Multi-Layer Video Streaming with Helper Nodes using Network Coding

Pouya Ostovari*, Abdallah Khreishah†, and Jie Wu*
*Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
†Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102

***Abstract*—Video streaming is one of the dominant forms of traffic on the Internet. This increases workload on the video servers, which leads to substantial slowdowns. In order to resolve the slowdown problem, and to provide a scalable and robust infrastructure to support on-demand streaming, helper-assisted video-on-demand (VoD) systems have been introduced. In this architecture, helper nodes, which are micro-servers with limited storage and bandwidth resources, download and store the user-requested videos from a central server to decrease the load on the central server. Multi-layer videos, in which a video is divided into different layers, can also be used to improve scalability. In this paper, we study the problem of utilizing the helper nodes to minimize the pressure on the central servers. We formulate the problem as a linear programming (LP) optimization using joint inter- and intra-layer network coding (NC). We show that a lightweight triangular inter-layer NC can be used, instead of the general form of inter-layer NC, to achieve the optimal solution. Our solution can also be implemented in a distributed manner. We show how our method can be extended to the case of wireless live streaming, in which a set of videos is broadcast. We carefully study the convergence and the gain of our distributed approach.**

***Index Terms*—Video-on-demand (VoD), streaming, multi-layer video, intra-layer coding, inter-layer coding, wireless network.**

I. Introduction

As the requirements of life and technology change, people use real-time and multicast services, such as video streaming and video conferencing, more. Recent studies have shown that multimedia streaming produces the most traffic on the Internet. For example, 20-30% of the web traffic on the Internet is from YouTube and Netflix [1], [2]. Thousands of hours of video are uploaded on YouTube every day, and millions of hours of movies are available on Netflix, Hulu, and iTunes sites. Another application that is becoming an integral part of our lives is surveillance to provide public security, which requires real-time and multicast networking services.

In order to provide a scalable and robust infrastructure that will support large and diverse on-demand streaming, the concept of helpers has been introduced, and the design of *helper-assisted video-on-demand* (VoD) systems has been explored [3]–[7]. Helper nodes are micro-servers with limited storage and bandwidth resources, which can download and store the requested videos and can provide users with their requests. The helper nodes work in conjunction with a central server, which provides the users with the video files that cannot be obtained from their neighboring helper nodes (Fig. 1). It is obvious that the central server will be able to serve more users, if we can provide more portions of the requested videos through the helpers.

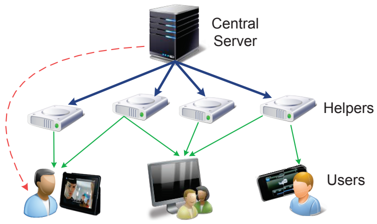


Fig. 1. The system architecture.

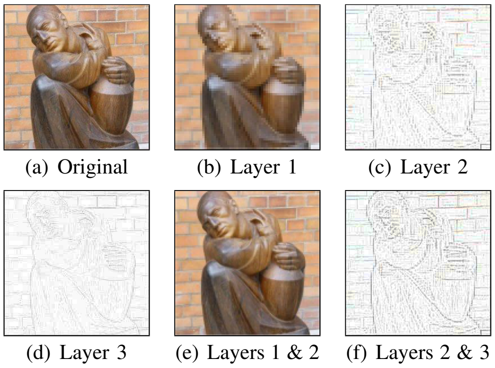

(a) Original (b) Layer 1 (c) Layer 2
(d) Layer 3 (e) Layers 1 & 2 (f) Layers 2 & 3

Fig. 2. Multi-layer video with 3 layers.

In addition to the use of helper nodes, *multi-layer videos* [8]–[10] can be used to provide a higher degree of scalable VoD systems. In multi-layer video, which is also called *multi-resolution codes* (MRC), videos are typically divided into a base layer and enhancement layers [9], [11]. The base layer (layer 1) is required to watch the video, but the enhancement layers augment the quality of the video streaming. Accessing more layers provides higher video quality, but the i-th enhancement layer is not useful unless the user has access to all of the enhancement layers with a smaller index. Fig. 2(a) shows an original image, and Figs. 2(b)-(d) show the constructed layers from this image. Layer 1 is the most important layer, which is required by all of the users. Layers 2 or 3 cannot be used without all of the layers with a smaller index, as depicted in Figs. 2(c) and (d). Fig. 2(f) shows that adding layers 2 and 3 together without layer 1 is useless, as well. Adding layer 2 to layer 1 increases the quality of the image as shown in Fig. 2(e).

