Scalable Multimedia Communication using Network Coding

SCALABLE MULTIMEDIA COMMUNICATION USING NETWORK CODING

BY

MINGKAI SHAO, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2011) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE: Scalable Multimedia Communication using Network Coding

AUTHOR: Mingkai Shao

M.Sc., (Computer Science)

Wuhan University, Wuhan, P.R.China

SUPERVISOR: Dr. Xiaolin Wu, Dr. Jun Chen

NUMBER OF PAGES: xi, 87

Dedications

To my parents

for letting me pursue my dream for so long.

&

To my wife

for offering me unconditional love and support.

Abstract

This dissertation devotes itself to algorithmic approaches to the problem of scalable multicast with network coding. Several original contributions can be concluded as follows.

We have proved that the scalable multicast problem is NP-hard, even with the ability to perform network coding at the network nodes. Several approximations are derived based on different heuristics, and systematic approaches have been devised to solve those problems. We showed that those traditional routing methods reduce to a special case in the new network coding context.

Two important frameworks usually found in traditional scalable multicast solutions, i.e. layered multicast and rainbow multicast, are studied and extended to the network coding scenario. Solutions based on these two frameworks are also presented and compared. Surprisingly, these two distinctive approaches in the traditional sense become connected and share a similar essence of data mixing in the light of network coding. Cases are presented where these two approaches become equivalent and achieve the same performance.

We have made significant advances in constructing good solutions to the scalable multicast problem by solving various optimization problems formulated in our approaches.

In the layered multicast framework, we started with a straight-forward extension of the traditional layered multicast to the network coding context. The proposed method features an intra-layer network coding technique which is applied on different optimized multicast graphs. Later on, we further improved this method by introducing the inter-layer network coding concept. By allowing network coding among data from different data layers, more leverage is gained when optimizing the network flow, thus higher performance is achieved.

In the rainbow multicast framework, we choose uneven erasure protection (UEP) technique as the practical way of constructing balanced MDC, and optimize this MDC design using the max-flow information of receivers. After the MDC design is finalized, a single linear network broadcast code is employed to deliver MDC encoded data to receivers while satisfying the individual max-flow of all the receivers. Although this rainbow multicast based solution may sacrifice the performance in some cases, it greatly simplifies the rate allocation problem raised in the layered multicast framework. The use of one single network code also makes the network codes construction process a lot clearer.

Extensive amount of simulation is performed and the results show that network coding based scalable multicast solutions can significantly outperform those traditional routing based solutions. In addition to the imaginary linear objective function used in the simulation, the practical convex objective function and real video data are also used to verify the effectiveness of the proposed solutions. The role of different parameters in the proposed approaches are analyzed, which gives us more guidelines on how to fine-tune the system.

Acknowledgements

I would like to express my appreciation to all the individuals who have provided invaluable assistance to made the completion of this dissertation possible.

First of all, I would like to thank my supervisor, Dr. Xiaolin Wu, for introducing me to the exciting areas of multimedia communication, for giving me so much freedom to explore and discover and for providing insightful comments and suggestions.

I would like to thank my co-supervisor, Dr. Jun Chen, for helping me unveil the esthetics of information theory and see the beauty behind the formulas, and for those informative conversations we had when I met problems.

I would like to thank in particular, Dr. Sorina Dumitrescu, for showing me what a creative yet rigorous researcher should look like, for those talented ideas that shared with me, and for the numerous helpful discussions we had in the past few years.

Special thanks goes to Dr. Dongmei Zhao and Dr. Jie Liang, for reviewing this thesis and providing valuable inputs.

I would also like to thank my friends and colleagues at McMaster University with whom I have had the pleasure of working over the years.

Contents

List of Figures

Chapter 1

Introduction

How does one transmit a source data over a network from the server to a set of receivers to fully utilize available network resource and realize the best possible reconstruction of the data at the receivers?

The answer to the above problem is crucial to the network applications which involve a large amount of data transmission. This is also the question we try to investigate in this dissertation.

1.1 Applications and Motivations

Real-time multimedia communication spans a wide range of applications, including online video streaming, video conferencing, video on demand, voice over IP (VoIP), online gaming, surveillance modules and many more. With the multimedia applications becomig the major form of bandwidth consumption over Internet, efficient multimedia data delivery mechanism is of great importance not only to the service subscriber, but also to the service provider. Better multimedia data delivery mechanism not only brings better user experience but also means lower cost for the service provider.

The intrinsic multicast nature of multimedia traffic (from a source toward many destinations) arguably renders multicast one of the most important forms of multimedia applications on the Internet.

In this dissertation, we are especially interested in the scalable multicast. Unlike the unirate multicast which emphasizes delivering the same of amount of data to all the receivers, scalable multicast considers the scenario with heterogeneous subscribers, i.e. users with different network resources, and aims to transmit different amount of data to different receivers according to their resources. The "scalable" means receivers with different amount of received data can recover the source to different reconstructed quality. The more data received, the better reconstructed quality can be achieved. The network resources we referred to here can be the physical limitation such as capability of display hardware, network bandwidth or some virtual restriction such as subscriber priority (paid/unpaid).

Unlike the traditional way of considering optimized source coding and optimized data transmission separately, we jointly consider the source coding and network transmission problems. As we will show in the following section, this joint design usually achieves better performance in the scalable multicast scenario.

1.2 Separability of Source Coding and Network Transmission

To solve the data communication problem in a lossless network, we have to face two major challenges. First, the source coding problem, i.e. how to encode the source data such that a minimal amount of bits is needed for transmission. Second, the network communication problem, i.e. how to transmit the encoded data over the networks such that a maximal transmission rate can be achieved.

Ideally, a simple solution that solves the aforementioned two problems separately without sacrificing the overall system performance is preferred. In other words, when finding the optimal source coding scheme, we do not need to know which transmission scheme will be deployed, and vice versa.

In the unicast scenario where single server communicate with single receiver, it is clear that the source coding problem and network transmission problem are separateble. We can compress source data to its entropy [1,2] (in the lossless communication) or any rate point on its RD curve [3](in the lossy communication), then transmit these encoded bits using a transmission scheme which achieves the maxflow value [4] between the server and receiver. Furthermore, only routing (simple store-and-forward) is needed in the network communication step, as the max-flow min-cut theorem [5] proves that there is always a set of edge-disjoint paths achieving the maxflow value.

Unlike the unicast scenario that the seperability is clear for all the cases, the multicast scenario is more complicated, and there is no single unified answer for all the cases. For some cases the source coding and network transmission can be separately performed without sacrificing any performance compared to the joint consideration. However, it turns out that source coding and network communication are non-separable in more general scenarios, and the joint design of source coding and network transmission can achieve better performance.

1.2.1 Unirate Multicast Scenario

In the traditional multicast problem which emphasizes on delivering the same amount of information to all receivers, the source coding problem and the network communication problem can be solved separately without loss of any potential performance.

Considering a heterogeneous network where different receivers have different maxflow values from the server node, the maximum achievable common information multicast rate is limited by the minimum maxflow value among all the receivers. Similar to the unicast cases, to achieve the best performance, one can compress source data to its entropy [2] (in the lossless communication) or any rate point on its RD curve [3](in the lossy communication), then transmit these encoded bits at the transmission rate specified by the minimum maxflow value between the server and receivers. However, in the network communication step, the traditional routing technique is no longer sufficient, and the network coding technique is required to achieve the minimum maxflow rate to all receivers.

The essence of network coding is to enabling data processing at the intermediate nodes. Precisely, instead of simply relaying the data packets they receive, the intermediate nodes of a network will mix several data packets together for transmission. It has been proven in the network coding theory that the minimum maxflow value among all the receivers is guaranteed to be achieved in a common information multicast with network coding.

1.2.2 Scalable Multicast Scenario

The more general setting of multicast problem is that the data rates to receivers are not identical, i.e. different receivers may receive different amount of data. However, the nature of the problem changes drastically when we relax the equal rate constraint. In this case, a joint consideration of source coding and network communication is necessary. Fig. 1.1 is a step by step example of how source coding and network communication (routing and network coding) can become entangled, in a complex way.

Fig. 1.1.(a) illustrates the case where the source data X has to be communicated only to the nodes t_1, t_2. The max-flow into both nodes t_1 and t_2 is 2. Thus, as stated before, one can optimally encode X into a source code stream of rate 2, break the stream into two sub-streams x and y, each of rate 1 and communicate them to nodes t_1 and t_2. Note that network coding (i.e., bitwise XOR operation on streams x and y) is necessary at node u_4.

But, what if nodes t_1, t_2 and t_3 constitute the set of sink nodes, as shown in Fig. 1.1.(b)? In this case, the max-flow is 1 for node t_3 and is 2 for nodes t_1 and t_2. The better strategy now is to progressively encode X into a stream of rate 2. Take the first portion of this stream to make a stream x of rate 1 and make the rest of the stream into another stream named y, again of rate 1. Network coding should be used to get both x and y to nodes t_1 and t_2 and stream x to node t_3. Note that node u_1 gets the stream y only. Since this is the second portion of a progressively encoded source code stream, node u_1 will not be able to reconstruct X at all (but this is OK,

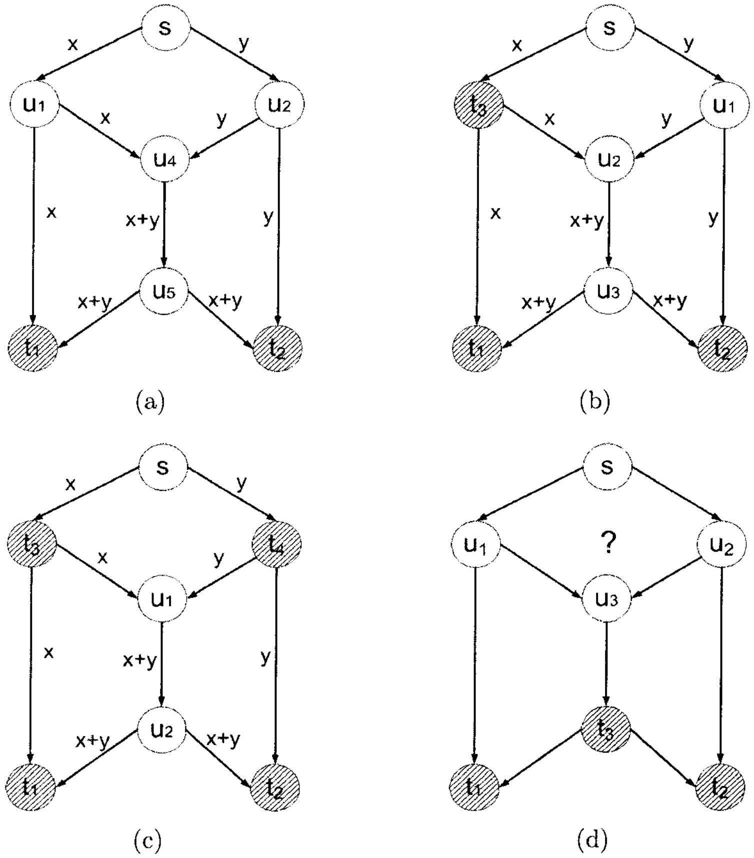


Figure 1.1: Optimal delivery strategy varies according to network structure

since node u_1 is not a sink node).

Fig. 1.1.(c) is yet another scenario, in which nodes t_1, t_2, t_3 and t_4 are sink nodes. In this case, a more suitable strategy is to encode X into two multiple description code (MDC) [6] streams each of rate 1 (name them streams x and y); then use network coding to communicate both x and y to nodes t_1, t_2. Node t_3 will receive only the description x and node t_4 will only receive y.

When nodes t_1 , t_2 and t_3 are sinks, for instance, none of the above strategies is necessarily optimal (Fig. 1.1.(d)).

As Fig. 1.1 suggest, in more general multicast scenario where receivers are not required to receive the same amount of data, a joint consideration of source coding and network communication is a must. Even if it is possible to break (without loss of generality) the task into a proper concatenation of source coding operations (e.g., progressive coding or MDC), followed by network communication techniques (e.g., network coding or routing). This breakdown, however, is not done blindly. In another word, to achieve the best possible performance, one should choose the network communication strategy in accordance with the source coding technique adopted, and vice versa.

1.3 Related Topics

1.3.1 Relation to Network Coding

Network coding, a new powerful paradigm of network communication, can greatly improve throughput over traditional routing. The essence of network coding is the provision of multiple paths and the coding ability at intermediate nodes which enable information flows for different receivers to share the common network capacity [7]. It has been proved that network coding can achieve the minimum of individual max-flow values in the unirate multicast scenario, where all the receivers demand for the same amount of information [7].

Linear network coding refers to network coding scheme in which the encoding functions at each node is linear, i.e. the outgoing data of a certain node is the linear combination of data from the incoming links. Li et al. [8] showed that linear network coding is sufficient to achieve the unirate multicast capacity. Koetter and Medard gave an algebraic characterization for a linear network coding scheme in [9]. They also gave an upper bound on the field size and a polynomial time algorithm to verify the validity of a network coding scheme. A polynomial time algorithm to construct the optimal unirate multicast network code is given in [10].

However, due to a decodability issue, network coding is less straightforward in the multirate multicast scenario. Unlike unirate multicast where network codes are guaranteed to exist and are easy to construct, multirate multicast network codes do not necessarily exist for the desired data rates of the receivers. Algebraic conditions for the existence of multirate multicast network code are derived in [9], and finding the optimal network code for a given network, which maximizes the total amount of information flow, is proven to be NP-hard.

Random network codes are linear network codes in which the encoding coefficients are chosen randomly from a finite field. The sink nodes can decode correctly if and only if the overall transfer matrix from the sources to each sink is invertible.

Random network coding was first described in [11], which gave, for acyclic delay-free networks, a bound on error probability, in terms of the number of receivers and random coding output links, that decreases exponentially with code length. The proof was based on a result in [12] relating algebraic network coding to network flows. The result was improved later with a success probability bound for randomized network coding in link-redundant networks with unreliable links, in terms of link failure probability and amount of redundancy.

Compared to the deterministic network coding, random network coding only provides asymptotic optimality for the unirate multicast case. The performance bound for the multirate multicast case is generally unknown. Since the main focus of the work is to achieve the highest throughput or best reconstructed quality as possible by using network coding technique, to avoid ambiguity and uncertainty, we confine our discussion only to the deterministic network coding. Actually, those solutions we introduce in the following chapters using unirate network coding can be immediately generalized to incorporate random network coding.

1.3.2 Relation to Linear Broadcast in Network Coding

At the very beginning of network coding research, most of the works dealt with common information multicast. Namely, the design objective is for all sinks to realize the maximum same data rate. This setup makes sense if all sinks need to achieve lossless decoding of the same message, but it is quite limiting for scalable signal network communications. Later, some of the pioneer researchers of network coding generalized the earlier work to include three types of linear network codes: linear multicast, linear broadcast and linear dispersion [13]. According to the definition, a w-dimensional linear network code is said to be a linear broadcast if the dimension of the received information of each receiver equals to the minimum value of w and the max-flow of that receiver ($dim(V_T) = min\{w, maxflow(T)\}$ for every sink node T) [13]. The authors proved the existence of the linear broadcast network code. By choosing sufficiently large w ($w \geq maxflow(T)$ for all sink T), one can find a network code that the dimension of linear independent information received by each receiver achieves its max-flow value.

