Layered Multicast with Inter-layer Network Coding for Multimedia Streaming

Mingkai Shao
Department of Electrical and Computer Engineering
McMaster University
Hamilton, ON, Canada
Email: mshao@mcmaster.ca
Sorina Dumitrescu
Department of Electrical and Computer Engineering
McMaster University
Hamilton, ON, Canada
Email: sorina@ece.mcmaster.ca
Xiaolin Wu
Department of Electrical and Computer Engineering
McMaster University
Hamilton, ON, Canada
Email: xwu@ece.mcmaster.ca

Abstract

Multirate multicast is a powerful methodology of multimedia communication in heterogenous networks. A variant of multirate multicast motivated by scalable multimedia streaming is layered multicast, where the transmitted signal is presented in successive data layers. With recent advances of network coding theory, many layered multicast schemes using network coding have been proposed to improve the performance of traditional routing based layered multicast. 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. In this paper, we propose a novel approach to layered multicast that allows network coding of data in different layers. This relaxation lends the proposed scheme greater flexibility in optimizing the data flow than previous layered solutions, and thus achieves higher throughput.

I. INTRODUCTION

Multirate multicast is an efficient network transmission methodology for streaming a multimedia content to multiple heterogenous clients. Unlike the conventional unirate multicast that either overwhelms low-bandwidth receivers or starves high-bandwidth ones, multirate multicast delivers a content at different rates to receivers of different network capacities. As a result, every client enjoys a multimedia session at a quality according to his/her available bandwidth.

Layered multicast is a variant of multirate multicast that is suited for scalable multimedia streaming. A scalable encoded 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, hence each node can only send copies of received messages, 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 [1]–[3] focus on the construction of multiple multicast trees, each representing a multicast layer.

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 [4]. 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 [4]. Moreover, optimal unirate multicast network code can be constructed in polynomial time [5].

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 [6], and finding the optimal network code for a given network, which maximizes the total amount of information flow, is proved to be NP-hard [7].

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.

In this paper, we propose a novel approach to layered multicast that allows network coding of data in different layers. This relaxation lends the proposed scheme greater flexibility in optimizing the data flow than previous layered solutions, and thus achieves higher throughput. Note that, network coding is applied here to increase network throughput rather than protecting again packet loss.

The rest of the paper is organized as follows. Some related works are discussed in Section II. The formulation of the target problem is described in Section III. Section IV introduces our new inter-layer network coding flow optimization scheme. The construction of the layered multicast network code is presented in Section V. Some experimental results are shown in Section VI. Section VII concludes the paper.

II. Related Works

A number of previous works applied networking coding in the layered multicast setting [7]–[9], 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 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 [5]. Specifically, in [8], 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. The algorithm in [7] assigns sinks to multicast layers in the same way, but without constructing a specific multicast sub-graph for each layer, thus has more freedom in optimizing the rate allocation between layers. In [9], the authors resort to layered multicast and session scheduling ideas to provide rate control in multirate multicast.

In [10], an interesting framework of jointly designing uneven erasure protection (UEP) [11] and linear network codes for transmission of layered data is presented. Unlike previous network coding-based layered multicast schemes, the proposed technique does not explicitly divide the transmission into multicast layers. Moreover, the computationally expensive rate allocation problem is reduced to an UEP optimization problem, which can be solved efficiently [12]. However, this scheme shares the main feature of previous work: the use of network coding only inside a data layer and not across data layers.

As pointed out in the above discussion, prior work on network coding-based layered multicast uses intra-layer network coding. 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 we propose a layered multicast framework with inter-layer network coding, which is the main contribution of this paper. To perform inter-layer network coding, we still divide the transmission into multicast layers, but the concept of a multicast layer in our framework is significantly different from previous work. As opposed to previous work, 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 other words, each multicast layer is not unirate. Moreover, we allow flow to transfer from one layer to a higher 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.

Independently, Wu [13] proposed a cross-layer network coding technique which is closely related to our work. However, the problem formulation in [13] 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 [13] uses simple random linear mixing of asymptotical optimality.