In order to use the resources optimally, we need a mechanism to distribute the packets of the videos on the helper nodes, since the helpers might not be able to store a full copy

This research was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

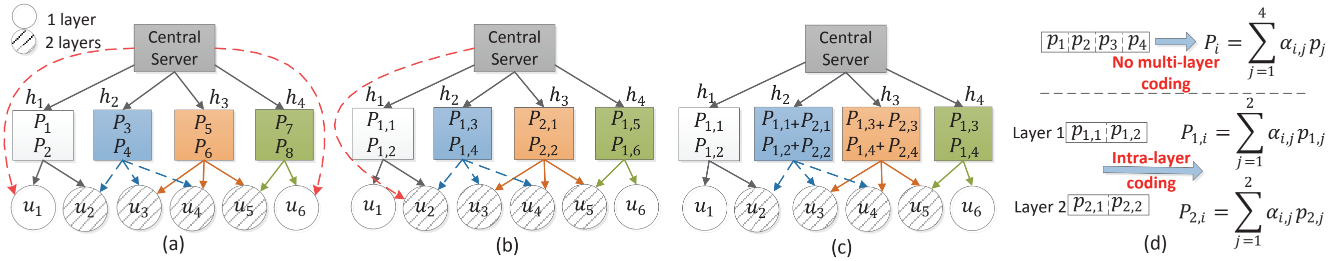


Fig. 3. The advantage of using NC, (a) No multi-layer NC, (b) Intra-layer NC, (c) Inter- & intra-layer NC, (d) Coding schemes.

of the videos due to the storage limitations. Network coding (NC) [12], [13] helps to simplify the content distribution problem and solves it in an efficient way. Consider packets $p_1, ..., p_n$. In *random linear NC*, each coded packet is in the form of $\sum_{i=1}^{n} a_i \times p_i$, where a_i is a random coefficient. In this scheme, if a user has access to any n linearly independent coded packets, it can use Gaussian elimination to decode the coded packets and retrieve the original packets. In [14], it is shown that randomly selecting the coefficients guarantees that the packets will be linearly independent. As a result of random linear NC, the coded packets contribute the same amount of data to the users, which simplifies the distribution of the packets. Linear NC can be classified into *intra- or inter-layer NC*, depending on whether the coding is performed between the packets from the same layer or different layers, respectively.

Consider Fig. 3, in which the users request a two-layer video, each of them consisting of 2 packets. The capacity of the helper nodes is equal to 2 packets. Assume that users u_1 and u_6 request layer 1, and the other users need both layers. Using the proposed method in [3], the whole video should be downloaded for playing, as the method does not support multi-layer coding; thus, the video is considered to be 4 packets p_1-p_4. Fig. 3(a) shows an optimal video placement option based on the proposed method in [3], in which random linear coded packets of p_1-p_4 are stored on the helpers (the no multi-layer NC is depicted in Fig. 3(d)). In this case, users u_2-u_5 have access to 4 coded packets over p_1-p_4; thus, they can decode the coded packets using just the helper nodes. However, users u_1 and u_6 need to download 2 more packets from the server to decode the coded packets.

Fig. 3(b) shows an optimal placement using intra-layer NC. The coding structure is shown in Fig. 3(d). In this case, only user u_2 needs to download 2 packets from the server, so the load on the server is less than that of in Fig. 3(a). Inter-layer NC can be used in conjunction with intra-layer NC to increase the efficiency of the content placement on the helpers. In Fig. 3(c), we benefit from inter-layer NC. Users u_2-u_5 have access to 4 linearly coded packets over layers l_1 and l_2, so the central server does not need to upload any layer. Moreover, users u_1 and u_6 have access to 2 linearly coded packets over layer l_1, which is sufficient for decoding the first layer.