In the above view, the linear broadcast problem appears similar to the scalable multicast problem, since both of them emphasize on delivering different amounts of information to receivers with different network resources. Moreover, the answer

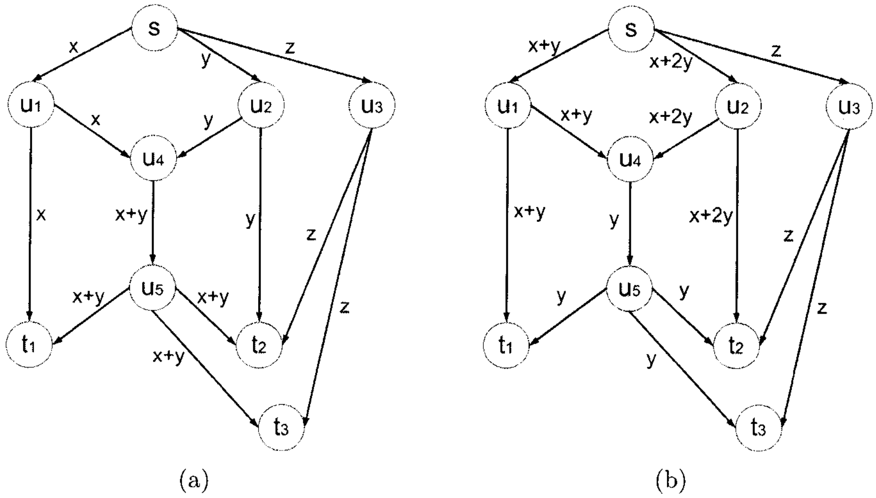


Figure 1.2: (a) Sink node t_3 can not decode all of its received descriptions, (b) by transmitting linear combinations from the source, sink nodes t_1, t_2 and t_3 can decode all of the received descriptions.

to the scalable multicast problem seems to be straightforward given that the linear broadcast theory states that each receiver can definitely achieve its max-flow value.

However, the dimension of linear independent information received by a receiver does not always imply the same number of distinct messages which are helpful for reconstructing original data. There are such situations that the messages received by a receiver are linearly undecodable. This will never happen in the linear multicast because the dimension of received information always equals to the code dimension w. But linear broadcast allows the dimension of received information for a receiver to be less than the code dimension w, thus leads to a possible linearly undecodable situation (as shown in Fig. 1.2(a)).

Fig. 1.2(a) shows a linear broadcast code. In this network, node s is the source node, and nodes t_1, t_2, t_3 are three sinks. All edges in the graph have unit capacity. Now supposing the dimension of the linear broadcast code w to be 3, the source node transmits three linear independent data messages x, y, z, and the dimension of the received information of each receiver achieves its own maximum flow value (2, 3, 2). However, we observe that although sink node t_3 receives two messages, i.e. $x+y$ and z, it can only decode message z. Since node t_3 can decode neither x nor y, the message $x+y$ makes no contribution to the reconstruction fidelity at receiver t_3. Thus, the actual numbers of *effective descriptions* received by the three receivers are (2, 3, 1) in this case.

In certain graphs, we can avoid the undecodable situation by letting the source node(s) transmit the linear combination of original data, i.e. performing network coding at source node(s). But this technique is not a general remedy, it can fail on some graphs. For example, Fig. 1.2(b) shows a network code that achieves max-flows if nodes t_1, t_2, and t_3 are the only sinks (the same as in Fig. 1.2(a)). But, if nodes u_1, u_2 and u_3 are also sinks in that network, then the network code in Fig. 1.2(b) cannot realize the max-flow values of u_1 and u_2. In order for sink nodes u_1 and u_2 to decode, they must receive single description, which in turn leads to the situation in Fig. 1.2(a) where not all three sink nodes t_1, t_2 and t_3 can achieve their max-flow.

Clearly, the dimension of the received information is not a suitable performance metric for many applications with scalable signal compression. Therefore, the scalable multicast problem is formulated with the objective of optimizing for the amount of actually decodable information.

1.3.3 Relation to Inter-session Network Coding

An extension of unirate multicast network coding is to apply network coding to multiple concurrent unirate multicast sessions, which is called inter-session network coding.

In [14] the authors showed that linear network coding is insufficient to achieve multicast capacity for multiple multicast sessions. Li and Li [15] showed that there is no coding gain for an undirected graph. Some preliminary work on inter-session network coding for the case with only two simple multicast sessions is given in [16]. The authors of [16] gave the condition under which there exists a linear network coding scheme for two multicast sessions. Wu [17] applied random network coding to all the sessions on a transformed network topology such that the source can only reach the receivers that are interested in the source. In [18], the authors proposed an optimization method which maximizes the inter-session network coding gain according to two metrics: overlap ratio and overlap width.

The differences between inter-session network coding and the network coding scalable multicast rely on the dependency of data transmitted among different sessions. Inter-session network coding focuses on the cases that the data transmitted in different network coding session are mutually independent. While in the scalable multicast, a single multicast session is considered. Although some of the scalable multicast solutions do divide the single scalable multicast session into multiple unirate multicast sessions, the data transmitted at different sessions is usually dependent and the data dependance among different sessions should be carefully considered.

1.3.4 Relation to Scalable Multicast using Fountain Codes

Fountain codes [19] are a class of random erasure codes with the property that a potentially limitless sequence of encoding symbols can be generated from a given set of source symbols. The original source symbols can ideally be recovered from any subset of the encoding symbols of size equal to or only slightly larger than the number of source symbols. The fountain codes are also called rateless erasure codes due to the fact that these codes do not exhibit a fixed code rate.

Fountain codes are known that have efficient encoding and decoding algorithms and that allow the recovery of the original N source symbols from any $N(1+\epsilon)$ of the encoding symbols with high probability, where the small value ϵ represents the coding overhead.

Many practical realizations of fountain, i.e. LT codes and Raptor code, are proposed and used to scalable multicast applications [20–22]. For example, in [20] the authors proposed an application of sliding window Raptor codes on scalable video coding (SVC) in a lossy networks. SVC layers are encoded independently of each other using a SW-Raptor code, and the rates of SW-Raptor code for SVC layers, as well as the number of coded packets generated for each layer, are optimized so as to yield the best possible expected quality at the receivers. [21, 22] design and use the optimal expanding window fountain (EWF) code to multicast scalable multimedia content through a lossy networks, such that the given quality-of-service requirements for different receiver classes are satisfied. The essence of these approaches resembles the idea of uneven erasure protection (UEP), with the difference that EWF codes allow overlap among different priority levers.

The fundamental difference between these works and ours relies in the fact that the fountain codes based schemes are deployed to fight against packet loss rather than increase network throughput. These schemes usually require predefined bandwidth information for the receivers instead of optimizing these parameters for a certain network topology. Besides, all the coding is done at the server (the source node), thus no network coding is performed.

1.4 Contributions

This dissertation devotes itself to algorithmic approaches to the problem of scalable multicast with network coding. Several original contributions can be concluded as follows.

We have proved that the scalable multicast problem is NP-hard, even with the ability to perform network coding at the network nodes. Several approximations are derived based on different heuristics, and systematic approaches have been devised to solve those problems. We showed that those traditional routing methods reduce to a special case in the new network coding context.

Two important frameworks usually found in traditional scalable multicast solutions, i.e. layered multicast and rainbow multicast, are studied and extended to the network coding scenario. Solutions based on these two frameworks are also presented and compared. Surprisingly, these two distinctive approaches in the traditional sense become connected and share a similar essence of data mixing in the light of network coding. Cases are presented where these two approaches become equivalent and achieve the same performance.

We have made significant advances in constructing good solutions to the scalable multicast problem by solving various optimization problems formulated in our approaches.

In the layered multicast framework, we started with a straight-forward extension of the traditional layered multicast to the network coding context. The proposed method features an intra-layer network coding technique which is applied on different optimized multicast graphs. Later on, we further improved this method by introducing the inter-layer network coding concept. By allowing network coding among data from different data layers, more leverage is gained when optimizing the network flow, thus higher performance is achieved.

In the rainbow multicast framework, we choose uneven erasure protection (UEP) technique as the practical way of constructing balanced MDC, and optimize this MDC design using the max-flow information of receivers. After the MDC design is finalized, a single linear network broadcast code is employed to deliver MDC encoded data to receivers while satisfying the individual max-flow of all the receivers. Although this rainbow multicast based solution may sacrifice the performance in some cases, it greatly simplifies the rate allocation problem raised in the layered multicast framework. The use of one single network code also makes the network codes construction process a lot clearer.

Extensive amount of simulation is performed and the results show that network coding based scalable multicast solutions can significantly outperform those traditional routing based solutions. In addition to the imaginary linear objective function used in the simulation, the practical convex objective function and real video data are also used to verify the effectiveness of the proposed solutions. The role of different parameters in the proposed approaches are analyzed, which gives us more guidelines on how to fine-tune the system.

1.5 Organization of This Dissertation

The rest of this dissertation is organized as follows.

In Chapter 2, we introduce and discuss some of the previous solutions to the scalable multicast problem first. Then, our methodology to solve the scalable multicast problem is described, followed by a formal definition of the target problem on which many of the discussions in the rest of the dissertation relies. Finally, the complexity analysis of the target problem is presented.

In Chapter 3, we try to solve the scalable multicast problem in a layered multicast framework. Starting from a straight-forward way to extend the traditional layered multicast to the network coding scenario, we provide a scalable multicast solution based on the intra-layer network coding. This solution divides source data into different data layers and a separate network code is applied on each data layer.

Chapters 4 solve the target problem in a different way. A rainbow multicast framework is employed instead of the layered multicast framework used in Chapter 3. Uneven erasure protection (UEP) technique is chosen as the practical way of constructing balanced MDC. A single linear network broadcast code is employed to deliver the optimized MDC encoded data to receivers while satisfying the individual max-flow of all the receivers. The differences and similarities between this rainbow multicast approach and the aforementioned layered multicast approach are discussed.

In Chapter 5, the layered multicast approach presented in Chapter 3 is reviewed, with its drawbacks being pointed out. Based on those analysis, we introduce the inter-layer network coding concept, and further improve our optimization formulation by enabling data combination among different data layers. An efficient network code construction algorithm is also proposed to deliver the inter-layer optimized flow.

In Chapter 6, an extensive amount of experiments are carried out to verify the effectiveness of the proposed approaches. The role of different parameters in the proposed approaches is analyzed in both imaginary and practical setting.

Chapter 7 summaries this dissertation and suggests some of the possible future directions.

Chapter 2

Scalable Multicast Problem: Review and New Perspective

Scalable multicast problem has been studied for many years, and many solutions have been proposed. In this chapter, we will first review some of the existing scalable multicast solutions. Then, our new approach of jointly designing scalable source coding and network coding will be discussed.

2.1 A Review of Previous Solutions

A common approach to achieve scalable multicast is layered multicast. A multimedia source is encoded into a sequence of progressively refinable layers $L_1, L_2, \cdots, L_M$, a base layer and several successive enhancement layers. Each layer n may be used to increase the quality of the data reconstruction, but only if all previous layers $1, \cdots, n-1$, are available as well. Thus, a receiver desires to receive as many as possible consecutive layers $1, \cdots, n$, not just any subset of layers. Therefore, for such applications, the design objective is to maximize the network throughput while also ensuring that each sink receives only consecutive layers of data (including the base layer). The additional constraint imposed is due to the layered characteristic of source data. For this reason we refer to this problem setting as layered multicast.

In traditional networks where only routing is allowed, each node can only send copies of received messages. Therefore, the transmission of data layers is naturally separated into transmission layers (or multicast layers). Each multicast layer will send a layer of data to all designated receivers over a single multicast tree. Therefore the solutions to this problem [23–25] focus on the construction of multiple multicast trees, each representing a multicast layer. Each receiver can subscribe to one or more multicast tree(s), and the more multicast trees a receiver subscribes to, the better multicast quality it can achieve. Still, the progressive decoding constraint is applied on the subscribed multicast trees.

Another approach to deliver scalable multicast is rainbow multicast [26,27]. In the rainbow multicast, source data are encoded using multiple description code (MDC). MDC encodes source data into independently decodable descriptions, and the receiver is able to reconstruct the source given any subset of these descriptions. And the more descriptions are available at the decoder, the better reconstruction it can get. The objective of rainbow multicast problem is to maximize not the total number of packets but rather the total number of distinct packets received by one or more clients. The problem is therefore called rainbow multicast which carries an intuitive connotation: distinctively color the MDC descriptions and optimize the network flows to achieve the rainbow effect of getting as wide a spectrum of colors as possible at the sinks. In contrast to the layered multicast approach with the progressive decoding constraint, rainbow multicast features the freedom in optimizing the network throughput without worrying which data segments are received.

Network coding, the new promising paradigm of network communication, is shown to be able to greatly improve throughput over traditional routing. However, due to the decodability issue as discussed in Chapter 1, network coding is less straightforward in the multirate multicast scenario. Unlike unirate multicast where network codes are guaranteed to exist and are easy to construct, multirate multicast network codes do not necessarily exist for the desired data rates of the receivers.

Despite of the difficulty of the problem, many layered multicast schemes using network coding have been proposed to improve the performance of traditional routing based layered multicast. In general, they divide the network into different layers and construct a unirate multicast network code for each layer. However, these schemes do not perform network coding between data layers, and consequently cannot realize the full potential of network coding.

A number of previous works applied networking coding in the layered multicast setting [28–30], demonstrating substantial improvement in the bandwidth efficiency over traditional methods. As in the network coding-free scenario, these methods transmit each layer of data in a single multicast layer. But instead of constructing different multicast trees as in the traditional approach, network coding-based layered multicast divides the network graph into multicast sub-graphs according to certain criteria and determines the optimal amount of information transmitted on each multicast graph (i.e., the size of each data layer). Each multicast layer is considered as a session, and network coding is performed on different layers separately. Since the rates for different receivers in the same layer are equal, it is easy to construct the network code for a single layer using the existing polynomial-time algorithm [10]. Specifically, in [28], sinks are grouped into subsets $T_1, T_2, \cdots, T_N$, such that all sinks in the same subset have the same max-flow value and moreover, the max-flow value of any sink in T_k is smaller than the max-flow value of any sink in T_{k+1}. Then for each layer k a multicast sub-graph containing all sinks in $T_k \cup T_{k+1} \cup \cdots \cup T_N$ is constructed. Thus, the scheme ensures that all sinks in T_k receive k layers of data. The rate allocation between data layers (i.e., the size of each data layer) is decided by solving a linear programming problem. In [29], the authors resort to layered multicast and session scheduling ideas to provide rate control in scalable multicast. In [30], the authors proposed a similar layered network coding based solution with a centralized deterministic algorithm as well as a distributed heuristic algorithm. The heuristic algorithm organizes the receivers into layers progressively. Each receiver can subscribe to a number of layers to maximize its throughput according to its available bandwidth.

Independently, Wu [31] proposed a cross-layer network coding technique which is closely related to the inter-layer network coding approach presented in this dissertation. However, the problem formulation in [31] uses the heuristic that sink nodes with the same max-flow value are partitioned into the same group, which is included in our formulation as a special case. Also, we proposed a deterministic polynomial-time algorithm with guaranteed optimality for network code construction, whereas the scheme in [31] uses simple random linear mixing of asymptotical optimality.

2.2 Our Methodology

In this dissertation, we try to solve the scalable multicast problem from a network coding perspective. Two traditional scalable multicast frameworks, i.e. layered multicast and rainbow multicast, are analyzed and extended to a network coding scenario.

Surprisingly, these two distinctive approaches in the traditional sense become related and share a similar essence of data mixing in the light of network coding. For some cases, these two approaches even become equivalent and achieve the same performance.

2.3 Problem Formulation

Consider a directed acyclic network $G = (V, E)$, a set of source nodes S, and a set of sink nodes T. The multimedia source is scalably encoded into source segments (or messages) $x_1, \cdots, x_M$, of equal size. A source segment is an indivisible flow unit, and all data messages $x_1, \cdots, x_M$ are available at all source nodes, for transmission during a transmission slot. Therefore, the edge capacities are expressed in terms of number of source segments during the transmission slot.

The source segments are grouped into data packages (either with or without redundancy) for transmission. For simplicity, we assume the size of the redundancy data (if there is any) in a data package is also a multiple of the size of single source segment. A transmission scheme using network coding allows each data package sent along an edge to be a function of the data package received at the node where the edge originates.

Define a transmission scheme using network coding achieving rate $R(t)$ (number of source segments) at each sink t, as a scheme which ensures that each sink t can recover the first $R(t)$ source segments (out of the M source segments available at the source nodes for transmission during the transmission slot) after decoding the received messages. We assume a non-decreasing fidelity function $\phi(R)$ is given, representing the fidelity of the reconstruction after decoding the first R messages. The simplest example is $\phi(R) = R$. Other examples of fidelity functions, meaningful in multimedia applications, are PSNR, SNR, or the negative distortion. The problem we address in this dissertation is finding a data packaging and designing a linear network code such that $\sum_{t \in T} \phi(R(t))$ is maximized.