Another related work is scalable multicast using the variants of digital fountain codes [14]–[16]. In [16], the authors proposed an application of sliding window (SW) Raptor codes on scalable video coding in 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. Since the coding is performed within each SVC layer, this is an intra-layer approach. [14], [15] design and use the optimal expanding window fountain (EWF) code to multicast scalable multimedia content through a lossy networks, such that given quality-of-service guarantees for different receiver classes to be satisfied. Although it seems that the EWF codes already do the inter-layer coding as we proposed here, there are substantial differences. The EWF codes still require receivers to receive the same amount of data in the same multicast layer, while our proposed inter-layer coding scheme allows each multicast layer to be multi-rate. Another 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. Therefore, these scheme usually require predefined bandwidth information for the receivers rather than 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.

An earlier version of this work appeared in [17]. Compared to [17], the current version further improves the algorithm performance by refining the initial partition of sinks. The optimization formulation is also extended to handle a general fidelity measure as the objective function. The new formulation allows us to optimize the average PSNR, aspect which is very important for practical multimedia applications, while the transmission scheme in [17] can only optimize the average rate received at all sinks. Moreover, while [17] does not report test result on practical multimedia data, the current work includes experimental results on a practical scalable video sequence, showing the advantage of using the PSNR optimization versus the rate optimization.

III. Problem Statement

Consider source segments (or messages) $x_1, \cdots, x_M$, of equal size, and a directed acyclic lossless network $G = (V, E)$, a set of source nodes S, and a set of sink nodes T. All data messages $x_1, \cdots, x_M$ are available at all source nodes, for transmission during a transmission slot. Each edge is assigned a capacity representing the number of source segments which can be transported over the edge during a transmission slot. Thus, a source segment is treated as an indivisible flow unit. A transmission scheme using network coding allows each message sent along an edge to be a function of the messages received at the node where the edge originates. As other previous work on network coding we assume that there is no transmission delay along the edges and that each node has a buffer large enough to store the received messages as long as needed in order to construct the messages to be sent over the emerging edges.

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 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 work is designing a layered multicast code such that $\sum_{t \in T} \phi(R(t))$ is maximized.

To make clearer the "layered" characteristic, we give an equivalent formulation to this problem. 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.

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 1.

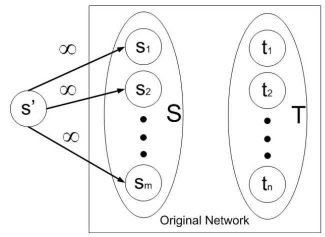


Fig. 1. Converting a problem with multiple sources to one with single source.

IV. Flow Optimization for Inter-layer Network Coding

A. Preliminaries

Solving Problem 1 over all possible sinks partitions is a difficult task. 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 $\mathcal{T}_{\text{max-flow}}$. The partition $\mathcal{T}_{\text{max-flow}}$ was used in prior works [7], [8]. Therefore, imposing this constraint will not lead to loss in performance versus the techniques in [7], [8].

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).

Previous layered multicast formulations using network coding [7]–[10] 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 inter-layer scheme we analyze the example shown in Figure 2. 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 has 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

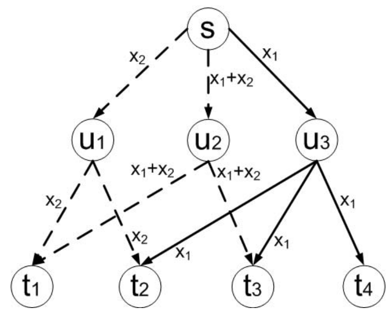


Fig. 2. Optimal layered multicast solution for the given network.

layered multicast solution shown in Figure 2 (which in this case is the optimal multirate multicast solution as well) cannot be achieved.

The above example was first used in [18] 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 2 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.

Our proposed solution to Problem 2 consists of two major parts. First a flow optimization problem is solved. Next a single network code is constructed to achieve the optimal rates for all sinks.

B. *Inter-layer Flow Optimization*

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 segment 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 to equal 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 3.

Constraint (1c) follows from the definition of node potential. (1d), (1e) are the potential constraints at the source node. The total flow sent out from the 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 (1f), (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 the 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.

(1h), (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 (1j) is the network coding constraint, meaning that the flow for different sinks in the same layer can be combined together. Constraint (1k) confines that the actual flow in each edge cannot exceed the edge capacity.

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

Fig. 3. Inter-layer flow optimization.

Finally, notice that inequality (1b) follows from (1e) since the sinks potentials are nonnegative, thus (1b) can be removed.