Motivated by the intuition drawn from the example, in this work, we answer the following questions: how should the packets of videos be distributed on helper nodes? how should the helper nodes allocate their bandwidth to the users to minimize the load on the central server? And, lastly, which coding scheme should be used for the content placement? While answering these questions, we have the following contributions:

- In contrast to previous works, which study the case of single video [6], [7] or no multi-layer videos [3], we study multi-layer multi-video streaming, and characterize the optimal solution using linear programming (LP).
- The problem of inter-layer NC is typically considered as an NP-complete problem [9], [11]. However, in this work, we come up with a setting where the optimal solution of the problem can be calculated using joint inter- and intra-layer NC in polynomial time. We also present a distributed approach to optimally utilize the helper nodes, which adapts to the system dynamic.
- We show that a lightweight *triangular inter-layer* NC can be used instead of the general form of inter-layer NC to achieve the optimal solution. Moreover, we empirically show the cases under which combining inter- with intra-layer coding provides benefit over intra-layer coding.

The remainder of this paper is organized as follows: In Section II, we introduce the settings. We formulate the problem for the case of wireless or wired VoD in Section III, and study the wireless live streaming application in Section IV. We introduce our distributed optimal solution in Section V, and evaluate our methods through simulations in Section VI. Section VII concludes the paper.

II. Setting

Consider a wireless VoD system, where a central server provides a set of videos to users with the help of a group of helper nodes. Helpers are micro-servers with limited storage and bandwidth resources. We represent the set of helpers, users, and videos as H, U, and M, respectively. The users are stationary, and each helper covers a subset of the users. The k-th video m_k has a constant streaming rate r_k and size v_k. User u_i has a stationary request, denoted as q_i, and watches only one video at a time from beginning to the end.

Helper h_j has storage and upload bandwidth capacities equal to S_j and B_j, respectively. If the helper nodes adjacent to user u can cumulatively provide the streaming rate of the requested video by the user, the video will be downloaded from the helpers. Otherwise, the user will request the remaining portion of the video directly from the central server (Fig. 1). Our objective is to minimize the server's total upload rate.

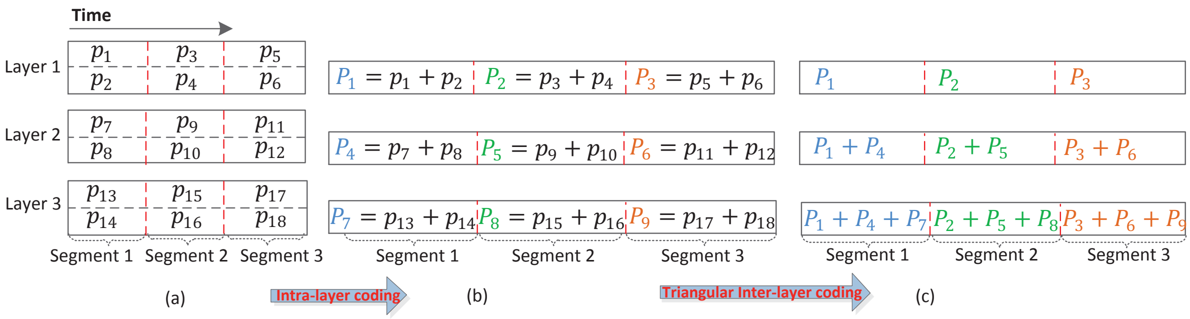

Time
Layer 1
Layer 2
Layer 3
p_1 p_2 p_3 p_4 p_5 p_6
p_7 p_8 p_9 p_{10} p_{11} p_{12}
p_{13} p_{14} p_{15} p_{16} p_{17} p_{18}
Segment 1 Segment 2 Segment 3
(a)
Intra-layer coding
$P_1 = p_1 + p_2$ $P_2 = p_3 + p_4$ $P_3 = p_5 + p_6$
$P_4 = p_7 + p_8$ $P_5 = p_9 + p_{10}$ $P_6 = p_{11} + p_{12}$
$P_7 = p_{13} + p_{14}$ $P_8 = p_{15} + p_{16}$ $P_9 = p_{17} + p_{18}$
Segment 1 Segment 2 Segment 3
(b)
Triangular Inter-layer coding
P_1 P_2 P_3
$P_1 + P_4$ $P_2 + P_5$ $P_3 + P_6$
$P_1 + P_4 + P_7$ $P_2 + P_5 + P_8$ $P_3 + P_6 + P_9$
Segment 1 Segment 2 Segment 3
(c)

Fig. 4. (a) Segmentation of a multi-layer video. (b) Intra-layer NC. (c) Joint inter- and intra-layer coding.

