)$ and write it as $(A, B) \prec (B, C)$ if there is a path in the flow to $t$ that contains the subpath $A \rightarrow B \rightarrow C$. If transitivity is added for the relation "$\prec$", the flow induces a partial order on the set $\mathcal{E}$ of edges.

**Definition 4.** *A* flow cycle *is a link cycle in the flow graph, in which the partial orderings induced by the flows involved in the cycle are not mutually consistent. See [25, 26].*

If a flow graph contains at least one flow cycle, it is still possible to find an efficient deterministic coding as shown in [25, 26].

## 6.3 Central Problems in Network Coding

### 6.3.1 Solved Problems

The major questions around network coding are concentrated over the flow graph formation and encoding phases. In the flow formation phase, optimization algorithms based on linear programming have been introduced [24]. However, this linear programming when it is constrained to have integer solutions is **NP**-hard [23]. In the encoding phase, finding the encoding with the lowest field size is known to be

**NP**-hard. However, there are polynomial time complexity algorithms that can provide a practical field size. Algorithms proposed in [18] and [25, 26] can perform an encoding in polynomial time.

In contrast to packing Steiner trees, network coding has the advantage that it can always find a feasible integer solution (though it may not be the optimal solution), while such thing is not possible in packing Steiner trees.

### 6.3.2 Open Problems

Network coding has been widely studied for wired networks. The intention of this section is to show that for future study of network coding, it is more interesting to study wireless network coding. Some open problems are also introduced.

In the domain of wired networks, the technology in the communication links (*physical layer*) has improved remarkably, and very high rate transmissions are made possible. *Fiber optic* communication and high speed *Ethernet* are well known achievements of this kind. But, in communication networks, links make up only one part of the system, the other part consists of the nodes or computers. Although the computational capacity of computers is increasing everyday, it is still far behind the improvements in the communication links. The consequence of this difference in capacity is that no matter how fast the computers operate, they are not capable of saturating the capacity of the communication links. Network coding may be used when two or more flows overlap in a link and the link does not have the required capacity to allow different streams to pass independently. In wired networks, as we discuss here, such a scenario is often unlikely since sufficient capacity is usually available. Therefore, in such networks, network coding presents limited advantages[7]. In wireless networks, this is different, and higher capacity for links is very expensive in many terms, e.g. overall capacity, transmission power, etc. In this case network coding is a great help.

In wireless broadcast networks, the capacity of links is influenced by other links and it is not as clear as it is in wired networks. In addition, the capacity region for a very simple Gaussian relay network[8] is still to be precisely determined. With these issues around wireless networks, the notion of max-flow is not straightforward for wireless networks, and is therefore an open problem. In special cases where CDMA approaches

[7]Except if nodes are defined as channels with finite capacity, e.g. [29].

[8]This network consists of a single source, single destination and a relay node.

are used, a simplified problem is studied in which the interference is taken care of by CDMA and the broadcast property of links is ignored [22]. In such studies, an optimization algorithm is introduced that produces a solution based on minimizing the transmission power and maximizing the overall capacity. The solutions of these algorithms produces flows with relaxed values. In cases like network coding, we are interested in integer solutions. This problem is also open.

Another direction in network coding is the multi-session network coding. In this scenario multiple network coding sessions are performing simultaneously. The capacity for this case is not known in general. This problem is common for both wired and wireless networks.

# 7 Summary of Papers

This thesis consists of five papers. In the following sections, a short overview of each paper is given.

## 7.1 Paper I

The first paper, entitled "Power savings of cyclic network coding for multicast on wireless networks," is co-authored by Ángela I. Barbero, Dag Haugland and Øyvind Ytrehus and will be presented at the *IEEE Information Theory Workshop* (ITW 2010) in Cairo, Egypt.

Cyclicity is one of the features that can be found in the flow graph of a network coding setup. A special form of cycles exists in network coding, and they need to be treated in a specific manner when encoding is taking place. These cycles are known as flow cycles. From the encoding point of view, this problem has been studied in the network coding literature. In this paper, we have studied the cycles for wireless network coding.