C. 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 4. 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. Edges in the first layer are

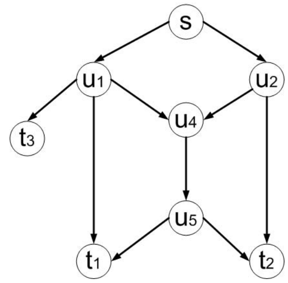


Fig. 4. Example network with 3 sinks.

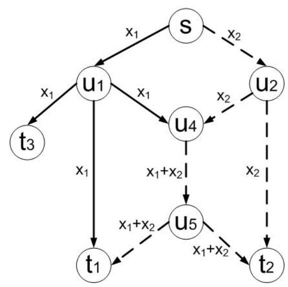


Fig. 5. Solution produced by the inter-layer network coding scheme. This is the optimal layered multicast solution for the given network.

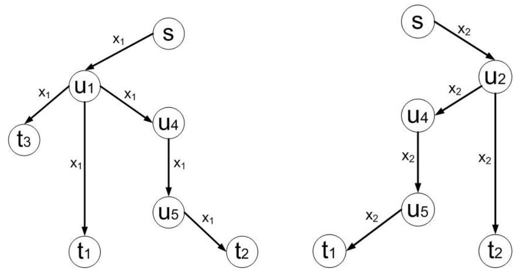
shown as solid lines, while edges in the second layer are shown as dashed lines.

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 6. 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.

D. *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 [7], [8]. 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 (1d), (1f), (1h) into equalities, we will obtain the exact formulation of layered multicast in [7].

Another notable observation is that the solution of problem in Figure 3 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



(a) Layer 1 (b) Layer 2

Fig. 6. Optimal layered multicast solution for network in Figure 4 cannot be achieved with the intra-layer network coding technique unless edge (u_4, u_5) has capacity 2.

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 [17], it is expected that we can achieve better performance than in [17].

E. Linearization of Flow Optimization Problem

A simple choice for the fidelity function is $\phi(R) = R$, as considered in [17]. Then the problem of Figure 3 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 the 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

$$\phi(R) = \phi(0) + \sum_{j=1}^{R} c_j \times 1 + \sum_{j=R+1}^{M} c_j \times 0, \tag{2}$$

for any integer R with $1 \leq R \leq M$. To linearize the flow optimization problem we introduce the 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

$$\max \quad \sum_{k=1}^{N}\sum_{j=1}^{M} c_j r_{k,j} \tag{4a}$$

$$\text{such that} \quad \sum_{j=1}^{M} r_{k,j} = R_k, \quad 1 \le k \le N \tag{4b}$$

$$r_{k,1} \ge r_{k,2} \ge \cdots \ge r_{k,M}, \quad 1 \le k \le N \tag{4c}$$

$$r_{k,j} \in \{0,1\}, \quad 1 \le k \le N, \quad 1 \le j \le M \tag{4d}$$

$$\text{conditions (1b)-(1l) hold} \tag{4e}$$

Fig. 7. Linearization of the flow optimization problem.

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

$$\phi(R_k) = \phi(0) + \sum_{j=1}^{M} c_j r_{k,j} \tag{3}$$

and the optimization problem can be recast as in Figure 7. Notice that condition (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.

V. 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.

A. 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 $1 \le l \le N$. 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 (1k). Moreover, some of these sets may be empty. In particular, the l-th set is empty if $y_{i,j}^{l} = 0$.

Upon this conversion we construct for each sink $t \in T$ a set of $R_{L(t)}$ edge-disjoint paths $Q_t^1, \cdots, Q_t^{R_{L(t)}}$ 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 $Q_t^1, \cdots, Q_t^{R_{L(t)}}$ are constructed such that the following conditions to be 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_t^i is in a lower or the same layer as the first edge in Q_t^{i+1}.

The existence of such paths satisfying the above requirements is ensured by the constraints imposed on the node potentials (1d-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 the set of sinks using e in the flow paths.