Since the main purpose of this dissertation is to establish that higher throughput is possible, we do not consider the transmission delay of the network links. Also, we assume that the nodes in the networks have buffers large enough to store received data.

Without loss of generality, the following development is confined to the case of a single source node. The case of multiple sources can be converted into one with a single source by adding a super source node and connecting the super source node to all source nodes by edges of infinite capacity. This conversion is depicted in Figure 2.1.

2.4 Complexity Analysis

Theorem 1. Finding the optimal solution of the scalable multicast problem is NP-hard even when there is a single server node, the underlying topology is a directed acyclic graph (DAG) and network coding is allowed.

The proof is constructed by reducing the well known NP-complete problem [32,33] of graph k-colorability [34, 35] to a special instance of the decision version scalable

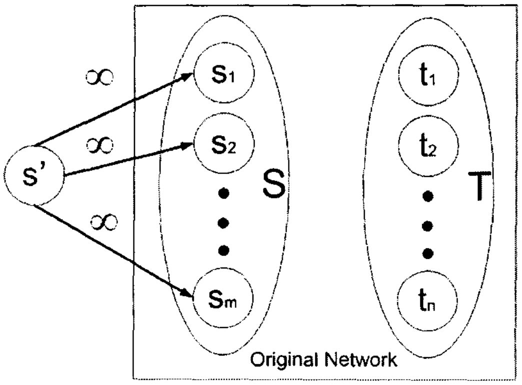


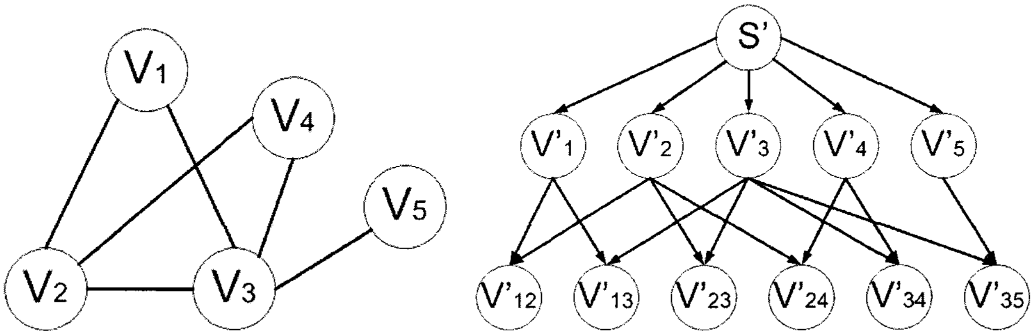
Figure 2.1: Converting a problem with multiple sources to one with a single source.

multicast problem on a DAG with only a single server node.

Given a k-colorable graph $G = \langle V, E \rangle$, we create an instance of scalable multicast problem, which is a directed acyclic graph $G' = \langle V', E', S', T' \rangle$. The reduction is carried out as follows.

1. Initialize $V' = E' = S = T = \phi$;
2. add a dummy source node s^* to V' and let $S' = \{s^*\}$;
3. for each vertex $v \in V$, add a v' to V' and add an edge (s^*, v') to E';
4. for each edge $(u, v) \in E$, add a vertex v'_{uv} to V' and T', then add edges (u', v'_{uv}), (v', v'_{uv}) to E'
5. set the capacity of all edges in E' to be 1;
6. let the total number of data segment to be k.

The above reduction can be done in polynomial time. Fig. 2.2 shows an example of this reduction.



(a) k-colorability problem (b) Equivalent scalable multicast

Figure 2.2: The reduction from an instance of k-colorability problem (a) to an instance of scalable multicast problem (b).

Lemma 1. The graph $G\langle V, E\rangle$ is k-colorable if and only if the transformed scalable multicast problem has a solution such that the total number of received decodable data of the nodes in T' equals to the sum of the number of incoming edges of T'.

Proof. $\Rightarrow$: Assume the graph $G = \langle V, E\rangle$ is k-colorable by function f: $V \to \{1, 2, \cdots, k\}$. For each $v \in V$, let $v' \in V'$ have the source segment i from s' through the edge (s', v'), if $f(v) = i$. Then for each $v' \in V'$, the only source segment is transmitted along all outgoing edges. Note that each vertex $t' \in T'$ corresponds to an edge in G. The two incoming edges must carry different source segments to t', otherwise the assumption is contradicted. Therefore, the incoming edges of each sink in T' deliver distinct source data segments to that sink, and it is obvious that the resulting solution is optimal for the transformed scalable multicast problem.

$\Leftarrow$: Assume that in the transformed scalable multicast problem, the total number of distinct source data segments delivered to the sink nodes equals to the sum of the number of the incoming edges of T'. For this to hold, all incoming edges of each sink must deliver distinct source data segments. Since each $v' \in V'$ only receives one source segment from s', we just use that source segment to color the corresponding vertex in G. Thus the graph G is k-colorable.

Note that there is only one source segment available at each $v' \in V'$, so the ability of network coding does not change the solution. □

The decision version of the scalable multicast problem is to determine whether the optimal value of the sum of received source data segments equals to an arbitrary integer n. From Lemma 1, it is clear that the decision version of the scalable multicast problem is in NP. Therefore Theorem 1 follows.

Since the target problem is NP-hard, we can only refer to heuristic solutions. In the following chapters, two important frameworks usually found in traditional scalable multicast solutions, i.e. layered multicast and rainbow multicast, are studied and extended to the network coding scenario. Surprisingly, these two distinctive approaches in the traditional sense become related and share a similar essence of data mixing in the light of network coding.

Chapter 3

Scalable Multicast using Intra-layer Network Coding

Starting from this chapter, we will look at several linear network coding based scalable multicast schemes. All of these schemes share the same key idea: make the full usage of network bandwidth by delivering linear combinations of source data to heterogeneous receivers.

We start our exploitation of network coding based scalable multicast approaches with a straight-forward extension of the layered multicast to the network coding scenario. The transmission is still divided into layers, but network coding is applied on each layer instead of simple routing.

3.1 Intra-layer Network Coding Multicast

As we discuss in the previous chapter, traditional layered multicast solutions divide multicast transmission into different layers, and a multicast tree is constructed for each multicast layer. All the leaves in a multicast tree receive the same amount of data within that multicast layer. However, the multicast tree structure is no longer suitable for network coding scenario due to the inherent property of network coding technique. The main advantage of network coding over the traditional routing lies in the ability of coding at intermediate nodes, which can "share" conflict links used by different receivers. Apparently, the single path multicast tree can not provide any leverage for network coding to do data mixing.

In order to exploit the full potential of network coding, the underlying network structure on which network coding is applied must have multiple paths from the source node to sink nodes. Therefore, we still divide the multicast transmission into different layers, but a multicast graph rather than a multicast tree is constructed for each multicast layer where network coding is applied. Considering that the multirate network coding problem is still open and there is currently no good way to construct multirate network codes with guaranteed decodability of the received packets, we use the well-studied uni-rate network codes in each multicast graph.

Therefore, the proposed method consists of two major steps. First, a flow optimization problem is solved to determine the optimal rate allocation among different layers and the multicast graph of each multicast layer. Then, the standard uni-rate network codes construction algorithm is run for each layer to construct the corresponding network codes.

Since the proposed method only applies network coding within each multicast layer, and there is no interaction of the source data among different multicast layers, we call this method scalable multicast using intra-layer network coding.

3.1.1 Problem Reformulated

Given the layered multicast nature of the method discussed in this chapter, we can re-formulate the target problem to a more precise one.

We define a layered multicast code achieving rate $R(t)$ (number of source segments) at each sink t, as a transmission scheme using network coding which ensures that each sink t can recover the first $R(t)$ source segments (out of the M source segments available at the source nodes for transmission during the transmission slot) after decoding the received messages.

We partition the set of data sequences into several layers, and each sink subscribes a certain number of layers. Note that, due to the property of scalable source coding, only the base layer and the following consecutive layers can contribute to the reconstruction fidelity. By grouping together the sink nodes which receive the same data flow layers we get a partition of sink nodes $T_1, \cdots, T_N$, such that $R(t) = R(t')$ for any t, t' in the same subset T_k, and $R(t) < R(t')$ for any $t \in T_k$, $t' \in T_{k+1}$ and any k. Let R_k denote the common value of $R(t)$ for the sinks $t \in T_k$. Define the k-th data layer as the set of source messages $\{x_{R_{k-1}+1}, \cdots, x_{R_k}\}$. Clearly, the layered multicast code guarantees that any sink in T_k receives the first k data layers. Then the above problem can be reformulated as follows.

Problem 1. Find the partition $\mathcal{T} = \{T_1, \cdots, T_N\}$ (where N is also a variable), the values $0 < R_1 < R_2 < \cdots < R_N$, and a layered multicast code achieving rate R_k at each sink $t \in T_k$, for each k, such that $\sum_{k=1}^{N} \sum_{t \in T_k} \phi(R_k)$ is maximized.

3.1.2 Proposed Solution

Since the problem is proven to be NP-hard, we resort to some heuristic in order to simply the problem. A simplification of the problem is to impose a fixed partition $\mathcal{T}$ of the sinks and find the optimal rate allocation corresponding to $\mathcal{T}$. Intuitively, the number of layers received by each sink should be proportional to the sink's max-flow value. This intuition motivates choosing for $\mathcal{T}$ the partition induced by the max-flow values, i.e., where all sinks in the same subset have the same max-flow value and moreover, the max-flow value of any sink in T_k is smaller than the max-flow value of any sink in T_{k+1}. We will denote this partition by $\mathcal{T}_{\text{max-flow}}$.

Having the partition of sinks specified, our target problem can be formulated as follows.

Problem 2. Given a partition $\mathcal{T} = \{T_1, T_2, \cdots, T_N\}$ of the set of sinks, find the values $0 \leq R_1 \leq R_2 \leq \cdots \leq R_N$ and a layered multicast code achieving rate R_k at each sink $t \in T_k$, such that $\sum_{k=1}^{N} \sum_{t \in T_k} \phi(R_k)$ is maximized.

Remark 1. By allowing equality between consecutive rates R_k and R_{k+1} in the formulation of Problem 2, we actually perform a search over all sink partitions obtained from $\mathcal{T}$ by cumulating consecutive subsets. (The equality $R_k = R_{k+1}$ means that T_k and T_{k+1} are merged into a single subset).

The proposed solution to Problem 2 consists of two major steps. First an intra-layer flow optimization problem is solved to decide the optimal rate allocation among different layers. Then, the standard uni-rate network code construction algorithm is run for each layer to construct a network code for each layer. The uni-rate network code construction algorithm is well established, we will skip it and only discuss the flow optimization part in the rest of the chapter.

3.2 Flow Optimization for Intra-layer Network Coding

For every node $v \in V$, let $In(v)$ denote the set of incoming links to v and let $Out(v)$ denote the set of outgoing links from v. $C_{i,j}$ is the capacity of edge (i, j), and s is the source node.

We divide the flow into N layers. Any sink $t \in T_k$, for some k, may receive flow in the first k layers with the requirement that the total amount of flow received over the first k layers equals R_k. Let $x_{i,j}^{t,l}$ be the flow on edge (i, j) for sink t in layer l. Define $b_j^{t,l}$ to be the potential of node j for sink t in layer l, which is defined as the difference between the incoming flow and the out-going flow. Negative node potential indicates a supplying node, while positive node potential indicates a demanding node. Let $y_{i,j}^k$ be the actual flow on edge (i, j) in layer k (over all sinks). For each sink node t, let $L(t)$ denote the number of data layers that sink t will receive (i.e., $L(t) = k$ if and only if $t \in T_k$). Then, the flow optimization problem can be formulated as shown in Figure 3.1.

Constraint (3.1c) follows from the definition of node potential. (3.1d) is the potential constraints at the source node. The total flow sent out from the source s to sink t over each layer should equal $R_{L(t)}$.

Constraint (3.1e) concerns the potential of sink nodes. The flow received in each layer should equal the flow sent by the source over the same layer.

$$
\begin{aligned}
\max \quad & \sum_{t\in T} \phi(R_{L(t)}) && \text{(3.1a)}\\
\text{subject to} \quad & R_1 \le R_2 \le \cdots \le R_N && \text{(3.1b)}\\
& b_j^{t,l} = \sum_{(i,j)\in In(j)} x_{i,j}^{t,l} - \sum_{(j,h)\in Out(j)} x_{j,h}^{t,l} && \text{(3.1c)}\\
& R_j = -\sum_{i=1}^{j} b_s^{t,i}, \quad \forall t\in T, 1\le j\le L(t) && \text{(3.1d)}\\
& \sum_{i=1}^{j} b_t^{t,i} = -\sum_{i=1}^{j} b_s^{t,i}, \forall t\in T, 1\le j\le L(t) && \text{(3.1e)}\\
& \sum_{i=1}^{j} b_n^{t,i} = 0, \forall t\in T, 1\le j\le L(t), n\notin\{s,t\} && \text{(3.1f)}\\
& y_{i,j}^{l} = \max_{t\in T}\{x_{i,j}^{t,l}\}, \quad \forall l \quad 1\le l\le N && \text{(3.1g)}\\
& \sum_{l=1}^{N} y_{i,j}^{l} \le C_{i,j}, \quad \forall (i,j)\in E && \text{(3.1h)}\\
& x_{i,j}^{t,l} \text{ is non-negative integer}, \forall t\in T, \forall (i,j)\in E && \text{(3.1i)}
\end{aligned}
$$

Figure 3.1: Inter-layer flow optimization.

(3.1f) is the constraint for the potential at the intermediate nodes. The total flow received by an intermediate node n in certain layer j should be equal to the flow send out by the intermediate node in the same layer. Therefore the node potential must be 0;

Constraint (3.1g) is the network coding constraint, meaning that the flow for different sinks in the same layer can be combined together. Constraint (3.1h) confines that the actual flow in each edge cannot exceed the edge capacity.

Finally, notice that inequality (3.1b) follows from (3.1d) since the sink potentials are nonnegative, thus (3.1b) can be removed.

3.3 Polynomial Time Randomized Approximation

The preceding section proposes a RNC solution based on integer programming. However integer programming is a NP-hard problem itself, and further simplification is necessary. For linear rate-fidelity functions, we can develop a provably good polynomial-time randomized approximation algorithm. The idea is to relax the integer constraints and further reduce the RNC problem to one of the ordinary linear programming problems that is polynomially solvable. This approach is called the rounding method in combinatoric optimization literature [36].

The algorithm consists of the following three phases:

- Solve the non-integral version of (3.1)
- Construct candidate flows.
- Random flow selection.

Algorithm Details:

Step 1. Finding the fractional solution

By removing the integral constraint in (3.1), we get a relaxed version of the original integer programming problem which can be solved efficiently. Let $\hat{L}$ be the optimum solution of the relaxed problem and $\hat{y}^k_{i,j}$, $\hat{x}_k$ be the resulting variables. $\hat{L}$ is obviously an upper bound on the integral value of L.

Step 2. Flow stripping

Flow stripping is the process of constructing potential flows used in the random flow selection. For each layer k, we divide the resulting flow $\hat{x}_k$ into $\lceil \hat{x}_k \rceil$ subflows such that $x_k^i = 1$ for $1 \leq i \leq \lceil \hat{x}_k \rceil - 1$ and $x_k^{\lceil \hat{x}_k \rceil} = \hat{x}_k - \lfloor \hat{x}_k \rfloor$.

Step 3. Random graph selection

Pick a suitable scaling factor $\lambda \in [0, 1]$ (the computation of λ is described below). Let $\tilde{x}_k^i = 1 - \lambda x_k^i$. Then, for each subflow x_k^i, augment it to 1 and add it to the final resulting flow with probability λx_k^i while discard it with probability $\tilde{x}_k^i$. After the random flow selection, check the capacity constraint on all the edges. Accept the resulting flow design if no capacity constraint is violated, otherwise repeat the random selection process.

To prove the quality of the integer result obtained by above method, we need to show three things. First, the expected value of the resulting integral solution L^* is $\lambda \hat{L}$. Second, to bound the running time of our algorithm, we need to show that by choosing an appropriate value of λ, the probability that any of the capacity constraint is violated can be made arbitrarily small. Finally, we need to show that the value of L^* remains close to its expected value $\lambda \hat{L}$ with high probability.