TABLE I
THE SET OF SYMBOLS USED IN THIS PAPER.

Notation	Definition
u_i, U	The i-th user, The set of users
h_j, H	The j-th helper, The set of helper nodes
m_k, M	The k-th video, The set of videos
B_j/S_j	The bandwidth/capacity of helper h_j
r_i, v_i	The rate and the size of video m_k, respectively
$N(u_i)/N(h_j)$	The set of adjacent helpers/users to u_i/h_j
x_{ji}^{kl}	Upload rate from h_j to u_i over layer l of video m_k
f_j^{kl}	The fraction of layer l of video m_k stored on helper h_j
e_k	The number of layers of video m_k
q_i	The requested video by user u_i
c_i	The number of requested layers by user u_i
x_j^k	Upload rate of helper h_j over video m_k (in live streaming)
d_{ji}^k	The download rate of user u_i from helper node h_j over video m_k (in live streaming)

In other words, we want to maximize the total amount of provided videos from the helper nodes to the users.

In order to provide the users with different levels of video qualities, each video m_k is divided into e_k layers with the same streaming rate and size equal to $\frac{r_k}{e_k}$ and $\frac{v_k}{e_k}$, respectively. Each user u_i can subscribe to its desired number of layers c_i. The l-th layer of a video is not useful unless all of the layers with a smaller index are available. Let the j-th helper's upload rate to user u_i over the l-th layer of video m_k be x_{ji}^{kl}. We represent the set of adjacent helpers to users u_i and adjacent users to helper h_j as $N(u_j)$ and $N(h_j)$, respectively. Table I summarizes the set of symbols used in this paper.

The optimal distribution of the videos on the helper nodes and the bandwidth allocation to the users is a challenging problem, even under a fixed network and demands assumption. We introduce a distributed algorithm to find the optimal solution for the stationary case. In the simulation results section, we show that our algorithm converges to the optimal solution, even in the case of dynamic networks.

III. VoD with Multi-Layer Videos

In general, a helper might not be able to store a full copy of a video because of storage limitations. Moreover, a helper node might provide more help to the central server by storing more videos in part rather than storing a small number of them in full [15]. Under this setting, to minimize the pressure on the central server, the following questions have to be addressed:

- *Content placement*: Which packets of which layers of each video should a helper node store?
- *Bandwidth allocation*: Which packets, and to which users, should each helper node serve its content?
- *Coding scheme*: Which coding scheme should be used?

Intra-layer NC helps to simplify the content placement problem on the helpers. As stated in the introduction, intra-layer NC also increases the efficiency of the content placement on the helper nodes. For this purpose, we divide each layer of a video into segments of n packets. Fig. 4(a) shows a video with 3 layers. In our intra-layer NC scheme, each coded packet of a segment is a random linear combination of the whole packets in that segment. In Fig. 4(b), the coefficients are not shown for simplicity. For instance, $p_1 + p_2$ means $a_1 p_1 + a_2 p_2$, where a_i is a random coefficient. When using intra-layer NC, all of the coded packets from the helper nodes will contribute the same amount of information, and a user will be able to view the segment if it downloads any n linearly independent coded packets from the helper nodes that have the segment stored.

In order to enable a helper to serve any users watching video m, regardless of their playback time, we uniformly store the packets from each segment of the video. Using this scheme, in order to store a fraction f of a layer of video m on helper h, we store $f \times n$ random linearly coded packets of each segment on the helper. Consider the video layer in Fig. 5(a), in which each segment contains 4 packets. Assume that we want to store half of the video layer on a helper. We store 2 random linearly coded packets for each segment as shown in Fig. 5(b). Note that the 2 coded packets of each segment are different, since they have different random coefficients. The coefficients are not shown for simplicity. Using this scheme, helper h can supply at the rate of $f \times r$ to the users that need video m, where r is the rate of the video. The use of intra-layer NC enables a flow-based model of the content, which changes our questions to finding 1) the rate at which coded packets of a video layer should be stored on a helper node, 2) the rates at which coded packets of a video layer should be uploaded to a helper's adjacent users, and 3) the optimal coding scheme.