Algorithm 1, which is inspired by the LIF algorithm [5], 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.

for *each* $e \in E$ **do**
  Set $f(e)$ = $[0^{R_N}]$;
end
for *each* $t \in T$ **do**
  Construct $R_{L(t)}$ edge-disjoint paths $\{Q_t^1, \cdots, Q_t^{R_{L(t)}}\}$ from s to t such that conditions C1-C3 are satisfied;
end
Insert a super source s' into V
for *each* $t \in T$ **do**
  Add $R_{L(t)}$ parallel imaginary edges $\{e_t^1, \cdots, e_t^{R_{L(t)}}\}$ 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 $C_t = \{e_t^1, \cdots, e_t^{R_{L(t)}}\}$; —(*)
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 $f_j(e) = 0$ for all $j, R_{L(e)} + 1 \leq j \leq R_N$, and $\forall t \in T(e)$, $f(e)$ is linearly independent of $\{f(c) : c \in C_t \setminus \{\phi_t(e)\}\}$; —(**)
    for *each* $t \in T(e)$ **do**
      $C_t = (C_t \backslash \{\phi_t(e)\}) \cup \{e\}$;
    end
  end
end

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 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 $C_t \subseteq In(t)$.

B. *Proof of Correctness*

The correctness of Algorithm 1 follows from the following lemmas. In order not to disrupt the flow of the exposition we defer their proof to the appendix.

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,

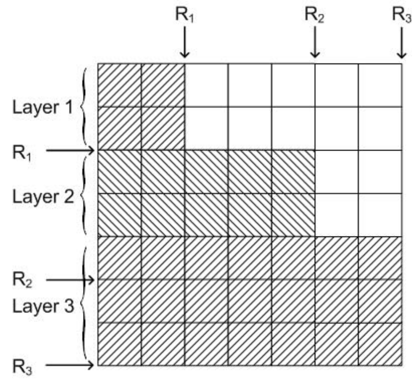


Fig. 8. Example of data allocation among layers for a sink in layer 3.

the invariant holds at the initialization step.

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

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.

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 8. The example in Figure 8 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.

C. 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

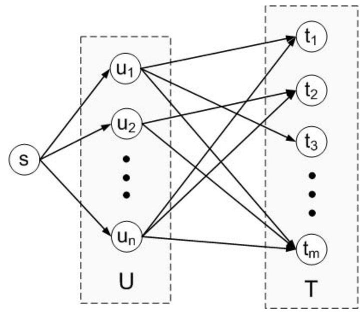


Fig. 9. Network model used in simulations.

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.

VI. Performance Evaluation

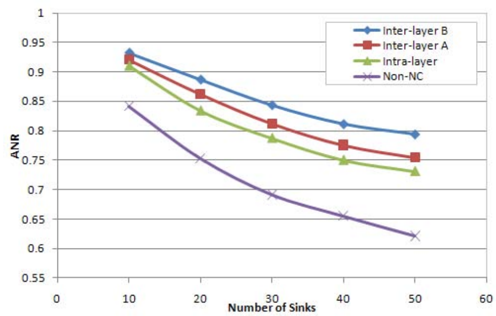
This section contains some simulation results of the proposed inter-layer network coding scheme. In the simulations, we consider a family of networks which were first introduced in [5]. In this network model, all the sinks are connected to the source through a group of intermediate nodes (as the network shown in Figure 9). 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 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.

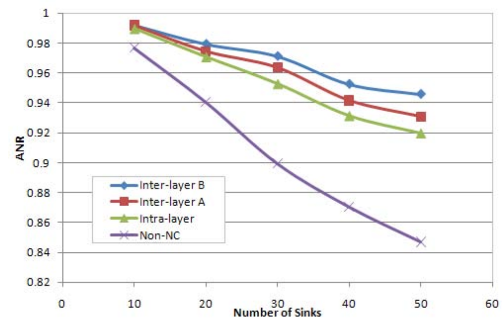
We compare the performance of the proposed layered multicast scheme with inter-layer network coding versus the layered multicast with intra-layer network coding of [7] and a simple layered multicast without network coding. We also test the impact of refining the sink partition $\mathcal{T}$, therefore we consider two cases for the proposed scheme: 1) $\mathcal{T} = \mathcal{T}_{\text{max-flow}}$, as proposed in [17]; 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 IV-D, we expect for scheme B to achieve a higher performance than scheme A, and both to outperform the other two methods.