We start by the following theorem.

Theorem 2. The expected value of L^* is $\lambda \hat{L}$.

Proof. When selecting subflows x_k^i for each layer k, the expected number of times a subflow is chosen is precisely $\lambda \hat{x}_k$. Applying this to the total flow on the graph and noting the linearity of L in the amount of flow proves the lemma. □

We then need to show that by choosing an appropriate value of λ, the probability that any of the capacity constraint is violated can be made arbitrarily small. To do this, we need to use Chernoff/Hoeffding bounds [37] as follows.

Lemma 2(Chernoff/Hoeffding Bounds). Let X be the sum of a number of independent random variables and let $\mu = E[X]$. Then we have

$$Prob\{x < (1-\beta)\mu\} \leq e^{-\frac{\beta^2\mu}{2}} \tag{3.2a}$$

$$Prob\{x > (1+\beta)\mu\} \leq e^{-\frac{\beta^2\mu}{3}} \tag{3.2b}$$

where $0 \leq \beta \leq 1$.

Let C_{min} be the smallest capacity of all the links in the network. Using the above lemma, we can prove the following theorem.

Theorem 3. Choose any $\epsilon \in [0,1]$ and assume that $C_{min} > 6ln\frac{|E|}{\epsilon}$. Then there exists a scaling factor $1/2 \leq \lambda \leq 1$ such that the probability that any of the capacity constraint is violated is less than ϵ.

Proof. Take a arbitrary link e. The expected flow in e with our randomized scheme is at most $\lambda C(e)$. We are interested in bounding the probability that the flow exceeds $C(e)$ and hence violates the capacity constraint on e. Using the notation of Lemma 2, we are interested in the probability of an event that $x > (1+\beta)\lambda C(e) = C(e)$, thus $\beta = 1/\lambda - 1$. If $1/2 \leq \lambda \leq 1$, then $0 \leq \beta \leq 1$, this probability is bounded as:

$$P_e = Prob\{x > (1+\beta)\mu\} \tag{3.3a}$$

$$\leq e^{-\frac{\beta^2\mu}{3}} \tag{3.3b}$$

$$= e^{-\frac{(1-1/\lambda)^2\lambda C(e)}{3}} \tag{3.3c}$$

To ensure that $P_e \leq \epsilon/|E|$, it is sufficient to have:

$$(\lambda^{\frac{1}{2}} - \frac{1}{\lambda^{\frac{1}{2}}})^2 \geq \frac{3ln\frac{|E|}{\epsilon}}{C(e)} \tag{3.4}$$

for some $1/2 \leq \lambda \leq 1$. Now consider the equation:

$$\eta(\lambda) = (\lambda^{\frac{1}{2}} - \frac{1}{\lambda^{\frac{1}{2}}})^2 - \frac{3ln\frac{|E|}{\epsilon}}{C(e)} = 0 \tag{3.5}$$

in the unknown λ. To see that this equation has a solution in the interval $[1/2, 1]$, it is sufficient to show that $\eta(1)\eta(1/2) < 0$. Note that $\eta(1) = -\frac{3ln\frac{|E|}{\epsilon}}{C(e)} < 0$ and $\eta(1/2) = \frac{1}{2} - \frac{3ln\frac{|E|}{\epsilon}}{C(e)} > 0$ by the assumption of the theorem. Which shows the existence of $\lambda \in [1/2, 1]$ for which $P_e \leq \epsilon/|E|$.

There are totally $|E|$ edges, so the probability that any constraint is violated is less than $|E|P_e \leq \epsilon$ by union bound. □

We have the following theorem on the quality of our integral solution L^* for any scaling factor λ provided that the constraints are satisfied.

Theorem 4. For all $0 < \epsilon < 1$, with probability at least $1 - \epsilon$, the value of L^* is lower bounded by $\lambda\hat{L} - \sqrt{2\lambda\hat{L}ln\frac{1}{\epsilon}}$.

Proof. Using the first part of Lemma 1 and letting $\beta = \sqrt{\frac{2ln(1/\epsilon)}{\lambda\hat{L}}}$ gives the result.

□

Chapter 4

Scalable Multicast using Multiple Description Codes

In this chapter, a new framework of jointly designing multiple description codes and linear network codes for scalable multicast is presented. The uneven erasure protection (UEP) [38] packetization scheme is applied as a practical way to generate balanced MDC packets. Unlike the network coding based scalable multicast schemes, the proposed technique does not explicitly divide the transmission into multicast layers and construct a separate network code for each layer. Instead, it only constructs one unified network code for the whole multicast transmission, and the computational expensive rate allocation problem is reduced to an UEP optimization problem, which can be solved efficiently.

The idea of transmitting UEP packets using linear network code to satisfy the individual max-flow values seems to be quite straightforward. However, as we will show in the following section, if it is not well designed, the decodability of the received UEP packets is hard to guarantee. The novelty and strength of this scalable multicast framework lies in a joint design of UEP and linear network codes. This joint design ensures that any linear combination of different UEP packets can be successfully decoded.

4.1 Transmitting UEP Packets using Linear Network Code

Apparently, the best performance of a scalable multicast system can achieve is when the max-flow rates of all receivers are satisfied, if possible. A topic closely related to this objective is the linear broadcast network code [13].

A w-dimensional linear network code is said to be a linear broadcast if, for each receiver, the number of received linearly independent packets equals the minimum value of w and the max-flow of that receiver ($dim(V_T) = min\{w, maxflow(T)\}$ for every sink node T) [13]. The existence of the linear broadcast network code is proved for an arbitrary network. By choosing a sufficiently large w ($w \geq maxflow(T)$ for all sink T), one can find a network code such that the dimension of linear independent packets received by each receiver achieves its max-flow value. However, the number of linearly independent packets available at a receiver does not always equal the number of packets which can be used to reconstruct the original data. There are such situations where the packets received by a receiver are linearly undecodable. This will never happen in the linear multicast because the dimension of received information always equals the code dimension w. But it happens in the linear broadcast when the k messages available at a receiver are linearly independent, but they are combinations of more than k source messages (as shown in Fig. 4.1). It is this undecodability issue that makes the problem of constructing optimal multirate network code to be NP-hard.

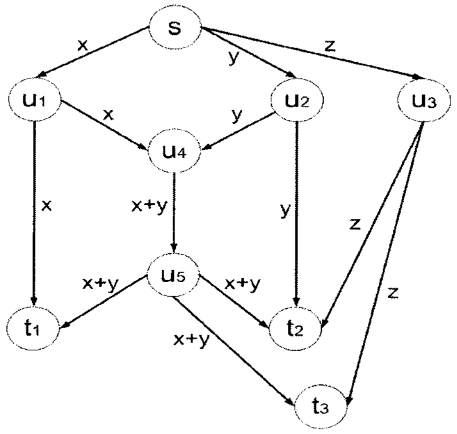


Figure 4.1: Receiver t_3 cannot decode all of its received packets.

To circumvent the aforementioned decodability issue, it is natural for one to resort to the universal decoding ability of multiple description code (MDC). MDC encodes source data into independently decodable descriptions, and the receiver is able to reconstruct the source given any subset of these descriptions. The more MDC descriptions received, the better reconstruction quality one can achieve. If the source data are encoded with MDC, those undecodable packets in the linear broadcast network code can be decoded partially if we take into account the redundancy among different MDC encoded descriptions. Then our problem can be formulated as designing the MDC scheme with minimum overall redundancy and the linear broadcast network code such that the max-flow rates of all receivers are satisfied.

Based on the reasoning above, a two-step joint design framework is proposed. First, the UEP packetization scheme is applied to the scalable data, as a practical way to generate balanced MDC packets. Next the UEP packets are sent over the network using a linear network code.

4.1.1 Uneven Erasure Protection (UEP) Packetization

Uneven erasure protection is a packetization scheme used to achieve robust transmission in lossy network. The idea is to partition the scalable coded source segments into data layers of decreasing importance, and protect these data layers by progressively weaker erasure correction codes. Precisely, in order to create w UEP packets, the scalable source segments are partitioned into w consecutive layers $L_1, L_2, \cdots, L_w$. Each layer L_k consists of k source segments $x_{k,1}, x_{k,2}, \cdots, x_{k,k}$. Each source layer is protected by a maximum distance separable (MDS) code, e.g. Reed-Solomon (RS) code. Precisely, the k source segments of the k-th source layer are protected by an (w, k) RS code. The effect of such a code is that, all k source segments can be recovered from any k channel symbols. Further, the UEP packets are formed accross the channel codewords. This packetization scheme guarantees that the first k source layers can be recovered from any k packets, $1 \leq k \leq w$. Fig. 4.2 shows an example of UEP packetization with three packets, and hence three layers. In this example, systematic RS codes are used.

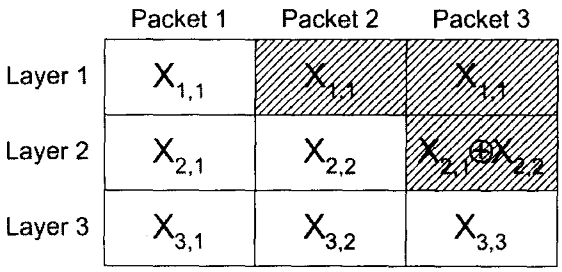


Figure 4.2: UEP packetization with three layers.

4.1.2 Linear Network Code

The linear network code applied to the UEP packets should ensure that any node is able to recover a number of source layers equal to its max-flow value. This is possible because the k-th layer of any UEP packet is a linear combination of only k source messages. Then, when only k packets are received at some sink, the first k layers can be recovered.

However, given an UEP design, there are certain constraints on the construction of the linear network code. For example, suppose the UEP design in Fig. 4.2 is used, and a receiver with max-flow rate 2 receives 2 packets. Assume that the global coding vectors of the received packets are $[1, 1, 0]$ and $[0, 0, 1]$ respectively, i.e. the first one is packet 1 $\oplus$ packet 2 in Fig. 4.2 while the second one is packet 3 in Fig. 4.2 alone. Although the rank of the received global coding vectors is 2, we cannot decode the first 2 layers because the layer 2 data in both packets are $X_{2,1} + X_{2,2}$. This example shows that the condition that the rank of received global coding vectors equals k does not ensure the decodability of the first k source layers.

Remark. Although careful construction of the network code is required, using UEP still greatly simplifies the problem: since any received packet is useful, we can simply construct a linear network code achieving all the individual max-flow rates according to a specific UEP design. As we will discuss later, with certain simplifications on our proposed solution, the complexity of constructing such a network code is equivalent to the complexity of ordinary linear network code construction.

4.2 Polynomial Time Algorithm

In this section, we describe the detailed polynomial time algorithm for joint UEP and network code design.

4.2.1 Basic Notation

Consider an acyclic unit capacity network $G = (V, E)$ where parallel edges are allowed. Any non-unit capacity link can be replaced by several parallel unit capacity links. Node $s \in V$ is the source node, and $T \subseteq V$ is the set of receivers. For every node $t \in V$, let $In(t)$ denote the set of incoming links to t; $Out(t)$ denote the set of outgoing links from t; $Start(e)$ denote the node at which edge e starts.

Let m_t denote the max-flow rate from s to receiver $t \in T$, and $w = \max\{m_t : t \in T\}$. We use linear codes over a finite field F. For each edge $e \in E$, let $f(e) \in F^w$ denote the w-dimensional global coding vector, and $k_e \in F^{|In(Start(e))|}$ denote the $|In(Start(e))|$-dimensional local coding vector. Then we have

$$f(e) = \sum_{e' \in In(Start(e))} k_e(e') f(e') \tag{4.1}$$

For an edge e in a path from s to $t \in T$, let $\phi_t(e)$ denote the predecessor edge on the path. Let $T(e)$ denote the set of sinks using e in the max-flow paths and let $P(e) = \{\phi_t(e) : t \in T(e)\}$ denote the set of predecessor edges of e in some flow paths. The notation $\langle \cdot \rangle$ stands for the linear span of a set of vectors.

4.2.2 Condition of Decodability

Let G_t be the $m_t \times w$ global network coding matrix of a receiver t, in which each row is the global coding vector of a received packet. As argued in the preceding section, the condition $rank(G_t) = m_t$ does not guarantee that receiver t recovers the first m_t source layers. Now we derive a sufficient condition under which the first m_t source layers can be decoded. Let M_i be the $w \times i$ generator matrix of the RS code applied in the i-th UEP layer, for any $1 \leq i \leq w$. For example, the generator matrix of the $(3, 2)$ RS code used in the second layer of Fig. 4.2 is $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$. Further, note that layer i of the packets received at node t is a linear combination of the i source segments $X_{i,1}, X_{i,2}, \cdots, X_{i,i}$. This linear combination, described by the matrix $G_t \cdot M_i$, must have dimension i in order to ensure that these i source segments can be recovered. Therefore, the coding algorithm should construct a network code such that for any receiver t,

$$rank(G_t \cdot M_i) = i \tag{4.2}$$

for all $i = 1, \cdots, m_t$.

A straightforward way to satisfy condition (4.2) is to check the linear independency in all the first m_t layers when we choose the coding vector for a certain edge. In fact, as long as condition (4.2) is satisfied for all $i = 1, \cdots, m_t$, the recovery of the first m_t source layers is guaranteed even if the channel code applied in each UEP layer i is an arbitrary (w, i) linear code, not necessarily an MDS code. The simplest example of such a code is a systematic code with all redundancy bits equal to 0. This corresponds to filling with zeros the shadowed area in Fig. 4.2. Moreover, using such codes in the UEP packetization scheme simplifies the algorithm for the network code design. This is because we only need to check the linear independency condition (4.2) in layer m_t, according to the following Lemma.

Lemma 1. If $M_i = \begin{bmatrix} I_i \\ 0_{(w-i)\times i} \end{bmatrix}$, where I_i is the i-dimensional identity matrix, for all $i = 1, \cdots, w$, then $rank(G_t \cdot M_k) = k$ implies that $rank(G_t \cdot M_i) = i$ for $i = 1, \cdots, k-1$.

Proof. Matrix G_t can be written as $[c_1, \cdots, c_w]$, where $c_1, \cdots, c_w$ are the w columns of G_t. Then $rank(G_t \cdot M_k) = rank([c_1, \cdots, c_k]) = k$ implies that the first k columns of G_t are linearly independent to each other. Clearly, since $G_t \cdot M_i = [c_1, \cdots, c_i]$, we have $rank(G_t \cdot M_i) = i$, for any $i = 1, \cdots, k-1$. □

4.2.3 Algorithm Description

Our multirate multicast network coding algorithm consists of the following three steps.

Step 1. Calculate the max-flow rate

For each receiver $t \in T$, calculate the max-flow rate m_t. Since the network is lossy, the expected capacities of edges are used to calculate the max-flow rate. This can be done using the shortest augmenting path algorithm in $O(|V|^2|E|)$ time, or using the FIFO preflow-push algorithm in $O(|V|^3)$ time [4].

Step 2. Design the optimal UEP

Let the total number of the packets in the UEP packetization scheme equal w, so there are also at most w UEP layers. The optimal UEP packetization scheme can be derived using the polynomial time algorithms described in [39]. The optimal solution can be found in $O(w^2L^2)$ time for the most general setting, and $O(wL^2)$ time for the data stream with convex R-D function. The UEP packet loss probability in the input of the algorithm is, in our case, the max-flow rate distribution among all the receivers. The algorithm minimizes the expected overall distortion within the given total rate budget $(L \times w)$ and outputs the size of each source layer. Note that the size of a layer can be zero. Then, fill the scalable source sequence into the optimized UEP packets, and fill all the redundant bits with 0.

Step 3. Construct the linear network code

Algorithm 1, which is inspired by the LIF algorithm [10], constructs a linear network code for a given UEP design. The key idea is to maintain an invariant that for each receiver $t \in T$ there is a set of m_t edges C_t, such that $\{f(c) \cdot M_{m_t} : c \in C_t\}$ form a basis of F^{m_t}.

Algorithm 1. (Construction of linear network code) the objective is to construct a w-dimensional F-valued linear network code satisfying condition (4.2) when $|F| > |T|$.