A user might receive the packets of the current segment from the helpers with different delays, so the user is not able to decode the segment until it receives enough coded packets. In order to address this problem, which might result in the video lag problem, each user buffers the received coded packets and delays the playback so that the differences of the transmission delays does not result in playback lags. Computing the buffering time is beyond the scope of this work.

A. Intra-Layer Coding

Minimizing the server load is equivalent to maximizing the help provided by the helpers, which can be modeled as the

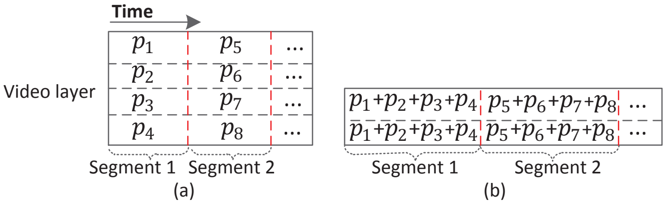


Fig. 5. (a) Segments of a video layer. (b) Storing half part of the video layer on a helper node.

following LP optimization problem:

$$\max \sum_{\substack{i,k:u_i\in U\\ m_k=q_i}} \sum_{\substack{j,l:h_j\in N(u_i)\\ l\le c_i}} x_{ji}^{kl} \tag{1}$$

$$s.t \quad x_{ji}^{kl} \le f_j^{kl} \times \frac{r_k}{e_k}, \quad \forall j,i,l : u_i \in N(h_j), l \le e_k \tag{2}$$

$$\sum_{\substack{i,k:u_i\in N(h_j)\\ m_k=q_i}} \sum_{l\le c_i} x_{ji}^{kl} \le B_j, \quad \forall j : h_j \in H \tag{3}$$

$$\sum_{k:m_k\in M} \sum_{l:l\le e_k} f_j^{kl} \times \frac{v_k}{e_k} \le S_j, \quad \forall j : h_j \in H \tag{4}$$

$$\sum_{j:h_j\in N(u_i)} x_{ji}^{kl} \le \frac{r_k}{e_k}, \quad \forall i,l : u_i \in U, l \le c_i \tag{5}$$

$$0 \le f_j^{kl} \le 1, \quad \forall j,k,l : h_j \in H, m_k \in M, l \le e_k \tag{6}$$

Objective function (1) is the summation of the helper's upload rates to the users over the subscribed layers of the requested videos. Function (1) is linear, so it is a concave function. We use set of Constraints (2) to limit each helper's upload rate at the available service rate of the videos. This upload rate differs for different layers of a video; thus, for each layer of a video, we have a separate constraint. Constraints (3) and (4) are feasibility constraints on bandwidth and storage, respectively. The total upload rate of a helper and the total stored data on it cannot exceed its bandwidth and capacity limit. Note that in VoD applications, even in the case that the adjacent users to a helper watch the same video, their playback times are different; so, the helper needs to allocate separate bandwidths to the users. Thus, this optimization problem works for both a wireless or wired network.

We assume that the rates of different layers of a video are the same. Therefore, the streaming rate of each layer of video m_k is equal to $\frac{r_k}{e_k}$. It is sufficient for user u to download each layer of its requested video at a rate equal to the streaming rate of the layer, and more than that value will not be useful. The set of Constraint (5) limits the aggregated download rate of the requested layers of video m to user u at the rate of the layer. The set of Constraints (6) are the feasibility constraints on the fraction of stored video layers on the helpers.