In order to study the cycles in a wireless network coding, we have considered some properties of wireless networks into our modeling. These properties are the broadcast nature of wireless links and the transmission power. In order to simplify the discussion, we study a network model which is common in the literature, and which ignores interference and the possibility of broadcast encoding. [9] From the broadcast feature in wireless networks, we proved that any cycle in the flow graph of wireless network coding is a flow cycle.

In some network coding papers, for simplicity, the flow graph is considered to be cycle free, and algorithms have been devised to extract a cycle free flow graph from a network infrastructure [30]. We studied such a constraint for wireless network coding over random geometric graphs. From the transmission power point of view we observed, through simulation and based on heuristic algorithms for flow graph computation, that on average such a restriction leads to an increase in the overall transmission power compared to the case where cycles are allowed.

In Figures 9 and 10 the average total transmission power for different cases is shown. The red lines are the acyclic results. The complete detail about the optimization algorithms can be found in [32].

[9] Observe, however, that a transmission power efficient coding scheme in general should induce less interference problems than a scheme which is less power efficient.

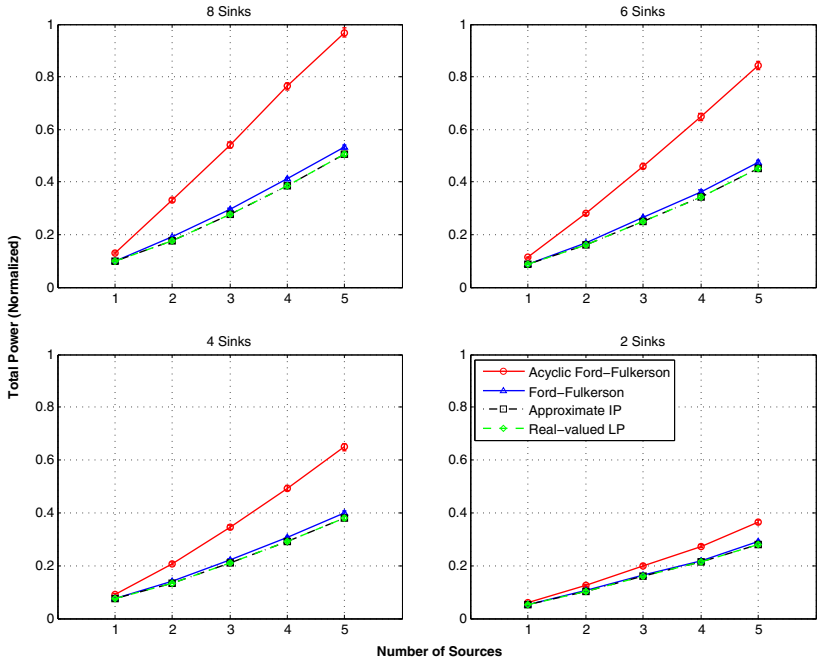


**Fig. 9:** *Power Comparison. The interval for each point indicates the 0.001- confidence interval for the total power. We have used $\alpha = 2$*