{

Insert a new source s' into V

Insert w parallel edges $\{e'_1, \cdots, e'_w\}$ from s' to s into E

for *each imaginary link* e'_i **do**

  $f(e'_i) = [0^{i-1}, 1, 0^{w-i}]$;

end

for *each* $t \in T$ **do**

  Construct m_t disjoint paths from s' to t, which start with imaginary links $\{e'_1, \cdots, e'_{m_t}\}$;

  Set $C_t = \{e'_1, \cdots, e'_{m_t}\}$;

end

for *each node* $t' \in V \backslash \{s'\}$ *in topological order* **do**

  for *each edge* $e \in Out(t')$ **do**

    Choose a global coding vector $f(e)$ such that $\forall t \in T(e)$, $f(e) \cdot M_{m_t}$ is linearly independent of $\{f(c) \cdot M_{m_t} : c \in C_t \setminus \{\phi_t(e)\}\}$; —(*)

    for *each* $t \in T(e)$ **do**

      $C_t = (C_t \backslash \{\phi_t(e)\}) \cup \{e\}$;

    end

  end

end

}

To see the existence of such a vector $f(e)$ in the step $(*)$ of Algorithm 1, note that for a receiver $t \in T(e)$, $g_{m_t}(e) = f(e) \cdot M_{m_t}$ equals the first m_t components of $f(e)$. Then, relation (5.6) implies: $g_{m_t}(e) = \sum_{e' \in In(Start(e))} k_e(e') g_{m_t}(e')$. Further, since $\{g_{m_t}(c) : c \in C_t\}$ form a basis of F^{m_t}, it follows that for any combination of $\{k_e(e') : e' \in P(e) \setminus \{\phi_t(e)\}\}$, there is one and only one $k_e(\phi_t(e))$ to make $g_{m_t}(e)$ linearly dependent of $\{g_{m_t}(c) : c \in C_t \setminus \{\phi_t(e)\}\}$. So there are $|F|^{|P(e)|-1}$ invalid local coding vectors for a receiver $t \in T(e)$, and the total number of invalid local coding vectors is $N \leq |T| \cdot |F|^{|P(e)|-1} < |F|^{|P(e)|}$. Therefore, there must exist at least one valid local coding vector.

Initializing the imaginary links takes $O(w^2)$ time. Finding a flow augmenting path takes $O(E)$ time. Hence constructing m_t disjoint path for each $t \in T$ takes $O(|E||T|w)$ time. The vector $f(e)$ can be found in $O(|T|^2 w)$ time, similarly to the deterministic implementation in LIF. Combining all the parts, the total running time of Algorithm 1 is $O(|E||T|^2 w)$.

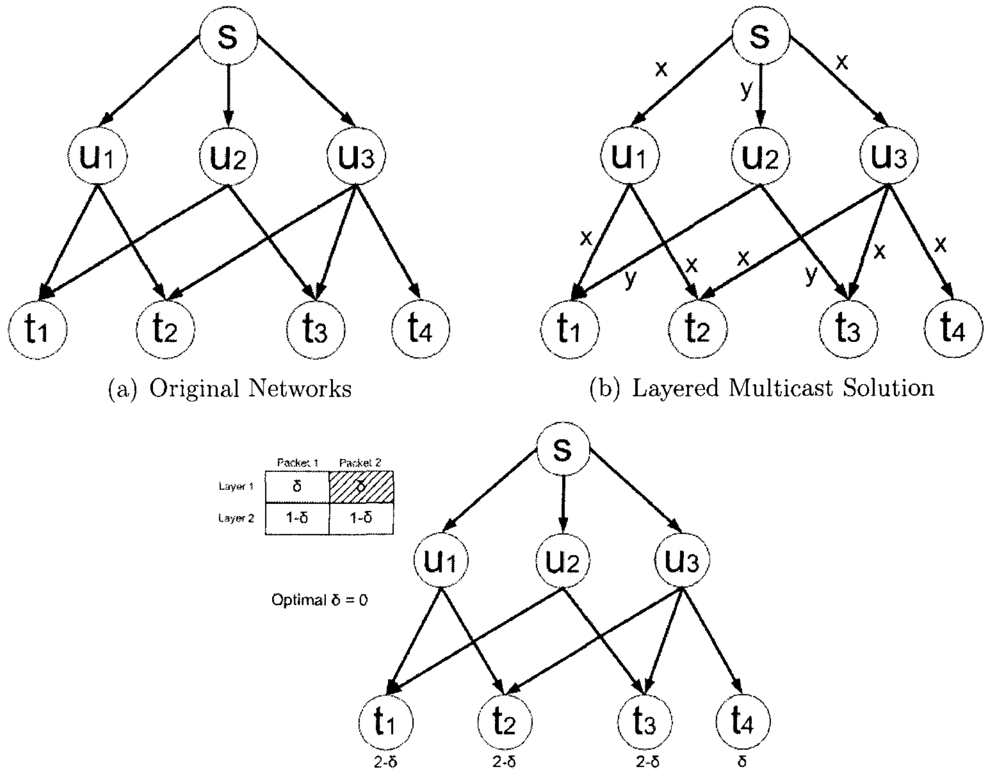
4.3 Relation to Scalable Multicast using Intra-layer Network Coding

Although the joint design framework described in this chapter takes a very different methodology as the intra-layer network coding method presented in previous chapter, they share the same essence: the use of linear combination of source data inside each data layer.

In the intra-layer network coding method, the optimal layer division is firstly calculated. Then a single uni-rate network code is constructed to linearly mix the data in each layer. While in the UEP plus network coding method, although there is only one network code constructed and no explicit division of data layers, the data layer division is actually included in the UEP design process. In fact, the UEP packetization itself is a linear combination of source data. The linearly combined data packets are further mixed by the single linear network code to form the final received packets.

Another connection between these two methods is the layered structure. In the intra-layer network coding method, the multicast transmission is explicitly divided into layers and each layer transmits a data layer using a separate network code. In the UEP plus network coding method, although only one single network code is used for the whole networks, the layered structure is actually implied inexplicitly by the UEP packetization.

With regard to the system performance, there are cases that these two methods achieve the same performance, and cases that they perform differently. For simplicity, suppose the objective function is linear, i.e. our goal is to maximize the overall throughput. Figure 4.3(a) shows an sample network that both method achieve the same throughput. Assume the capacity of all the links are 1, then the two approach can both achieve a throughput of 6 as shown in Figure 4.3(b)(c). Note that, in the UEP based method each UEP packet consists two data layers. However the first data layer reduces to 0 during the UEP optimization process due to the linearity of the objective function. If the objective function is non-linear, i.e. convex, the optimized data layers in the final UEP design are more likely to be non-zero. In Figure 4.4, a sample networks which leads the two methods to perform differently is illustrated. With the intra-layer network coding method, we can achieve a total throughput of 6,



(a) Original Networks

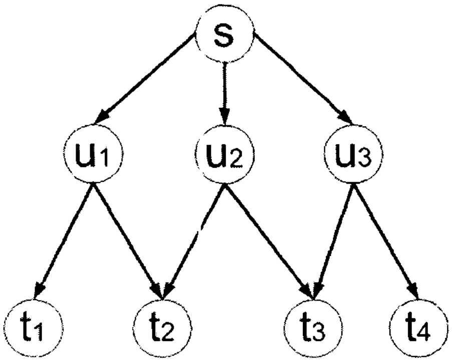
(b) Layered Multicast Solution

(c) Rainbow Multicast Solution

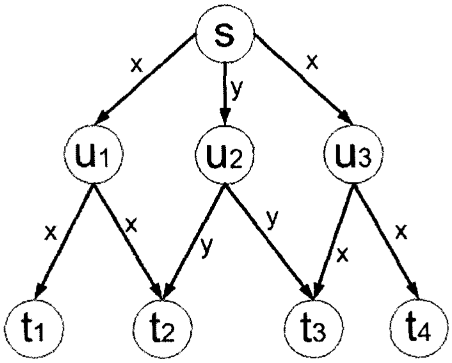
Figure 4.3: Case that two methods perform the same.

while the UEP based method can only achieve a total throughput of 4.

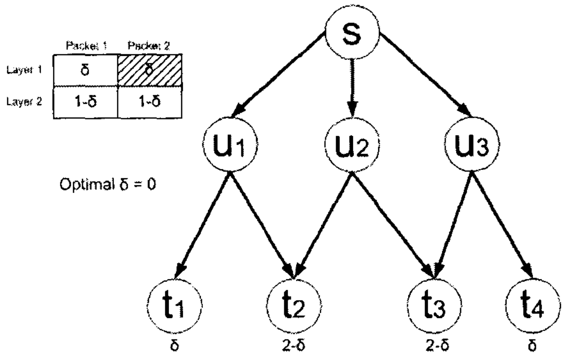
In fact, we can conclude that intra-layer network coding based method introduced in previous chapter always outperforms, or at least performs the same as the UEP based method. This is because any solution of the UEP based method is also a valid solution for the intra-layer network coding based method. Precisely, in an UEP based solution, sink nodes with the same maxflow value will receive the same amount of data (UEP layers), and the sink node with larger maxflow value will receive larger amount of data (UEP layers).



(a) Original Networks



(b) Layered Multicast Solution



(c) Rainbow Multicast Solution

Figure 4.4: Case that two methods perform differently.

Chapter 5

Scalable Multicast using Inter-layer Network Coding

The scalable multicast schemes introduced so far use intra-layer network coding. However the intra-layer constraint does not realize the full potential of network coding. To overcome the above limitation and take full advantage of network coding technique, we present a scalable multicast framework using inter-layer network coding in this chapter.

To perform inter-layer network coding, we still divide the transmission into multicast layers. However the concept of a multicast layer in the new framework is significantly different from previous work. In the previous methods, we do not differentiate multicast layer and data layer, since they are always consistent. Precisely, multicast layer k can only transmit the mixture of data in data layer k. While in the new inter-layer setting, we allow flow in multicast layer k to carry data in all data layers $1, 2, \cdots, k$. Network coding is applied inside each multicast layer, thus, messages sent in layer k are linear combinations of data in layers $1, 2, \cdots, k$. Therefore, network coding is actually applied across data layers, exhibiting the inter-layer characteristic of our network coding-based layered multicast technique. Another notable difference versus prior work is that the amount of flow delivered to different sinks in a single multicast layer is not necessarily the same in the new inter-layer framework. In other words, each multicast layer is not unirate. Moreover, we allow flow to transfer from one multicast layer to a higher multicast layer. Due to all the relaxations mentioned above, the proposed scheme has greater flexibility in optimizing the data flow compared to previous methods, and thus achieves higher throughput. At a first glance all these relaxations seemingly make it difficult to ensure the decodability of messages received at each sink. Indeed, the decodability is not guaranteed in each multicast layer separately, not even cumulatively over all lower multicast layers. But it is ensured over all multicast layers assigned to that node, and this is all that is needed.

5.1 Intra-layer Network Coding Solutions Revisited

As pointed out in the above discussion, previous layered multicast formulations using network coding perform intra-layer network coding only. Specifically, each layer of data is transmitted to the sinks in a multicast layer. Network coding is applied only inside each layer, not across layers. Data flow in different multicast layers cannot be encoded together, i.e. the messages transmitted in multicast layer k can only be the linear combination of source segments in data layer k. In order to understand the drawback of the intra-layer technique and to provide some insight on our proposed

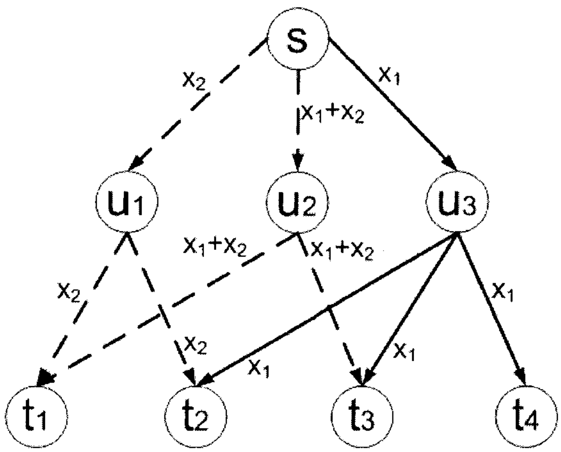


Figure 5.1: Optimal layered multicast solution for the given network.

inter-layer scheme we analyze the example shown in Figure 5.1. The network illustrated in the figure is a unit capacity network with 4 sinks t_1, t_2, t_3, t_4. The max-flows of the sinks are 2, 2, 2, 1, respectively. In order to achieve the max-flow rates at all sinks, the source data have to be divided into two layers x_1 and x_2. x_1 has to be delivered to all sinks in the first multicast layer and x_2 must be delivered to sinks t_1, t_2, t_3 in the second multicast layer. Since cooperation across layers is not allowed, no edge can transmit the combination of x_1 and x_2. Thus, the optimal layered multicast solution shown in Figure 5.1 (which in this case is the optimal multirate multicast solution as well) cannot be achieved.

The above example was first used in [40] when discussing the multilevel diversity coding problem. The author proved that the optimal rate cannot be achieved if network coding across layers was not allowed.

By carefully examining the optimal solution we observe that the flow can indeed be divided into two multicast layers, but using different criteria than in the intra-layer formulations. Consider the first multicast layer to consist of edges depicted in Figure 5.1 with solid lines and the second multicast layer to consist of dotted lines edges. It can be seen that the second multicast layer carries data in both data layers, not just in data layer 2. Precisely, edges (s, u_2), (u_2, t_1), (u_2, t_3) transmit a combination of x_1 and x_2. Moreover, note that sink t_1 does not receive any unit of flow in the first multicast layer, while in the intra-layer schemes it would receive one unit of flow. But then in order to compensate, t_1 receives two units of flow in the second multicast layer, while under the intra-layer framework it would receive only one. The above observations could be interpreted in the following way: for some sinks part of the data in the first layer is "delayed" and transmitted together with the data in the second layer.

This thought leads to our novel layered multicast technique based on inter-layer network coding. As usual the transmission is divided into multicast layers, but the concept of a multicast layer in our framework is significantly different from the intra-layer network coding formulations. Precisely, the flow in the k-th multicast layer is not confined to carry only combinations of data in data layer k. Instead, it is allowed to transport combinations of all data in the first k data layers. This is how network coding across data layers is performed. Another notable difference versus prior work is that the amount of flow delivered to different sinks during a single multicast layer is not necessarily the same. In other words, each multicast layer is not unirate. The number of data layers received by each sink is decided by the sink partition. Sinks in subset T_k will receive k layers of data. Moreover, we allow flow to transfer from one layer to a higher layer. The transfer of flow at the source node could be explained in the following way. In the first multicast layer the source node s has available R_1 units of flow (i.e., data layer 1) for transmission to any sink. If only $R_1 - 1$ units of flow are transmitted to some sink $t \in T_k$, then the remaining unit of flow is available for transmission to sink t in the second multicast layer together with the new $R_2 - R_1$ flow units corresponding to data layer 2. This unit of flow could be used in multicast layer 2 (i.e., transmitted to t), in which case we say that it is "delayed" and transferred to layer 2, or it can be further "delayed" and transmitted to t in a higher layer. However, by the "end" of multicast layer k all "delayed" flow must reach sink t in order to ensure that t receives all data in the first k data layers. Flow transfer from a lower multicast layer to a higher layer is admitted at intermediate nodes as well.

Similar to the previously mentioned schemes, the inter-layer network coding solution also consists of two major steps. First, a flow optimization problem is solved to maximize the overall received flow among all the receivers. Next, a single network code is constructed to achieve the optimal rates for all sinks.

5.2 Flow Optimization for Inter-layer Network Coding

5.2.1 Inter-layer Flow Optimization Formulation

For every node $v \in V$, let $In(v)$ denote the set of incoming links to v and let $Out(v)$ denote the set of outgoing links from v. $C_{i,j}$ is the capacity of edge (i, j), measured in number of source segments during a transmission slot of fixed duration δ, and s is the source node. Recall that the size of a source segment is fixed. Moreover, a source segment represents an indivisible unit of flow.