Assuming that each user is connected to all of the helpers, the number of variables x and f are equal to $|U| \times |H| \times e$ and $|H| \times |M| \times e$, where e is the maximum number of video layers. Moreover, the number of Constraints (2)-(6) are equal to $|U| \times |H| \times e$, $|H|$, $|H|$, $|U| \times e$, and $|H| \times |M| \times e$, respectively. Therefore, the solution of the optimization can be calculated in polynomial time [16]. The proposed optimization can be easily extended to the case of layers with different sizes and streaming rates, by substituting different sizes and streaming rates for each layer of the videos in the constraints.

$$\begin{cases} p_1 \\ p_2 \\ p_3 \end{cases} \qquad \begin{cases} p_1,\ p_2,\ p_3 \\ p_1+p_2,\ \ p_1+p_3,\ p_2+p_3 \\ p_1+p_2+p_3 \end{cases} \qquad \begin{cases} p_1 \\ p_1+p_2 \\ p_1+p_2+p_3 \end{cases}$$

(a) (b) (c)

Fig. 6. $p_1 + p_2$ means $a_1p_1 + a_2p_2$, where a_i is a random coefficient. (a) Original packets. (b) General form of random linear NC. (c) Triangular NC.

B. Joint Inter- and Intra-Layer Coding

In the general form of random linear NC, each packet can be coded with any other packets. Thus, in the case of n packets, there are $2^n - 1$ random linear NC possibilities. In contrast with the general form, in *triangular NC* [17], each coded packet is a random linear combination of the first i packets, $\forall i : 1 \le i \le n$. Therefore, there are just n possibilities for coding n original packets. Figs. 6(b) and (c) show the possible coded packets using the general form of NC and triangular coding, respectively. The coefficients are not shown in the figures.

As stated in the introduction, inter-layer NC helps to increase the provided help of the helper nodes. In order to benefit from joint inter- and intra-layer NC, we first perform intra-layer NC (Fig. 4(b)). Then, we use the triangular NC scheme to code the intra-layer coded packets together. In our scheme, the coded packets of each segment of a video's l-th layer are a random linear combination of that segment in layers 1 to l. Fig. 4(c) depicts the joint inter- and intra-layer coded packets using the triangular scheme. In this figure, the packets of layer 1 are the same as in the intra-layer approach, but the packets of layer 2 are a linear combination of layers 1 and 2. Also, the packets of layer 3 are a random linear combination of all the 3 layers. The random coefficients are not shown for simplicity. For example, $P_1 + P_4$ means $a_1P_1 + a_2P_4$.

We prefer using triangular NC over the general form for two reasons. First, it limits the coding space of the coding problem such that we can formulate the joint inter- and intra-layer NC as a convex optimization problem. Moreover, in our setting, the gain of the triangular NC is not less than the general form of NC, which is illustrated by the following theorem:

Theorem 1: Under the proposed setting, the gain of the triangular NC is at least equal to that of the general form of inter-layer NC.

Proof: We show that changing the general linear codes (non-triangular codes) to triangular codes does not decrease the total gain. Assume that P is a non-triangular code and I is the set of the indices of the layers in P. Also, assume that the largest index in I is d. Clearly, the coded layer P is not useful for the users that requested fewer layers than d, so changing P does not have a negative effect on these users. On the other hand, the users that requested at least d layers need to retrieve all of the layers from 1 to d. As a result, changing the coded layer P to a triangular code P' does not have any negative impact on these users. ■

Assume that user u_i has subscribed to c_i layers, each of which contains n packets. We represent the received coded packets of the l-th coded layer as Z_l. In [17], it is shown that under the triangular coding scheme, a user can decode all of the c_i layers if $\sum_{j=c_i-l+1}^{c_i} |Z_j| \geq l \times n,\ \forall l \in [1, c_i]$. This means that the total number of received coded packets should be at least equal to $c_i \times n$. Also, the total number of received coded packets from layers 2 to c_i needs to be equal to or more than $(c_i - 1)n$. In general, the number of received coded packets from layers l to c_i should not be less than $(c_i - l + 1)n$, which gives us an insight into the following lemma:

Lemma 1: Providing more than $l \times n$ coded packets from the first l coded layers is not useful to user u.