A. *Performance Comparison of Rate-maximized Layered Multicast Schemes*

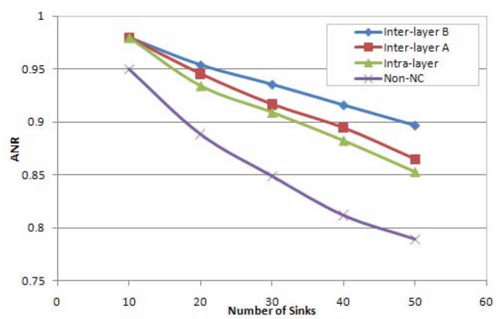
We start with the performance comparison of all candidate schemes using the rate as fidelity function in the flow optimization problem. In other words we compare the solutions which maximize the overall received flow. The



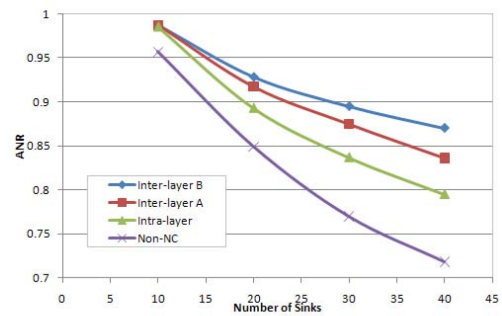
(a) N_U=10, C_1=320 kbit/s, C_2=320 kbit/s, P=50%



(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%

Fig. 10. Average normalized rates of four different layered multicast schemes.

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 10. In Figure 10(a), $C1$ and $C2$ are both set to be 320kbit/s, while each sink node connects to 50% of the intermediate nodes. In Figure 10(b), $C1$ is changed into 640 kbit/s, twice the value of $C2$, while the connectivity remains 50%. In Figure 10(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 case (d).

We can see from the figure that the schemes using network coding always outperform the scheme without network coding. Moreover, as predicted in theory, the proposed inter-layer technique always performs no worse than the intra-layer counterpart and it exhibits a net improvement as the ratio N_T/N_U becomes sufficiently large. This can be attributed to the fact that, as N_T/N_U increases, relatively more sink nodes rely on each intermediate node, thus increasing the number of different sinks requesting the same link to transmit different data. The inter-layer scheme has more flexibility in optimizing the flow such that to counterweight this conflict, thus leading to higher throughput.

Notably, inter-layer scheme B is never worse than scheme A, as expected, and it strictly outperforms the latter scheme as N_T/N_U becomes large enough, fact which demonstrates that using a finer sink partition strictly improves the solution of the flow optimization problem when the bottleneck probability is high enough.

Another notable observation is that, as the network size increases, the gap to the upper bound of 1 increases for all the schemes. This can be explained intuitively as follows. As the number of sinks becomes higher the upper bound of 1 becomes looser since it is more difficult to satisfy the max-flow value for all the sinks. Moreover, comparing Figure 10(a) with Figure 10(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 behavior can be motivated by the fact that fewer intermediate nodes means that a higher number of sink nodes rely on each intermediate node, thus increasing the bottleneck probability.

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

B. 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 [20] confined scalable video stream, generated by the JSVM 9.15 codec [19]. 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) 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 11. 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 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 $q_0 = \max\{q | r_q \delta \leq SR\}$.

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 12 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 12 show that maximizing PSNR directly always outperforms the rate maximization approach in terms of reconstruction fidelity.

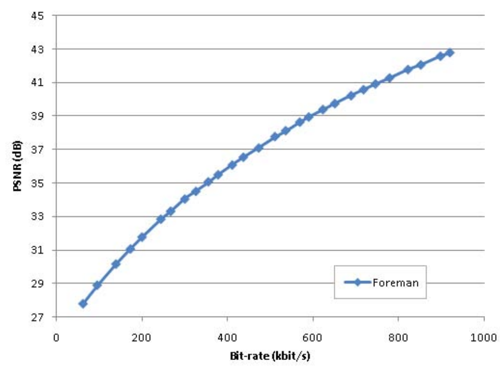


Fig. 11. Rate-PSNR curve used in the experiments.

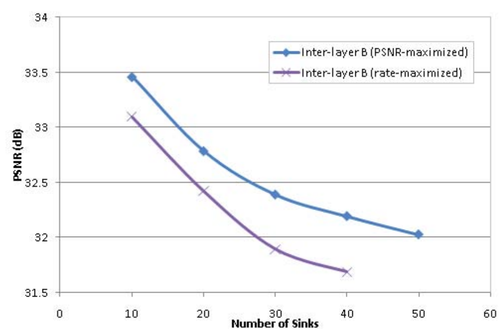


Fig. 12. PSNR comparison between the PSNR-optimized and rate-optimized solutions for Inter-layer scheme B.

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 13, show that the relative performance between the candidate schemes is similar to that exhibited by 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 14 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, 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.