We divide the flow into N layers. Any sink $t \in T_k$, for some k, may receive flow in the first k layers with the requirement that the total amount of flow received over the first k layers equals R_k. Let $x_{i,j}^{t,l}$ be the flow on edge (i, j) for sink t in layer l. Define $b_j^{t,l}$ to be the potential of node j for sink t in layer l, which is defined as the difference between the incoming flow and the outgoing flow. Negative node potential indicates a supplying node, while positive node potential indicates a demanding node. Let $y_{i,j}^k$ be the actual flow on edge (i, j) in layer k (over all sinks). For each sink node t, let $L(t)$ denote the number of data layers that sink t will receive (i.e., $L(t) = k$ if and only if $t \in T_k$). Then, the flow optimization problem can be formulated as shown in Figure 5.2.

Constraint (5.1c) follows from the definition of node potential. (5.1d), (5.1e) are the potential constraints at the source node. The total flow sent out from source s to sink t over the first $L(t)$ layers should equal $R_{L(t)}$. But the flow sent to t over the first j layers, $j < L(t)$, can be less than R_j because some part of the flow in lower layers can be "delayed" and transmitted in higher layers.

Constraints (5.1f), (5.1g) concern the potential of sink nodes. Since flow in lower layers can transfer to higher layers, the total flow received by sink t over the first j layers, $j < L(t)$, can be less than the total flow sent out to t by source s over those layers. But the total flow received over the first $L(t)$ layers should equal the total flow sent by the source over those layers.

(5.1h), (5.1i) are the constraints for the potentials at the intermediate nodes. Note that the potential of some intermediate node l can be negative for some layers. For example, negative potential in layer j means that some flow in layer $1, \cdots, j-1$, transfers to layer j at the current node. Since the transfer of flow is allowed only from a lower to a higher layer, the total incoming flow (designated to sink t) at intermediate node l, over the first j layers, cannot be smaller than the total outgoing flow. Moreover if $j = L(t)$ the two quantities have to be equal.

Since the amount of flow which transfers from layer $1, \cdots, j-1$ to layer j should be less or equal to the sum of flow in layer $1, \cdots, j-1$, the sum of potentials in first j layers must be non-negative.

Constraint (5.1j) is the network coding constraint, meaning that the flow for different sinks in the same layer can be combined together. Constraint (5.1k) confines that the actual flow in each edge cannot exceed the edge capacity.

Finally, notice that inequality (5.1b) follows from (5.1e) since the sink potentials are nonnegative, thus (5.1b) can be removed.

5.2.2 Example

The following example illustrates a solution to the above flow optimization problem and the proposed inter-layer network code. To better illustrate the idea, we use a linear cost function $\phi(R) = R$. Consider the unit capacity network shown in Figure 5.3. s is the source node and t_1, t_2, t_3 are the sinks. The max-flow to t_1, t_2, t_3 are 2, 2, 1 respectively, and therefore sinks are divided into 2 two subsets: $T_1 = \{t_3\}$, $T_2 = \{t_1, t_2\}$. The flow paths found by the flow optimization algorithm (indicated by the values $y_{i,j}^k$) are shown in Figure 5.4. Edges in the first layer are shown as solid lines, while edges in the second layer are shown as dashed lines.

Note that under the proposed intra-layer NC framework in order to achieve the optimal solution, we would need two multicast layers as shown in Figure 5.5. Edge (u_4, u_5) should be included in both multicast layers in order to carry x_1 to t_2 and x_2 to t_1. This is not possible since the edge is a unit capacity edge. On the other side, in the inter-layer network coding framework edge (u_4, u_5) is included in the second multicast layer and carries $x_1 + x_2$. Then the information about x_2 reaches sink t_1 via the path $s - u_2 - u_4 - u_5 - t_1$, which is completely included in layer 2. The information about data x_1 reaches sink t_2 via the path $s - u_1 - u_4 - u_5 - t_2$, which has the first two edges in the first layer and the next two edges in the second layer. This path illustrates the concept of transfer of flow between layers. At node u_4 one unit of flow directed to sink t_2 transfers from layer 1 to layer 2.

5.2.3 Observations

An important observation is that the solution of the inter-layer network coding flow optimization problem, when the sinks partition is $\mathcal{T}_{\text{max-flow}}$, is guaranteed to be at least as good as that of intra-layer network coding flow optimization schemes shown in Chapter 3. This is due to the fact that the intra-layered optimization formulation is included in the proposed inter-layered formulation as a special case. Precisely, if we change the inequalities in constraints (5.1d), (5.1f), (5.1h) into equalities, we will obtain the exact formulation of layered multicast in Chapter 3.

Another notable observation is that the solution of problem in Figure 5.2 improves as the partition $\mathcal{T}$ becomes finer. In order to justify this claim consider two partitions $\mathcal{T}_1 = \{T_1, \cdots, T_N\}$ and $\mathcal{T}_1' = \{T_1', \cdots, T_{N'}'\}$, such that $\mathcal{T}_1$ is finer than $\mathcal{T}_1'$. Then there are integers $1 = m_1 < m_2 < \cdots < m_{N'} < m_{N'+1} = N + 1$ such that $T_k' = \cup_{j=m_k}^{m_{k+1}-1} T_j$, $1 \leq k \leq N'$. Then any feasible solution of flow optimization problem corresponding to $\mathcal{T}_1'$ can be converted to a feasible solution corresponding to $\mathcal{T}_1$, by letting the flow in multicast layer m_k (on each edge and for each sink) for the latter case to be equal with the flow in multicast layer k for the former case, $1 \leq k \leq N'$, and by assigning zero flow in any other multicast layer for $\mathcal{T}_1$. Therefore, in order to improve the performance of the proposed layered multicast scheme we will consider an $|T|$-size sinks partition in Problem 2, in other word a partition where each subset consists of only one sink. Since this partition is finer than the partition proposed in [41], it is expected that we can achieve better performance than in [41].

5.2.4 Optimization with Convex Cost Function

A simple choice for the fidelity function is $\phi(R) = R$. Then the problem of Figure 5.2 is an integer linear program, for which heuristic algorithms are widely available. However, there are other more meaningful fidelity measures for multimedia applications (e. g. PSNR, SNR, negative mean squared error), and the solution which maximizes the overall received flow is not necessarily the solution with the highest overall reconstruction fidelity. Thus, in order to improve the performance of the layered multicast scheme, the real fidelity function (which is not linear) is more suitable. To handle such a case, we will convert the flow optimization problem into a linear integer program.

Let the fidelity function be a non-decreasing function $\phi(R)$, defined for any integer $R, 1 \leq R \leq M$. Recall that M is the number of source segments available at source node s, for transmission during a time unit. Because function $\phi(\cdot)$ is non-decreasing, it follows that there are non-negative real numbers c_j, $1 \leq j \leq M$, such that

$$\phi(R) = \phi(0) + \sum_{j=1}^{R} c_j \times 1 + \sum_{j=R+1}^{M} c_j \times 0, \tag{5.2}$$

for any integer R with $1 \leq R \leq M$. To linearize the flow optimization problem we introduce additional binary variables $r_{k,j} \in \{0, 1\}$, $1 \leq k \leq N$, $1 \leq j \leq M$. The value of $r_{k,j}$ indicates whether or not the data segment x_j is included in the first k data layers. Precisely, $r_{k,j} = 1$ if $j \le R_k$ and $r_{k,j} = 0$ otherwise. Then

$$\phi(R_k) = \phi(0) + \sum_{j=1}^{M} c_j r_{k,j} \tag{5.3}$$

and the optimization problem can be recast as in Figure 5.6. Notice that condition (5.4c) enforces the fact that, if $r_{k,j} = 1$ then $r_{k,j'} = 1$ for all $1 \le j' \le j$, in other words, if data segment x_j is included in the first k data layers then all previous segments are also part of the first k data layers.

5.3 Network Code Construction

In this section, we present a polynomial time algorithm which constructs a linear network code for the given network, such that the optimized flow rates for all the sinks are achieved.

5.3.1 Algorithm Description

Given the optimized rates $R_1, \cdots, R_N$, and the flow in each multicast layer, we want to construct a layered multicast code which achieves the rate R_k at each sink in $T_k, 1 \le k \le N$.

We choose the maximum rate R_N as the message dimension, i.e. the source transmits R_N source segments (or messages) in a unit time. Our framework guarantees that any sink in subset T_k receives R_k messages, which are the linear combinations of the first R_k source segments. Moreover, these messages are linearly independent, thus ensuring the decodability of all first R_k source segments.

Note that any network G can be converted to an equivalent unit-capacity network G' and the solution to the data flow optimization problem for network G can be transformed to an equivalent solution for network G'. For this purpose, any edge (i, j) with capacity n of G is replaced by n parallel unit-capacity links of G'. These n links are further partitioned into N sets such that the l-th set contains $y^l_{i,j}$ edges, for $1 \leq l \leq N$. Note that the total number of links in these N sets equals $\sum_{l=1}^{N} y^l_{i,j}$, which can be less than the total number of parallel links between i and j. The edges in the l-th set will carry only flow in layer l. Notice that such a partition is possible due to relation (5.1k). Moreover, some of these sets may be empty. In particular, the l-th set is empty if $y^l_{i,j} = 0$.

Upon this conversion we construct for each sink $t \in T$ a set of $R_{L(t)}$ edge-disjoint paths $Q^1_t, \cdots, Q^{R_{L(t)}}_t$ from s to t in the graph G'. If an edge e carries flow in layer l, we say that edge e is in layer l and use the notation $L(e) = l$. Notice that such an edge transports only one unit of flow. Then the edge-disjoint paths $Q^1_t, \cdots, Q^{R_{L(t)}}_t$ are constructed such that the following conditions are satisfied.

C1) All the paths contain only edges in layers 1 through $L(t)$.

C2) For any path Q and any two consecutive edges e_1 and e_2 of Q, edge e_1 is in a lower or the same layer as e_2, i.e., $L(e_1) \leq L(e_2)$.

C3) For any $i, 1 \leq i \leq R_{L(t)} - 1$, the first edge of Q^i_t is in a lower or the same layer as the first edge in Q^{i+1}_t.

The existence of such paths satisfying the above requirements is ensured by the constraints imposed on the node potentials (5.1d-5.1i). Further, for an edge e in a path from s to $t \in T$, let $\phi_t(e)$ denote the predecessor edge on the path. Let $T(e)$ denote

```
for each e ∈ E do
|  Set f(e) = [0^{R_N}];
end
for each t ∈ T do
|  Construct R_{L(t)} edge-disjoint paths {Q_t^1, ··· , Q_t^{R_{L(t)}}} from s to t such that conditions
|  C1-C3 are satisfied;
end
Insert a super source s' into V
for each t ∈ T do
|  Add R_{L(t)} parallel imaginary edges {e_t^1, ··· , e_t^{R_{L(t)}}} from s' to s into E;
|  Set f(e_t^i) = [0^{i-1}, 1, 0^{R_N - i}];
|  Assign e_t^i to a path Q_t^i;
|  Set C_t = {e_t^1, ··· , e_t^{R_{L(t)}}};            —(*)
end
for each node t' ∈ V\{s'} in topological order do
|  for each edge e ∈ Out(t') do
|  |  Choose a global coding vector f(e) such that f_j(e) = 0 for all j, R_{L(e)} + 1 ≤ j ≤ R_N,
|  |  and
|  |  ∀t ∈ T(e), f(e) is linearly independent of {f(c) : c ∈ C_t \ {φ_t(e)}};        —(**)
|  |  for each t ∈ T(e) do
|  |  |  C_t = (C_t\{φ_t(e)}) ∪ {e};
|  |  end
|  end
end
```

Algorithm 1: Construction of inter-layer linear network code. The objective is to construct an R_N-dimensional F-valued linear network code achieving the rate $R_{L(t)}$ for each sink node $t \in T$, when $|F| > |T|$.

the set of sinks using e in the flow paths.

Algorithm 1, which is inspired by the LIF algorithm [10], constructs a linear network code such that the optimized flow rates are achieved.

The algorithm constructs an R_N-dimensional global encoding vector over a finite field F with $|F| > |T|$, $f(e) = (f_1(e), f_2(e), \cdots f_{R_N}(e))$, for each edge e which carries flow to some sink.

The key idea in order to ensure the algorithm correctness is to maintain an invariant that for each sink t there is a set C_t of $R_{L(t)}$ edges such that the global encoding vectors in the set $\{f(c) : c \in C_t\}$ are linearly independent and, moreover, $f_j(c) = 0$ for all $j, R_{L(c)} + 1 \leq j \leq R_N$. The meaning of the latter condition is that since edge c is in layer $L(c)$, the message passed along this edge can only be a function of source segments $x_1, \cdots, x_{R_{L(c)}}$. Furthermore, the set C_t must contain an edge from each path Q_t^i, $1 \leq i \leq R_{L(t)}$, and at the end of the algorithm we must have $C_t \subseteq In(t)$.

5.3.2 Proof of Correctness

The correctness of Algorithm 1 follows from the following lemmas.

Lemma 1. Assign each imaginary edge e_t^i to a layer as follows. Let $L(e_t^i) = k$ if and only if $R_{k-1} < i \leq R_k$. Then, after assigning the imaginary edges to the $s - t$ paths, condition C2 is still satisfied for all flow paths. Moreover, the invariant holds at the initialization step.

Proof. The fact that the invariant holds at the initialization step is obvious. It remains to prove the first claim. For any sink t, and any $k, 1 \leq k \leq R_{L(t)}$, let $n(t, k)$ denote the number of $s - t$ paths for which the first edge (before the inclusion of imaginary edges) is in layer k. According to the source potential constraints (5.1d) and (5.1e), we have

$$\sum_{i=1}^{j} n(t,i) \leq R_j, \quad 1 \leq j < L(t) \tag{5.5a}$$

$$\sum_{i=1}^{j} n(t,i) = R_j, \quad j = L(t) \tag{5.5b}$$

The above conditions together with C3 imply that the first edge in the path Q_t^i (before the inclusion of the imaginary edge e_t^i in the path) is in at least k-th layer, where $R_{k-1} < n \leq R_k$. Since $L(e_t^i) = k$ the conclusion of the lemma follows.□

Lemma 2. The global coding kernel $f(e)$ in step (**) can be found, when $|F| > |T|$.

Proof. This proof closely follows the proof of Lemma 4 in [10]. Let $P(e) = \{\phi_t(e) : t \in T(e)\}$ denote the set of predecessor edges of e in some flow paths. The global encoding vector $f(e)$ is constructed by finding first a local encoding vector $(k_e(e') : e' \in P(e))$ and setting

$$f(e) = \sum_{e' \in P(e)} k_e(e') f(e'). \tag{5.6}$$

Since all flow paths satisfy conditions C2 it follows that $L(e') \leq L(e)$ for all $e' \in P(e)$. By the invariant, we have $f_j(e') = 0$ for all $j, R_{L(e')} + 1 \leq j \leq R_N$. Hence, (5.6) will further ensure that $f_j(e) = 0$ for all $j, R_{L(e)} + 1 \leq j \leq R_N$.

It remains to show that there exists a local encoding vector $(k_e(e') : e' \in P(e))$ such that $f(e)$ is linearly independent of $\{f(c) : c \in C_t \setminus \{\phi_t(e)\}\}$ for any $t \in T(e)$. By condition C1, we have $L(c) \leq L(t)$ for all $c \in C_t$, hence the last $R_N - R_{L(t)}$

components of $f(c)$ are zeros. Then, for each $t \in T(e)$ and $c \in C_t$, let $f'(c)$ denote the $R_{L(t)}$-dimensional vector obtained from $f(c)$ after removing the last $R_N - R_{L(t)}$ components. Define $f'(e)$ in the same manner. Clearly, relation (5.6) still holds if f is replaced by f', i.e.

$$f'(e) = \sum_{e' \in P(e)} k_e(e') f'(e'). \tag{5.7}$$

Note that due to the invariant, the set of vectors $\{f'(c) : c \in C_t\}$ forms a basis of $F^{R_{L(t)}}$. Then when writing $f'(e)$ as a linear combination of the vectors in this basis, the coefficient assigned to basis vector $f'(\phi_t(e))$ must be $k_e(\phi_t(e)) + \alpha$ for some uniquely determined α which does not depend on $k_e(\phi_t(e))$.