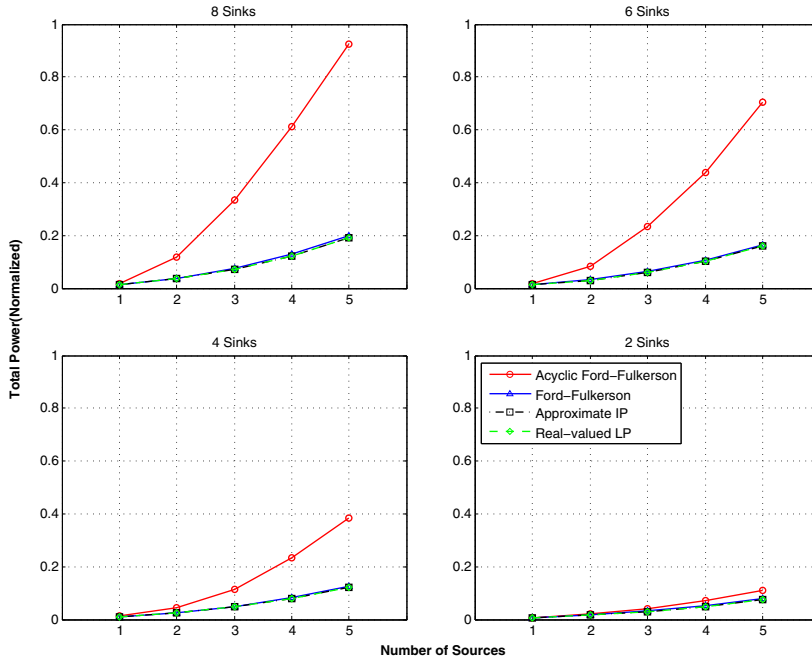


**Fig. 10:** *Power Comparison. The interval for each point indicates the 0.001- confidence interval for the total power. We have used $\alpha = 4$*

In Figure 11, an example of a cyclic solution is shown. Two other acyclic solutions for the same wireless network are shown in Figures 12 and 13. It is easy to check the extra transmission power shown in grey color. In Figures 11-13, for some wireless nodes the internal transmitter node that preserves the broadcast property is shown[10].

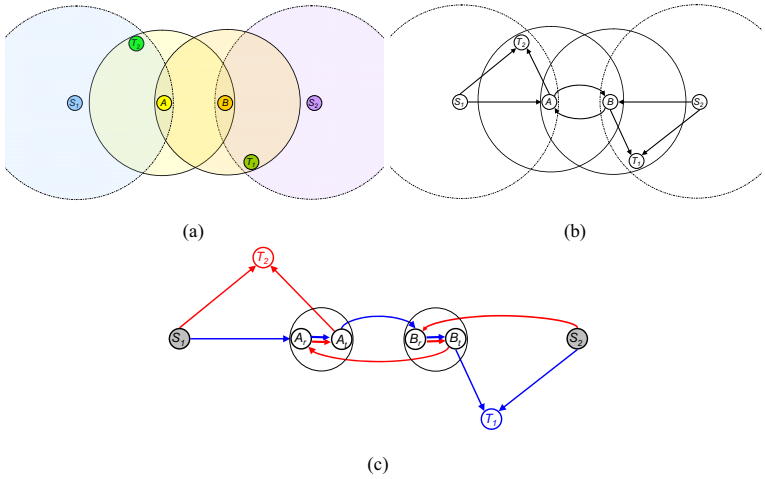


**Fig. 11:** *A wireless network coding with a cyclic solution. Here, the hearing distance is shown by circles.*

## 7.2 PAPER II

The second paper, entitled "Improved Delay Estimates for a Queueing Model for Random Linear Coding for Unicast," is co-authored by Ángela I. Barbero and Øyvind Ytrehus and was presented at *IEEE International Symposium on Information Theory* (ISIT2009) in Seoul, Korea.

Communication protocols based on coded packets, e.g. LT codes and network coding, have shown promising performance in terms of throughput. However, in a practical communication system, other features are of importance, depending on the applications. One of these features is the delay. For coding based communication, delay was mostly studied with the assumption that all packets are present at the source node before the start of the communication session. For the case where

[10]The procedure is explained in Section 5.3.