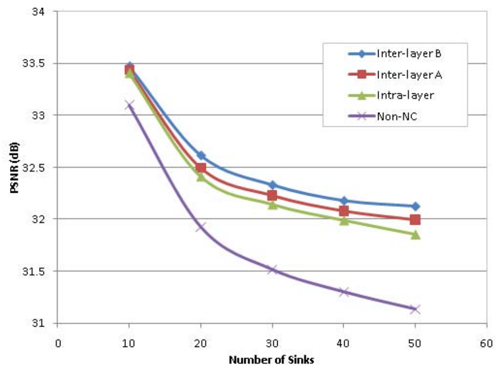


Fig. 13. Average PSNR of different layered multicast schemes.

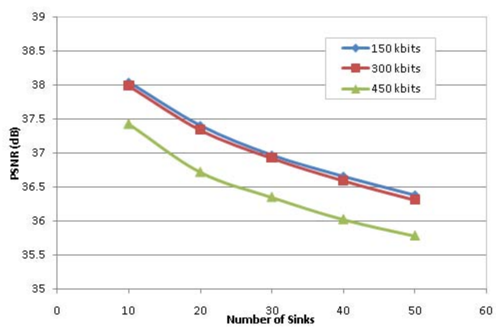


Fig. 14. Performance of Inter-layer scheme B for different source segment sizes.

C. Cost Analysis

In conclusion, our simulations have shown that the proposed inter-layer multicast scheme can outperform in throughput existing multicast solutions, and in particular, the intra-layer scheme. In this comparison we have disregarded other practical aspects like storage needs, computational complexity, delay, etc. It is expected that higher throughput to be achieved at the expense of increasing such costs. We defer a thorough analysis of the trade-offs, of methods to decrease the associated costs, as well as other aspects related to the practical application of the scheme, to future work. In the remaining part of this section we only briefly discuss the incurred costs with respect to the intra-layer scheme.

The flow optimization problem has roughly the same complexity for the intra-layer and inter-layer schemes, when the initial sink partition is the same. This stems from the fact that the number of variables and constraints in the optimization problem is the same as mentioned in section IV.D. An increment in complexity (which is difficult to evaluate analytically) occurs when the partition becomes more refined. On the other hand, if the transmission scheme is updated only after a large number of transmission slots or not updated at all, then this computational overhead is not necessarily an issue.

The complexity of the code construction algorithm is again approximately the same in the intra-layer and inter-layer case, irrespective of the sink partition, as noted in section V.C.

The decoding complexity is higher for the inter-layer scheme. If S denotes the size of a source segment in bits, then for a sink t, the decoding time amounts to $O(R^2_{L(t)}S)$ per transmission slot[1] compared to $O(S\sum_{i=1}^{L(t)}(R_i-R_{i-1})^2$ for the intra-layer counterpart. Additionally, the fact that the sink t needs to wait for all $R_{L(t)}$ messages to be received in order to start decoding, together with the fact that the decoding takes longer, leads to higher delay in the inter-layer case. To keep the delay small the transmission slot has to be small enough. Other possible methods to decrease the delay are to exploit the special form of the matrix of global encoding coefficients and perform early decoding when possible. For instance, for the example in Figure 8, the sink can decode the messages received in the first transmission layer without waiting for all messages to be received.

There are some other aspects related to the practical use of PSNR-optimized solution versus the rate-optimized solution, that we have not treated in this work. In a static network, as we have considered, the rate optimized flow does not need to be changed during the multimedia session. Since the PSNR data is dependent on the multimedia content the question arises how often should this solution be updated during a multimedia session to keep the complexity within reasonable limits[2]. This can be done periodically at fixed intervals in time, or when ”significant” changes are detected in the multimedia content.

VII. Conclusion

In this paper, we propose a novel layered multicast technique using inter-layer network coding. Compared to its predecessors that are confined to intra-layer network coding, the new technique has greater flexibility in optimizing the data flow, and thus achieves higher throughput. Future work includes finding better initial orderings among sinks, which do not necessarily agree with their max-flow value. Besides, our current work 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 another objective for our future work. Another subject to be addressed in future work is a detailed analysis of incurred costs and methods to decrease such costs.

Appendix

In this appendix the proofs of Lemmas 1-3 are presented.