It follows that for any choice of $\{k_e(e') : e' \in P(e) \setminus \{\phi_t(e)\}\}$, there is one and only one $k_e(\phi_t(e))$ to make $f(e)$ linearly dependent of $\{f(c) : c \in C_t \setminus \{\phi_t(e)\}\}$, namely, $k_e(\phi_t(e)) = -\alpha$. So there are $|F|^{|P(e)|-1}$ invalid local coding vectors for a receiver $t \in T(e)$, and the total number of invalid local coding vectors is $N \leq |T| \cdot |F|^{|P(e)|-1} < |F|^{|P(e)|}$. Therefore, there must exist at least one valid local coding vector.□

Remark 2. The new algorithm does not require a larger field size compared to previous layered multicast scheme.

Lemma 3. Any sink $t \in T$, is guaranteed to receive $R_{L(t)}$ messages which are linear combinations of the source messages $x_1, \cdots, x_{R_{L(t)}}$. Moreover, these $R_{L(t)}$ messages are linearly independent, thus ensuring the recovering of the first $R_{L(t)}$ data messages.

Proof. By Lemmas 1 and 2 the invariant holds at the end of the algorithm. Hence

t receives $R_{L(t)}$ messages, carried along the edges in C_t. Furthermore, due to condition C1, all edges in C_t are in layer $L(t)$ or lower layers. Therefore, these messages are necessarily linear combinations of the data in the first $L(t)$ layers, $x_1, \cdots, x_{R_{L(t)}}$. They are also linear independent by the invariant. Thus the proof is complete. □

Remark 3. For some sink t it is not guaranteed that the messages received over the edges in some multicast layer k are decodable. Moreover the decodability is not guaranteed either for all messages received over the first k multicast layers, but the decodability of messages received over the first $L(t)$ multicast layers is ensured, and this is all that matters. This concept is illustrated in Figure 5.7. The example in Figure 5.7 shows the source segment allocation for some sink t in T_3. The vertical axis denotes the flow received by the sink, while the horizontal axis denotes the source segments. Each row of the matrix can be considered as the global coding vector of a messages received at the sink. The shadowed blocks indicate non-zero coefficients and blank blocks indicate zero coefficients. Note that, given the flow over the first 2 layers, sink t cannot decode the first R_2 source segments. However, it can decode the first R_3 source segments received over the first three layers because the sink is guaranteed to receive R_3 units of flow.

5.3.3 Complexity Analysis

Initializing the imaginary links takes $O(R_N^2)$ time. Finding a flow augmenting path takes $O(E)$ time. Hence constructing $R_{L(t)}$ disjoint path for each $t \in T$ takes $O(|E||T|R_N)$ time. The global coding vector $f(e)$ can be found in $O(|T|^2 R_N)$ time, similarly to the deterministic implementation in LIF. Combining all the parts, the total running time of Algorithm 1 is $O(|E||T|^2 R_N)$, which is the same as in the previous intra-layer network coding schemes.

$$
\begin{aligned}
\max \quad & \sum_{t \in T} \phi(R_{L(t)}) && \text{(5.1a)}\\
\text{subject to} \quad & R_1 \leq R_2 \leq \cdots \leq R_N && \text{(5.1b)}\\
& b_j^{t,l} = \sum_{(i,j) \in In(j)} x_{i,j}^{t,l} - \sum_{(j,h) \in Out(j)} x_{j,h}^{t,l} && \text{(5.1c)}\\
& R_j \geq -\sum_{i=1}^{j} b_s^{t,i}, \quad \forall t \in T, 1 \leq j < L(t) && \text{(5.1d)}\\
& R_j = -\sum_{i=1}^{j} b_s^{t,i}, \quad \forall t \in T, j = L(t) && \text{(5.1e)}\\
& \sum_{i=1}^{j} b_t^{t,i} \leq -\sum_{i=1}^{j} b_s^{t,i}, \forall t \in T, 1 \leq j < L(t) && \text{(5.1f)}\\
& \sum_{i=1}^{j} b_t^{t,i} = -\sum_{i=1}^{j} b_s^{t,i}, \quad \forall t \in T, j = L(t) && \text{(5.1g)}\\
& \sum_{i=1}^{j} b_n^{t,i} \geq 0, \forall t \in T, 1 \leq j < L(t), n \notin \{s,t\} && \text{(5.1h)}\\
& \sum_{i=1}^{j} b_n^{t,i} = 0, \quad \forall t \in T, j = L(t), n \notin \{s,t\} && \text{(5.1i)}\\
& y_{i,j}^{l} = \max_{t \in T} \{x_{i,j}^{t,l}\}, \quad \forall l \quad 1 \leq l \leq N && \text{(5.1j)}\\
& \sum_{l=1}^{N} y_{i,j}^{l} \leq C_{i,j}, \quad \forall (i,j) \in E && \text{(5.1k)}\\
& x_{i,j}^{t,l} \text{ is non-negative integer}, \forall t \in T, \forall (i,j) \in E && \text{(5.1l)}
\end{aligned}
$$

Figure 5.2: Inter-layer flow optimization.

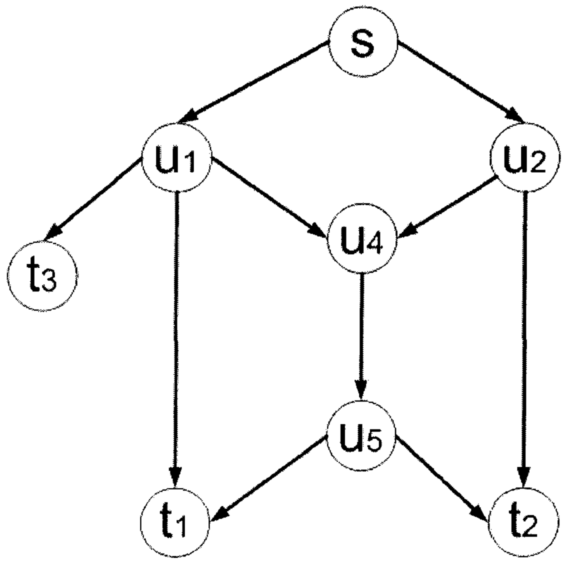


Figure 5.3: Example network with 3 sinks.

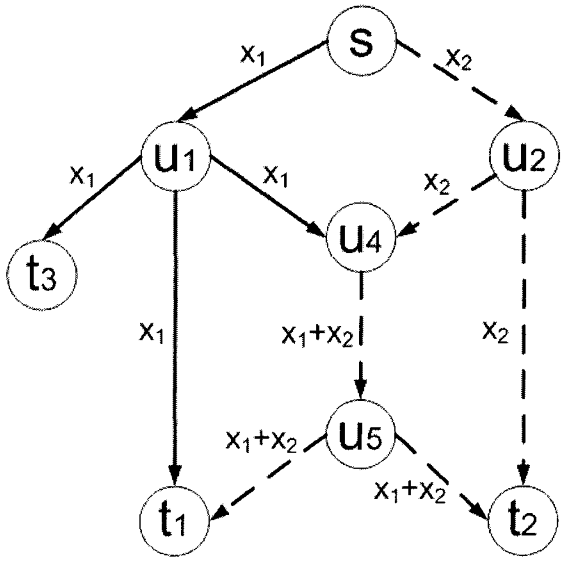
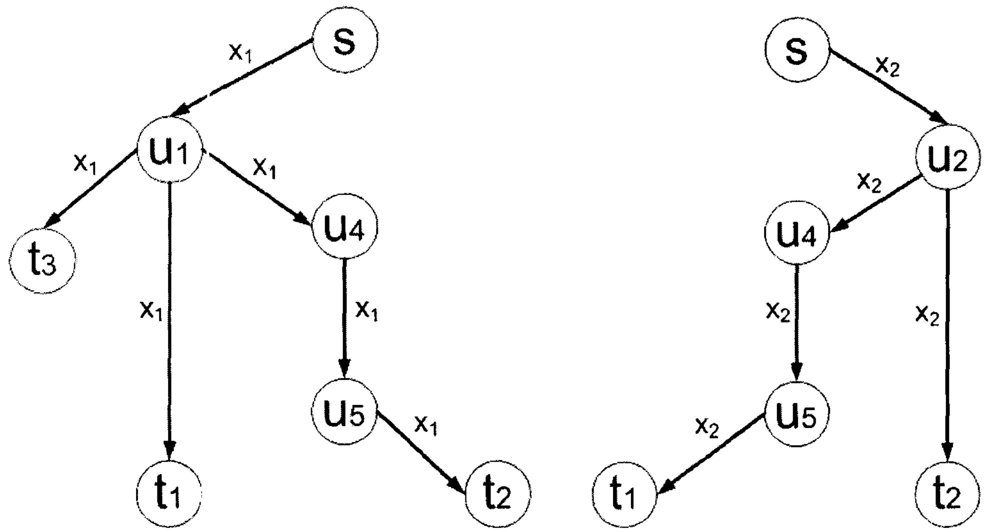


Figure 5.4: Solution produced by the inter-layer network coding scheme. This is the optimal layered multicast solution for the given network.



(a) Layer 1 (b) Layer 2

Figure 5.5: Optimal layered multicast solution for network in Figure 5.3 cannot be achieved with the intra-layer network coding technique unless edge (u_4, u_5) has capacity 2.

$$\max \quad \sum_{k=1}^{N}\sum_{j=1}^{M} c_j r_{k,j} \tag{5.4a}$$

$$\text{such that} \quad \sum_{j=1}^{M} r_{k,j} = R_k, \quad 1 \le k \le N \tag{5.4b}$$

$$r_{k,1} \ge r_{k,2} \ge \cdots \ge r_{k,M}, \quad 1 \le k \le N \tag{5.4c}$$

$$r_{k,j} \in \{0,1\}, \quad 1 \le k \le N, \quad 1 \le j \le M \tag{5.4d}$$

$$\text{conditions (5.1b)-(5.1l) hold} \tag{5.4e}$$

Figure 5.6: Linearization of the flow optimization problem.

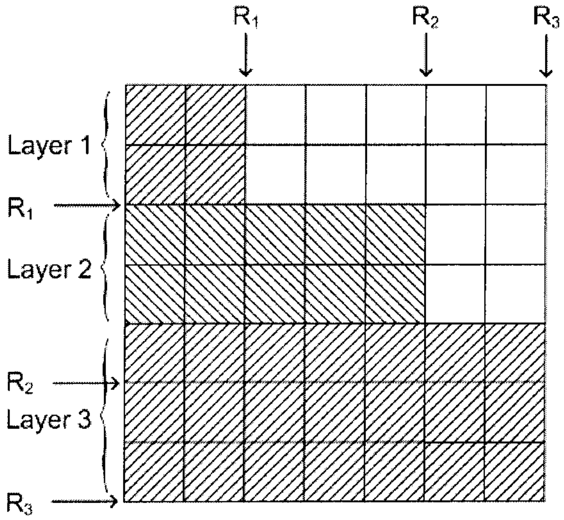


Figure 5.7: Example of data allocation among layers for a sink in layer 3.

Chapter 6

Simulation Results

This section contains some simulation results of the proposed network coding based scalable multicast schemes, as well as the previous non network coding based solutions.

6.1 Simulation Setup

In the simulations, we consider a family of networks which were first introduced in [10]. In this network model, all the sinks are connected to a central source node through a group of intermediate nodes (as the network shown in Figure 6.1). This network model mimics the practical multimedia distribution system with several distributed servers. All of the distributed servers in U connect to a central server s, and each client in T connects to several distributed servers.

The networks used in simulations are randomly generated as follows. We start with the source node s and add intermediate nodes and sink nodes sequentially. We set the number of intermediate nodes to be N_U, and each intermediate node connects

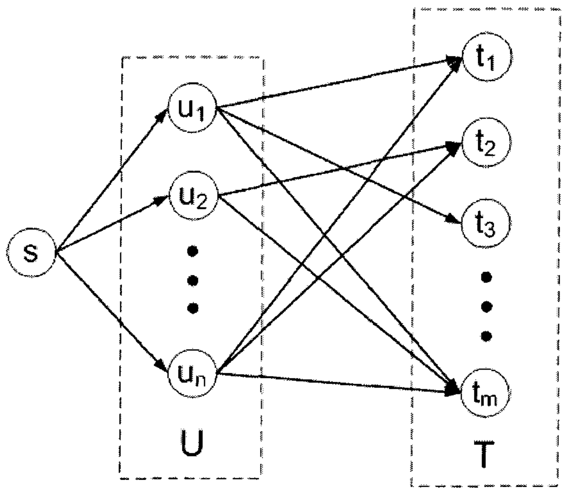
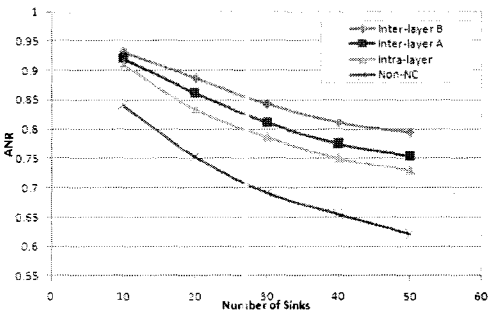


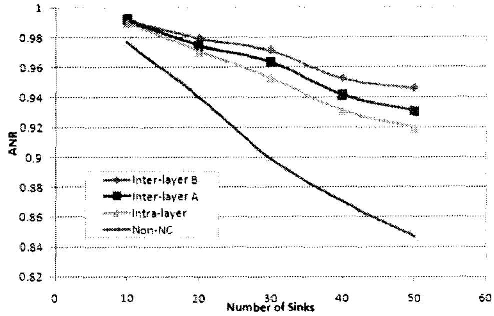
Figure 6.1: Network model used in simulations.

directly to the source s. The total number of sinks is N_T, and each sink randomly connects to P percent of the intermediate nodes. Once the network is constructed, we assign a random capacity between 0 to C_1 (kbits/s) to the edges between s and U, and assign a random integer capacity between 0 to C_2 (kbits/s) to the edges between U and T.

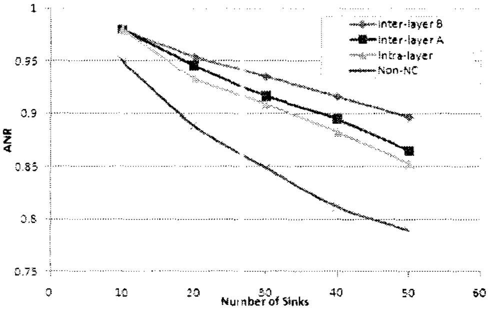
We will compare the performance of the scalable multicast with intra-layer network coding as proposed in Chapter 3, the scalable multicast scheme with inter-layer network coding discussed in Chapter 5 and a layered multicast scheme without network coding. We will also test the impact of refining the sink partition $\mathcal{T}$, as discussed in Chapter 5. We consider two cases for the inter-layer network coding scheme: 1) $\mathcal{T} = \mathcal{T}_{\text{max-flow}}$, i.e. the sink partition correspond to the max flow value of sink nodes; 2) $\mathcal{T}$ is a refinement of $\mathcal{T}_{\text{max-flow}}$ where each subset contains only one sink. We refer to the above two cases as scheme A and B, respectively. According to the observations in Subsection 5.2.3, we expect for scheme B to achieve a higher performance than scheme A, and both to outperform the other two methods.



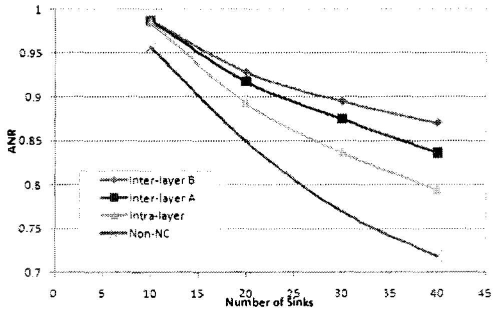
(a) N_U=10, C_1=320 kbit/s, C_2=320 kbit/s, P=50%



(b) N_U=10, C_1=640 kbit/s, C_2=320 kbit/s, P=50%



(c) N_U=10, C_1=640 kbit/s, C_2=320 kbit/s, P=25%



(d) N_U=20, C_1=320 kbit/s, C_2=320 kbit/s, P=25%

Figure 6.2: Average normalized rates of four different layered multicast schemes.