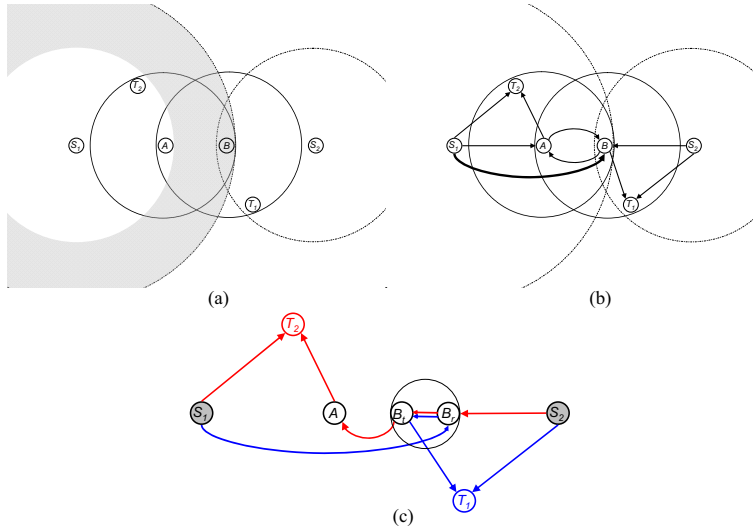


**Fig. 12:** *An acyclic solution to the wireless network coding problem. The grey region is the extra coverage required compared to the cyclic solution.*

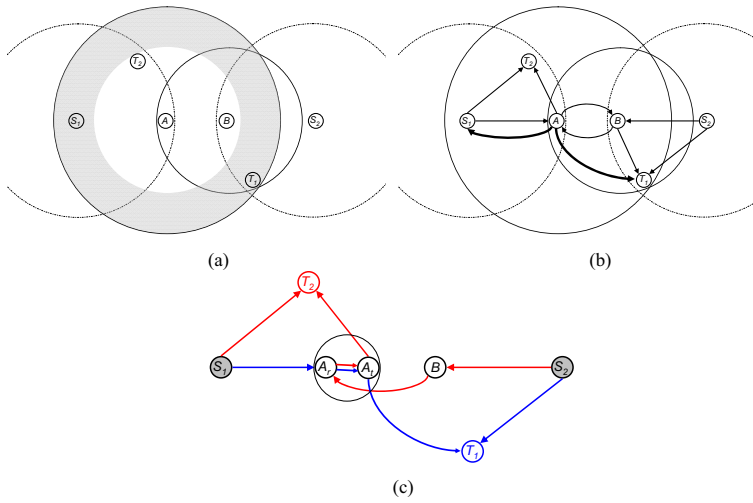


**Fig. 13:** *An acyclic solution to the wireless network coding problem. The grey region is the extra coverage required compared to the cyclic solution.*

packets arrive or are created at the source node based on a random pattern, there were not many studies. One step in this direction was taken in the paper by Shrader et al..

We observe, through simulations, that the computation of delay by the method proposed by Shrader et al. is rather crude estimation. In this paper, we analyze this problem and argue that the large gap is a result of bad measurement of the status of the queue in the sender node. We extend the model by Shrader et al. and propose an improved method for computation of the delay. Our computation gives results very close to simulation results, and shows the relation between bulk size and field size on the delay. This relation was not observed in the work of Shrader et al.

## 7.3 Paper III

The third paper, entitled "Wiretapping Based on Node Corruption over Secure Network Coding: Analysis and Optimization," is co-authored by Mehdi M. Hassanzadeh and Dag Haugland and was presented at the *Second International Castle Meeting on Coding Theory and Applications* (ICMCTA 2008) in Medina, Spain and published in *Lecture Notes in Computer Science* 5228, 2008.

In a type II wiretap channel, there are parallel communication links between the sender and the receiver, and some of these links are eavesdropped by a wiretapper. The goal is to use a coding scheme such that the wiretapper could not understand the secret message communicated between the sender and receiver unless enough number of links are listened to. Secure network coding is a generalization of this problem for multicasts.

In the secure network coding literature, two kinds of attacks were studied, nodewise and linkwise. In this paper, a new form of attack is studied, which is a generalization of the nodewise. Also, optimization algorithms for flow graph formation are proposed to improve the resistance against the attacks under study. Through simulations, the performances of the optimization algorithms are shown.