Proof: Obviously, receiving more than n coded packets from layer 1 is not useful to user u, since n coded packets are enough to decode layer 1. Coded layer 2 contains coded packets over the first two original layers. As a result, the user needs $2n$ coded packets from the first two layers. Since $|Z_1| \leq n$, the rank of $Z_1 \cup Z_2$ is equal to $2n$. With the same reasoning, receiving more than $l \times n$ coded packets from the first l coded layers is not useful. ■

By using Lemma 1, we can formulate the problem using joint inter- and intra-layer NC as follows:

$$\max \sum_{\substack{i,k:u_i \in U\\ m_k = q_i}} \sum_{\substack{j,l:h_j \in N(u_i)\\ l \leq c_i}} x_{ji}^{kl}$$

$$s.t \quad x_{ji}^{kl} \leq f_j^{kl} \times \frac{r_k}{e_k}, \quad \forall j,i,l : u_i \in N(h_j), l \leq e_k$$

$$\sum_{\substack{i,k:u_i \in N(h_j)\\ m_k = q_i}} \sum_{l \leq c_i} x_{ji}^{kl} \leq B_j, \quad \forall j : h_j \in H \tag{7}$$

$$\sum_{k:m_k \in M} \sum_{l:l \leq e_k} f_j^{kl} \times \frac{v_k}{e_k} \leq S_j, \quad \forall j : h_j \in H \tag{8}$$

$$\sum_{l=1}^{l'} \sum_{j:h_j \in N(u_i)} x_{ji}^{kl} \leq \frac{r_k}{e_k} \times l', \quad \forall i, l' : 1 \leq l' \leq c_i \tag{9}$$

Much like the proposed inter-layer optimization, Constraints (6), (7), and (8) are feasibility constraints on the fraction of each video on each helper node, bandwidth, and storage. The set of Constraints (9) implies that the total upload rates of the first l layers of the user-requested video should not be more than l times the streaming rate of each layer. This set of constraints ensures that the helpers will not provide coded packets to the users that are not useful for decoding.

IV. Wireless Live Streaming Applications

In this section, we show how the proposed solution for VoD can be extended for wireless live streaming (LS) applications. By LS we are referring to applications where some videos are broadcast to the users, such as TV station channels or surveillance systems. In VoD, the users can play the videos asynchronously. However, in LS, the playback time of the users that watch the same video are synchronous. Thus, the main difference between LS and VoD is that in LS, the helpers do not need to allocate separate bandwidths to their adjacent users that watch the same video.

In the case of VoD, the summation of the allocated bandwidth from each helper to its adjacent users should be less than or equal to its bandwidth. However, in LS, the summation of the allocated bandwidth from each helper for all of the videos should be less than or equal to its bandwidth. The reason for this is that more than one neighboring user might request the same video, and all of the users use the same broadcast packets. In order to formulate the case of LS, we represent the allocated bandwidth for the video m_k over helper h_j as x_j^k. The summation of these variables for each helper node should be less than or equal to the helper's bandwidth. Also, the download rate of user u_i over video m_k from the helper node h_j, which is represented as d_{ji}^k, should be less than or equal to x_j^k. The problem of LS in the case of single layer videos can be formulated as follows:

$$\max \sum_{\substack{i,k:u_i \in U\\ m_k = q_i}} \sum_{j:h_j \in N(u_i)} d_{ji}^k \tag{10}$$

$$s.t \quad x_j^k \leq f_j^k \times r_k, \quad \forall j,k : m_k \in M \tag{11}$$

$$\sum_{k:m_k \in M} x_j^k \leq B_j, \quad \forall j : h_j \in H \tag{12}$$

$$\sum_{k:m_k \in M} f_j^k \times v_k \leq S_j, \quad \forall j : h_j \in H \tag{13}$$

$$d_{ji}^k \leq x_j^k, \quad \forall i,j,k : h_j \in N(u_i), m_k = q_i \tag{14}$$

$$\sum_{k:m_k = q_i} \sum_{j:h_j \in N(u_i)} d_{ji}^k \leq r_k, \quad \forall i : u_i \in U$$

We also have Constraint (6). Objective function (10) is the summation of the download rates of users. The set of Constraints (11) ensures that the upload rate of a video by a helper node cannot exceed the available service rate of the video. Constraints (12), (13), and (6) are feasibility constraints on bandwidth and storage. We limit the download rate of a user from a helper node to the upload rate of its requested movie using the set of Constraints (14). We refer to our method as wireless live streaming (WLS).