6.2 Performance Comparison of Rate-maximized Layered Multicast Schemes

We first compare the performance of all candidate schemes using the received flow as fidelity function in the flow optimization problem (i.e. the fidelity function is linear to the received flow). In other words we compare the solutions which maximize the overall received flow. The comparison is with respect to a performance measure called Average Normalized Rate (ANR), which is defined as the ratio between the total rate received by all sinks and the sum of the max-flow values of all sinks. Clearly, the larger the ANR, the better the scheme. Although the optimal ANR value for a certain network is generally unknown, an obvious upper bound of ANR is 1. Since there does not necessarily exist a network code that achieves the individual max-flow of all the sinks, the upper bound 1 is not tight for all the networks, even for the optimal solution of multirate multicast.

The performance of the four schemes is evaluated for different network sizes, and the results are plotted in Figure 6.2. In Figure 6.2(a), $C1$ and $C2$ are both set to be 320kbit/s, while each sink node connects to 50% of the intermediate nodes. In Figure 6.2(b), $C1$ is changed to 640 kbit/s, twice the value of $C2$, while the connectivity remains 50%. In Figure 6.2(c), $C1$ and $C2$ remain 640 kbit/s and 320 kbit/s respectively, but each sink connects to 25% of the intermediate nodes. The duration of a transmission slot is $\delta = 1$ second, the size S of a source segment is 30 kbits, and $N_U = 10$ in cases (a-c) while $N_U = 20$ in cases (d). We can see from the figure that the proposed inter-layer techniques always outperforms the other two opponents. Moreover, scheme B outperforms scheme A as predicted in theory, and the improvement is generally larger than the improvement exhibited by scheme A over the intra-layer scheme. As the network size increases, the gap to the upper bound of 1 increases for all the schemes. We believe that is due to the fact that as the number of sinks increases, intuitively, the upper bound of 1 becomes looser since it is more difficult to satisfy the max-flow value for all the sinks. Comparing Figure 6.2(a) and Figure 6.2(d), we find that although each sink connects to the same number of intermediate nodes, the overall throughput will be larger in the cases with more intermediate nodes. This can be explained as fewer intermediate nodes means relatively more sink nodes rely on each intermediate node, thus increases the probability of the conflict that different sinks request the same link to transmit different data.

The performance comparison also shows the huge advantage of network coding based multicast schemes over the scheme without network coding, which reinforces the merit of using network coding in practical system.

6.3 Performance Comparison of PSNR-maximized Layered Multicast Schemes

In this section, we compare the performance of all candidate schemes for multicasting a H.264 SVC [42, 43] confined scalable video stream, generated by the JSVM 9.15 codec [44]. A simple IPPP coding structure is used in the experiments. We encode the "Foreman" video sequence (CIF) with 300 frames at a frame rate of 30fps. The video sequence is transmitted during a single transmission slot of duration $\delta = 10$ seconds. By enabling the median grain scalability (MGS) [45] feature in JSVM, we can get a scalable video stream with fine quality scalability. Notice that H.264 SVC supports the division of the bitstream into scalable data layers only at certain points. Out of the whole set of possible division points we have selected a subset such that the size of each scalable data layer to be approximately 300 kbits (i.e., 30 kbits/second). The average rate-PSNR curve of the scalable codestream is shown in Figure 6.3. The points marked with a diamond correspond to (r,PSNR(r)) pairs for the selected division points in the scalable bitstream, where $r \times \delta$ is the length in kbits of the prefix up to the division point. To generate the source segments $x_1, x_2, \cdots, x_M$, the scalable bitstream is divided into equal sized segments of S kbits each. The value of

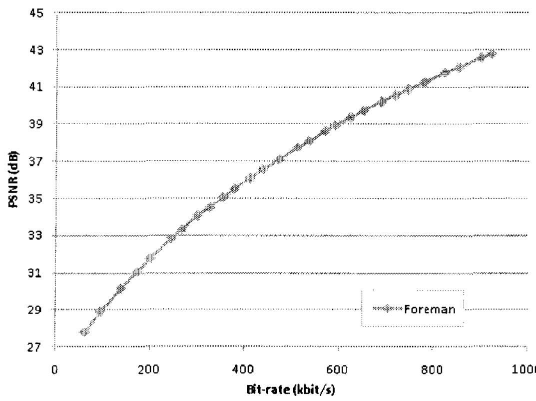


Figure 6.3: Rate-PSNR curve used in the experiments.

the fidelity function $\phi(R)$ used in the optimization problem is computed as the PSNR achieved after decoding all scalable data layers wholly included in the prefix of size $S \times R$ kbits. Precisely, if $r_1 < r_2 < \cdots < r_Q$ are the rates in kbits/second of the division points, then $\phi(R) = PSNR(r_{q_0})$, where $r_0 = \max\{q|r_q\delta \leq SR\}$.

We first compare the rate-maximized solution (i.e., where $\phi(R) = R$) with the PSNR-maximized solution (where the fidelity function is PSNR) of the proposed layered multicast scheme. In both cases the one sink-per subset partition is used (i.e., scheme B). The performance measure is the average PSNR at the sink nodes. Figure 6.4 plots the average PSNR for the PSNR-maximized solution and for the rate-maximized solution, when N_U=10, C_1=320 kbit/s, C_2=320 kbit/s, P=50% and S = 300 kbits. The comparison results in Figure 6.4 show that maximizing PSNR directly always outperforms the rate maximization approach in terms of reconstruction fidelity.

Next, we compare the PSNR-maximized solutions for all candidate schemes in the same network configuration (N_U=10, C_1=320 kbit/s, C_2=320 kbit/s, P=50%) and for the same source size (S = 300 kbits). The results in Figure 6.5, show that the relative performance between the candidate schemes is similar to that exhibited by

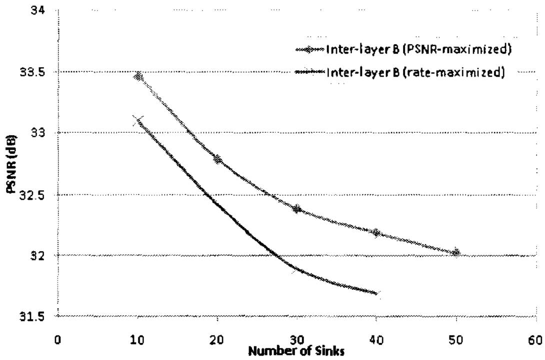


Figure 6.4: PSNR comparison between the PSNR-optimized and rate-optimized solutions for Inter-layer scheme B.

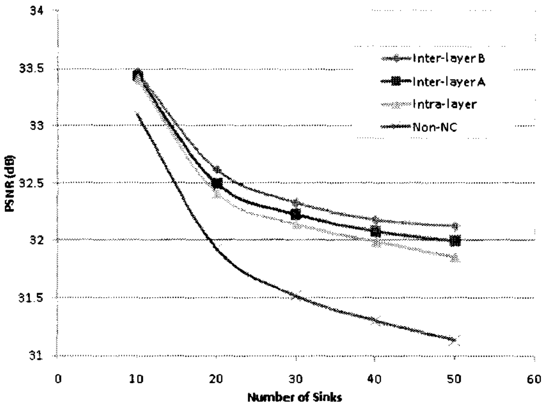


Figure 6.5: Average PSNR of different layered multicast schemes.

the rate-maximized solutions. Precisely, the proposed inter-layer schemes are always superior to the intra-layer scheme, and the network coding based schemes greatly outperform the scheme without network coding. Moreover, inter-layer scheme B is always superior to scheme A. As the network size increases, the achieved average PSNR of all schemes decreases. On the other hand, the inter-layer scheme B has the lowest decrease rate.

Figure 6.6 presents the performance of Inter-layer scheme B for different source segment sizes $S = 150, 300, 450$ kbits. The network is configured with N_U=10,

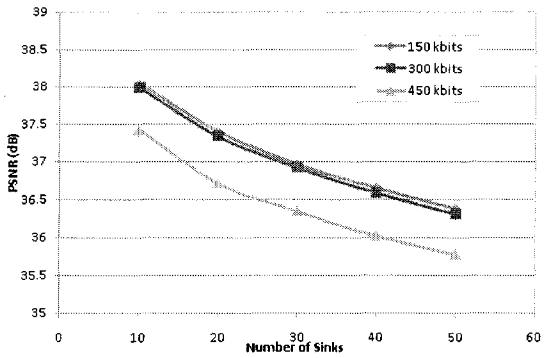


Figure 6.6: Performance of Inter-layer scheme B for different source segment sizes.

C_1=480 kbit/s, C_2=480 kbit/s, P=50%. The test results show that increasing the source segment size S above the approximate size of scalable data layer may lead to significant performance degradation. This was expected since in such a case the transmission scheme does not take full advantage of the bitstream scalability. On the other hand, reducing the source segment size below the size of the scalable data layer improves the performance since the capacity constraints in the optimization problem become more relaxed, but the improvement is very slim.

Chapter 7

Conclusions

Network coding, the new promising paradigm of network communication, is shown to be able to greatly improve throughput over traditional routing. In this work, we have studied the scalable multicast problem in the network coding scenario. We showed that those traditional routing methods reduce to a special case in the new network coding context.

We have proved that the scalable multicast problem is NP-hard, even with the ability to perform network coding at the network nodes. Several approximation problems are derived based on different heuristics, and systematic approaches have been devised to solve those problem.

Two important frameworks usually found in traditional scalable multicast solutions, i.e. layered multicast and rainbow multicast, are studied and extended to the network coding scenario. Solutions based on these two frameworks are also presented and compared. Surprisingly, these two distinctive approaches in the traditional sense become connected and share a similar essence of data mixing in the light of network coding. Cases are presented where these two approaches become equivalent and achieve the same performance.

We have made significant advances in constructing good solutions to the scalable multicast problem by solving various optimization problems that formulated in our approaches.

In the layered multicast framework, we started with a straight-forward extension of the traditional layered multicast to the network coding context. The proposed method features an intra-layer network coding technique which is applied on different optimized multicast graphs. Later on, we further improved this method by introducing the inter-layer network coding concept. By allowing the network coding among data from different data layers, more leverage is gained when optimizing the network flow, thus higher performance is achieved.

In the rainbow multicast framework, we choose uneven erasure protection (UEP) technique as a practical way of constructing balanced MDC, and optimize this MDC design using the max-flow information of receivers. After the MDC design is finalized, a single linear network broadcast code is employed to deliver MDC encoded data to receivers while satisfying the individual max-flow of all the receivers. Although this rainbow multicast based solution may sacrifice the performance in some cases, it greatly simplifies the rate allocation problem raised in the layered multicast framework. The use of one single network code also makes the network code construction process a lot clearer.

Simulation results show that the network coding based scalable multicast solutions can significantly outperform those traditional routing based solutions. In additional to the imaginary linear objective function used in the simulation, the practical convex objective function and real video data are also used to verify the effectiveness of the proposed solutions. The role of different parameters in the proposed approaches are analyzed, which gives us more guidelines on how to fine-tune the system.

However, the scalable multicast problem is still open. Some of the possible future directions include:

- In the layered multicast framework, find better initial orderings among sinks, which do not necessarily agree with their max-flow value. Besides, our current approach performs network coding across different data layers but still within each multicast layer. Further relaxing this constraint and allowing network coding among different multicast layers is expected to have greater potential to increase the performance.

- In the rainbow multicast framework, we only discussed UEP approach as a way to construct balanced MDC in this dissertation. Finding a systematic way to design more sophisticated MDC (i.e. unbalanced MDC) according to the network structure is also a promising direction.

Bibliography

[1] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656,, 1948.

[2] J. A. T. T. M. Cover, *Elements of information theory.* John Wiley and Sons, 2006.

[3] T. Berger, *Rate distortion theory: a mathematical basis for data compression.* Prentice-Hall, 1971.

[4] J. O. R. Ahuja, T. Magnanti, *Network flows: theory, algorithms, and applications*, united states ed ed. Prentice Hall, February 1993.

[5] K. S. Christos H. Papadimitriou, *Combinatorial optimization: algorithms and complexity.* Courier Dover Publications, 1998.

[6] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, September 2001.

[7] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[8] S.-Y. R. Li, R W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, pp. 371–381, 2003.

[9] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, October 2003.

[10] P. Sanders, S. Egner, P. Chou, M. Effros, S. Egner, K.Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[11] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of IEEE International Symposium on Information Theory*, June 2003.

[12] T. Ho, D. R. Karger, M. Medard, and R. Koetter, "Network coding from a network flow perspective," in *Proceedings of the 2003 IEEE International Symposium on Information Theory*, June 2003.

[13] R. W. Yeung, S. R. Li, N. Cai, and Z. Zhang, *Network coding theory.* now Publishers, 2005.

[14] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, pp. 2745–2759, 2005.

[15] Z. Li and B. Li, "Network coding: the case of multiple unicast sessions," in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2004.

[16] C. Wang and N. Shroff, "Intersession network coding for two simple multicast sessions," in *Proc. Annual Allerton Conference on Communication, Control, and Computing,*, Sept. 2007.

[17] Y. Wu, "On constructive multi-source network coding," in *Proc. IEEE ISIT*, Seattle, WA, July 2006.

[18] M. Yang and Y. Yang, "A linear inter-session network coding scheme for multicast," in *Network Computing and Applications, 2008. NCA '08. Seventh IEEE International Symposium on*, July 2008.

[19] D. MacKay, "Fountain codes," *Communications, IEE Proceedings*, vol. 125, pp. 1062 – 1068, Dec. 2005.

[20] P. Cataldi, M. Grangetto, T. Tillo, E. Magli, and G. Olmo, "Sliding-window raptor codes for efficient scalable wireless video broadcasting with unequal loss protection," *IEEE Trans. Image Process.*, vol. 19, pp. 1491–1503, June 2010.

[21] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Trans. Commun.*, vol. 57, pp. 2510 – 2516, Sept. 2009.

[22] D. VuVukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong, "Scalable video multicast using expanding window fountain codes," *IEEE Trans. Multimedia*, vol. 11, pp. 1094 – 1104, Oct. 2009.

[23] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proceedings of AXM SIGCOMM 96*, August 1996.

[24] X. Li, S. Paul, and M. Ammar, "Layered video multicast with retransmissions (lvmr): evaluation of hierarchical rate control," in *Proceedings of IEEE INFOCOM 1998*, March 1998.

[25] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multi-rate multicast sessions," in *Proceedings of IEEE INFOCOM 2001*, vol. 1, April 2001.

[26] X. Wu, B. Ma, and N. Sarshar, "Rainbow network flow of multiple description codes," *IEEE Trans. Inf. Theory*, vol. 54, pp. 4565 – 4574, 2008.

[27] N. Sarshar and X. Wu, "Rate-distortion optimized network communication," in *Proceedings of INFOCOM 2007*, May 2007, pp. 1595–1603.

[28] N. Sundaram, P. Ramanathan, and S. Banerjee, "Multirate media stream using network coding," in *Proceedings of Fourty-Third Annual Allerton Conference on Communication, Control, and Computing*, September 2005.

[29] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proceedings of INFOCOM 2007*, May 2007, pp. 1163–1171.

[30] J. Zhao, F. Yang, Q. Zhang, Z. Zhang, and F. Zhang, "Lion: layered overlay multicast with network coding," *IEEE Trans. Multimedia*, vol. 8, pp. 1021 – 1032, 2006.

[31] Y. Wu, "Distributing layered content using network coding," in *IEEE International Workshop on Wireless Network Coding, invited paper*, June 2008.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms.* The MIT Press, 2001.

[33] M. Sipser, *Introduction to the theory of computation.* Course Technology, 2005.

[34] R. J. Trudeau, *Introduction to graph theory.* Dover Publications, 1994.

[35] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity.* Dover Publications, 1998.

[36] P. Raghavan and C. D. Tompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, 1987.

[37] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, March 1963.

[38] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1737–1744, Nov 1996.

[39] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Trans. Multimedia*, vol. 6, pp. 230–239, April 2004.

[40] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 412–422, Mar 1995.

[41] S. Dumitrescu, M. Shao, and X. Wu, "Layered multicast with inter-layer network coding," in *Proceedings of Infocom 2009*, Rio de Janerio, April 2009.

[42] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560 – 576, 2003.

[43] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[44] http://ip.hhi.de/imagecom_G1/savce/downloads/SVCReference–Software.htm.

[45] *JSVM software manual.*