## 7.4 Paper IV

The forth paper is a joint work with Mehdi M. Hassanzadeh and Øyvind Ytrehus. It is entitled "Two Layer Secure Network Coding - (2-LSNC),"

and was presented at the *International Symposium on Telecommunications* (IST 2008),Tehran, Iran.

In this paper, the secrecy capacity of secure network coding is improved. We have proposed a secure network coding based on concatenation of secret sharing. By applying optimization techniques for flow formation, we show how a flow graph can be found for such a secure network coding that guarantees all the sink nodes to recover the secret message.

In the proposed method, more network resources are required for the network coding setup. However, we show, by introducing special cost-security metrics, that our scheme can provide a trade-off between cost per level of security. We also observed, through simulations, that our proposed method is more scalable compared to the ordinary (single layer) secure network coding, and it can improve the secrecy capacity.

### 7.5 Paper V

The fifth and last paper, entitled "Heuristic Methods for Flow Graph Selection in Integral Network Coding," is co-authored by Dag Haugland and is a technical report at *Department of Informatics, University of Bergen*, Bergen, Norway, (ISSN 0333-3590), Report Number:2009-391, 2009 and is a preprint of a paper that will be submitted for publication.

Because of the discrete nature of the network coding, there is a need for optimization algorithms that can produce integer flow graphs. In general, the problem is shown to be **NP**-hard. In this paper, we proposed two heuristics for integral flow network coding. The main feature of these algorithms is their low complexity, and their performance close to optimum. This has been observed by simulations in [31]. One of the algorithms proposed in this paper has the ability to not only provide an integer solution, but also to guarantee that the resulting subgraph is a directed acyclic graph. For the acyclic problem, it is not always expected to find a feasible solution even if there is one.

## 8 Future Research

- In **Paper I**, the broadcast property of wireless links is modeled, but no broadcast coding is considered. Therefore, the capacity for broadcast links is limited by the worst receiver. Since the knowledge on adapting network coding and broadcast coding is poorly understood, this can be a future topic for research.

- In **Paper I**, it is shown that in random geometric graphs, if we allow cycles, the solutions are on average more power efficient than if we do not allow cycles. One can conjecture that such improvement in the total transmission power can provide improvement in the overall interference experience, which in its term results in higher capacity for interference limited systems, e.g. CDMA. Study of this feature is a future work.
- In **Paper I**, deterministic network coding is considered. The effect of cycles for random network coding is not studied, and it would be interesting to analyze the performance of random network coding in cyclic networks.
- In **Paper V**, an algorithm for finding acyclic solutions is proposed. The performance of this algorithm is not theoretically studied and there is room for the study of other algorithms that can provide acyclic solutions.
- In optimization problems that consider interference, the final solutions are relaxed solutions. It would be a future research to investigate algorithms with integer solutions, e.g. **Paper I**, that can enter the parameter of interference as their input.
- In **Paper II**, the feedback channel is considered error/erasure and delay free. As a future work these parameters can be analyzed.
- There can be different settings for a packet delivery system, e.g. ARQ and FEC. As an extension to **Paper II**, the trade-offs between choosing different packet delivery systems and coding based systems is a future work.
- In **Paper III** and **Paper IV**, the proposed algorithms for secure network coding are studied up to the flow formation phase. The encoding process is considered to be done by well known encoding algorithms. As a future path to secure network coding, it could be studied how to select the coding in such a way that the secret message is not exposed at nodes other than sink nodes. The proposed algorithms are centralized algorithms and decentralized approaches can be investigated.
- In some wireless network coding applications, in order to adapt integral capacity of links with the broadcast property of wireless links a special graph model is required. A simple version for

this problem is introduced in [31]. As a more general case for future work, the following procedure that converts the wireless network into a graph with all unit rate links can be studied. In this procedure, the integrality and broadcast nature of the links is preserved.