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,

[1]This is the number of operations needed to multiply the inverse of the $R_{L(t)}\times R_{L(t)}$ matrix of global encoding coefficients with the $R_{L(t)}\times S/B$ matrix formed out of the $R_{L(t)}$ received messages, where B is the number of bits needed to represent a symbol in the field F. Notice that each message can be regarded as an S/B-dimensional row vector of elements in the field F. The overhead in running time of $O(R^3_{L(t)})$ needed to invert the matrix of global encoding vectors becomes negligible if the number of transmission slots using the same network coding scheme is high enough.

[2]The PSNR-wise flow optimization has higher complexity than the rate-wise optimization for both the intra-layer and inter-layer schemes.

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 $k, 1 \le k \le R_{L(t)}$, 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 (1d) and (1e), we have

$$\sum_{i=1}^{j} n(t,i) \le R_j, \quad 1 \le j < L(t) \tag{5a}$$

$$\sum_{i=1}^{j} n(t,i) = R_j, \quad j = L(t) \tag{5b}$$

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 $R_{k-1} < n \le R_k$. 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 [5]. 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

$$f(e) = \sum_{e' \in P(e)} k_e(e') f(e'). \tag{6}$$

Since all flow paths satisfy conditions C2 it follows that $L(e') \le L(e)$ for all $e' \in P(e)$. By the invariant, we have $f_j(e') = 0$ for all $j, R_{L(e')} + 1 \le j \le R_N$. Hence, (6) will further ensure that $f_j(e) = 0$ for all $j, R_{L(e)} + 1 \le j \le R_N$.

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 $L(c) \le L(t)$ 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 (6) still holds if f is replaced by f', i.e.

$$f'(e) = \sum_{e' \in P(e)} k_e(e') f'(e'). \tag{7}$$

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. □

References

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

[2] X. Li, S. Paul and M. Ammar, "Layered video multicast with retransmissions (LVMR): evaluation of hierarchical rate control," *Proceedings of IEEE INFOCOM 1998*, Volume 3, 29 March-2 April 1998.

[3] K. Kar, S. Sarkar and L. Tassiulas, "Optimization based rate control for multirate multicast sessions," *Proceedings of IEEE INFOCOM 2001*, Volume 1, 22-26 April 2001.

[4] 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.

[5] 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*, Volume 51, Issue 6, June 2005 Page(s):1973 - 1982.

[6] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "An algebraic approach to network coding," *IEEE Trans. Networking*, vol. 11, issue 5, pp. 782-795, October 2003.

[7] M. Shao, X. Wu and N. Sarshar, "Rainbow network flow with network coding," in *Proceedings of NetCod 2008*, January 2008.

[8] N. Sundaram, P. Ramanathan, and S. Banerjee, "Multirate media stream using network coding," in *Proceedings of Allerton 05*, September 2005.

[9] 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.

[10] M. Shao, S. Dumitrescu, and X. Wu, "Toward the Optimal Multirate Multicast for Lossy Packet Network," *Proceedings of ACM Multimedia 08*, 27-31 October 2008.

[11] A. Albanese, J. Blomer, J. Edmonds, M. Luby, M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, Volume 42, Issue 6, Part 1, Nov. 1996 Page(s):1737 - 1744.

[12] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Trans. Multimedia*, vol.6, April 2004, Page(s):230 - 239.

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

[14] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, R. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Trans. Communications*, vol.57, Sept. 2009, Page(s):2510 - 2516.

[15] D. VuVukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, Z. Xiong, “Scalable Video Multicast Using Expanding Window Fountain Codes,” *IEEE Trans. Multimedia*, vol.11, Oct 2009, Page(s):1094 - 1104.

[16] P. Cataldi, M. Grangetto, T. Tillo, E. Magli, G. Olmo, “Sliding-Window Raptor Codes for Efficient Scalable Wireless Video Broadcasting With Unequal Loss Protection,” *IEEE Trans. Image Processing*, vol.19, no.6, pp.1491-1503, June 2010

[17] S. Dumitrescu, M. Shao and X. Wu,“Layered multicast with inter-layer network coding,” *Infocom 2009*, Rio de Janerio, April 2009, pp.442-449.

[18] R. W. Yeung, “Multilevel diversity coding with distortion,” *IEEE Trans. Inf. Theory*, Volume 41, Issue 2, Mar. 1995 Page(s):412 - 422.

[19] ”http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm”

[20] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, ”Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Scalable Video Coding, Vol. 17, No. 9, pp. 1103-1120, September 2007.