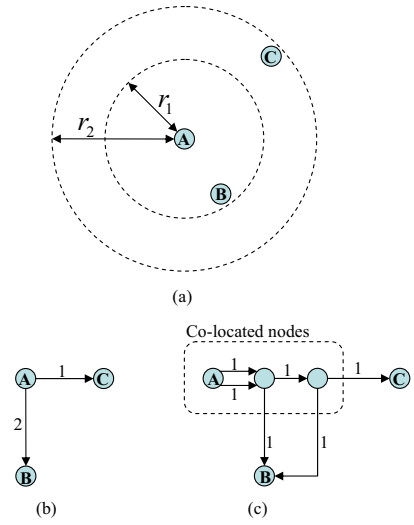


**Fig. 14:** *In (a) the inner region has two rate capacity at its border and the larger sphere has unit rate capacity. In (b) the equivalent graph with integer rate capacity links is shown. In (c) the equivalent all unit rate capacity links is given.*

In wireless broadcast networks, based on the factors that are involved in the capacity, concentric spheres that have an integer capacity on their borders are computed for each node and nodes around the transmitting node will be assigned an integer capacity link based on the smallest sphere that covers them. This process is shown in Figure 14(a) and (b). In Figure 14(c) it is shown how the resulting links can be modeled by all unit rate links. The conversion process for node A as shown in Figure 15 is the following. First, we compute $n$, the largest integer capacity link for the closest neighbor to node A. Second, we consider $n$ co-located nodes with node A and we call then $B_A^{(i)}$ for $i = 1 \cdots n$. Third, we connect A to $B_A^{(n)}$ with $n$ parallel unit rate links and we connect $B_A^{(j)}$ to $B_A^{(j-1)}$ by $j-1$ parallel unit rate links for $j = 2 \cdots n$. The final step is to connect neighbors of A to these nodes. For a neighbor with $m$

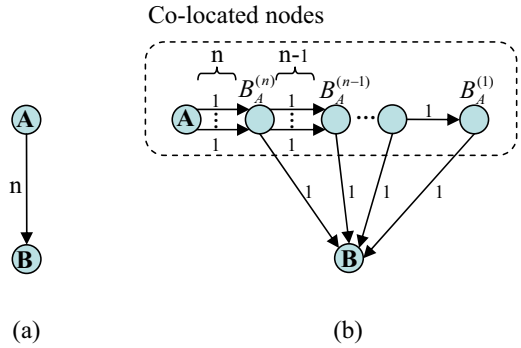


**Fig. 15:** *The procedure for converting integer capacity links into all unit rate capacity links. The integer link capacity in (a) is converted to all unit rate capacity links with adding extra co-located nodes in (b). In (b) co-located nodes with A are $B_A^{(i)}$ s from left to right.*

rate capacity link, we connect a unit rate link from $B_A^{(j)}$ to that node for $j = 1 \cdots m$.

Many algorithms are based on the unit rate assumption for links, and with this procedure it is possible to use those algorithms.

## REFERENCES

[1] N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory, 1736-1936*, Oxford University Press, 1986.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", *IEEE Transactions on Information Theory*, Vol. 46, April 2000, pp. 1204-1216.

[3] C.E. Shannon, "A Mathematical Theory of Communication", Bell System Technical Journal, 27, July & October, 1948, pp. 379-423 & 623-656.

[4] L. R. Ford, and D. R. Fulkerson, "Maximal Flow through a Network", *Canadian Journal of Mathematics*, 8 (1956), pp.399–404.

[5] P. Elias, A. Feinstein, and C. E. Shannon, "Note on maximum flow through a network", *IRE Transactions on Information Theory*, IT-2, pp. 117–199, 1956.

[6] B. Shrader, and A. Ephremides, "A queueing model for random linear coding", *IEEE Military Communications Conference*, October 2007, pp. 1-7.

[7] R. J. Vanderbei, "Linear Programming: Foundations and Extensions", *Boston: Kluwer*, 2001.

[8] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1", *IEEE International Conference on Communications*, vol.2, Geneva Switzerland, May 1993, , pp. 1064–1070.

[9] R. G. Gallager, *Low Density Parity Check Codes*, Research Monograph Series no. 21, Cambridge, M.I.T. Press, 1963.

[10] A. O. Allen, *Probability, Statistics and Queueing Theory with Computer Science Applications*, Second Edition, , Academic Press, 1990.

[11] K. Jain, M. Mahdian and M. R. Salavatipour, "Packing Steiner trees", *In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, Maryland, January 2003, pp. 266–274.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press and McGraw-Hill, 2001.

[13] R. Koetter and M. Medard, "An algebraic approach to network coding", *IEEE/ACM Transactions on Networking*, vol.11, no.5, Oct. 2003, pp. 782–795.

[14] M. Penrose, *Random geometric graphs*, Oxford Studies in Probability, 2003.

[15] P. Balister, B. Bollobás, A. Sarkar, and M. Walters, "Connectivity of random k-nearest neighbour graphs", *Advances in Applied Probability,* 37 (2005), 1-24.

[16] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks", *Wireless Networks*, vol. 10, Issue 2, March 2004, pp. 169–181.

[17] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial Time Algorithms for Network Information Flow", *Proc. SPAA'03,* San Diego, June 7-9, 2003, pp. 286-294.

[18] 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, June 2005, pp. 1973-1982.

[19] E. Erez and M. Feder,"On Codes for Network Multicas", *Proceedings of 41st Annual Allerton Conference on Communication Control and Computing*, October 2003.

[20] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, Shi Jun and B. Leong, "A Random Linear Network Coding Approach to Multicast", *IEEE Transactions on Information Theory,* Vol.52, October 2006, pp.4413-4430.

[21] T. M. Cover, "Comments on broadcast channels", *IEEE Transactions on Information Theory*, Vol. 44, Oct. 1998, pp. 2525-2530.

[22] Y. Xi and E. M. Yeh, "Distributed Algorithms for Minimum Cost Multicast with Network Coding in Wireless Networks," *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, April 2006.

[23] T. Cui and T. Ho, "Minimum Cost Integral Network Coding", *Proceedings of IEEE Symposium on Information Theory*, 2007.

[24] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-Cost Multicast over Coded Packet Networks", *IEEE Transactions on Information Theory*, Vol. 52, Issue 6, June 2006, pp. 2608– 2623.

[25] Á. Barbero and Ø. Ytrehus, "Cycle-logical Treatment of 'Cyclopathic' Networks", *IEEE Transactions on Information Theory,* Vol. 52, June 2006, pp. 2795-2805.

[26] Á. Barbero and Ø. Ytrehus, "Introduction to Network Coding for Acyclic and Cyclic Network", *to appear in "Selected Topics in Information and Coding Theory", World Scientific, I. Woungang (Ed.).*

[27] M. Luby, "LT Codes", *IEEE Symposium on the Foundations of Computer Science*, 2002, pp. 271-280.

[28] P. Minero, D. N. C. Tse and M. Franceschetti, "A Broadcast Approach to Random Access", *IEEE Information Theory Workshop*(ITW), Taormina-Italy, October 2009.

[29] G. Kramer, "Communication on Line Networks with Deterministic or Erasure Broadcast Channels", *IEEE Information Theory Workshop*(ITW), Taormina-Italy, October 2009.

[30] T. Ho, B. Leong, R. Koetter and M. Médard, "Distributed Asynchronous Algorithms for Multicast Network Coding", *Proceedings of 1st Workshop on Network Coding, WiOpt*, 2005.

[31] M. Ravanbakhsh, Á. I. Barbero, D. Haugland and Ø. Ytrehus,"Power savings of cyclic network coding for multicast on wireless networks", *IEEE Information Theory Workshop*(ITW), Cairo, Egypt, 2010.

[32] M. Ravanbakhsh and D. Haugland,"Heuristic Methods for Flow Graph Selection in Integral Network Coding", Reports in Informatics (ISSN 0333-3590), Dept. of Informatics, University of Bergen, Bergen, Norway, 2009:391.