V. Distributed Solution

In this section, we solve the proposed convex optimization problem for the case of multi-layer VoD streaming using intra-layer NC in a distributed way. The same approach can be used to find a distributed solution for the other settings. The idea is to solve the lagrangian dual of the problem using the gradient method. In this way, the helpers start from empty storage, and gradually update their storage and bandwidth allocation, based on the exchanged lagrange variables between them and their users. The objective function (1) is not strictly concave, due to the presence of a linear summation. Consequently, a direct application of standard gradient iterative method might lead to multiple solutions. In this case, the output of an iterative method may oscillate between multiple solutions. In order to overcome the problem due to the lack of strict concavity, we can apply the Proximal method described in [18], page 233. The idea behind the Proximal method is to add quadratic terms to the objective function and make it strictly concave. A detailed description of the Proximal method is in [18], [19]. To apply the Proximal method, we introduce auxiliary variables y_{ji}^{kl}. By using the Proximal method, the optimization becomes:

$$\max_{\vec{x},\vec{y}} \sum_{\substack{i,k:u_i\in U\\ m_k=q_i}} \sum_{\substack{j,l:h_j\in N(u_i)\\ l\leq c_i}} \left(x_{ji}^{kl} - (x_{ji}^{kl} - y_{ji}^{kl})^2\right) \tag{15}$$

subject to Constraints (2), (3), (4), (5), and (6).

The optimal solution of (15) is also the solution of (1). Let $\vec{x^*}$ and $\vec{f^*}$ be the optimal solution of (1) then, $\vec{x} = \vec{x^*}$, $\vec{f} = \vec{f^*}$, and $\vec{y} = \vec{x}$ is the maximizer of (15). The standard proximal method iteratively works as follows:

1) Fix $\vec{y}(t)$ and maximize (15) with respect to variables $\vec{x}(t)$ and $\vec{f}(t)$.
2) Set $\vec{y}(t+1) = \vec{x}(t)$, increment t and go back to step 1 .

Since the Slater condition holds (see reference [20]), there is no duality gap between the primal and the dual problems. Therefore, we can use the dual approach to solve the problem. Let λ_1^{jil}, λ_2^j, λ_3^j, and λ_4^{il} be the Lagrange variables for Constraints (2), (3), (4), and (5), respectively. Here, i, j, and l are corespondent to the indices in the set of Constraints (2) to (5). The Lagrange function of (15) is:

$$\begin{aligned}
L(\vec{x},\vec{f},\vec{y},\vec{\lambda}) = &\sum_{\substack{i,k:u_i\in U\\ m_k=q_i}} \sum_{\substack{j,l:h_j\in N(u_i)\\ l\leq c_i}} \left(x_{ji}^{kl} - (x_{ji}^{kl} - y_{ji}^{hl})^2\right)\\
&- \sum_{\substack{j,i:h_j\in H\\ u_i\in N(h_j)}} \sum_{\substack{k,l:m_k=q_i\\ l\leq c_i}} \lambda_1^{jil}(x_{ji}^{kl} - f_j^{kl} \times \frac{r_k}{e_k})\\
&- \sum_{j:h_j\in H} \lambda_2^j \sum_{\substack{i,k,l:u_i\in N(h_j)\\ m_k=q_i\\ l\leq c_i}} (x_{ji}^{kl} - B_j)\\
&- \sum_{j:h_j\in H} \lambda_3^j \sum_{\substack{k,l:m_k\in M\\ l\leq e_k}} (f_j^{kl} \times \frac{v_k}{e_k} - S_j)\\
&- \sum_{\substack{i,k,l:u_i\in U\\ m_k=q_i\\ l\leq c_i}} \lambda_4^{il} \sum_{j:h_j\in N(u_i)} (x_{ji}^{kl} - \frac{r_k}{e_k})
\end{aligned}$$

By rearranging the terms, we have:

$$\begin{aligned}
&L(\vec{x},\vec{f},\vec{y},\vec{\lambda}) =\\
&\sum_{\substack{i,k:u_i\in U\\ m_k=q_i}} \sum_{\substack{j,l:h_j\in N(u_i)\\ l\leq c_i}} \left[(1-\lambda_1^{jil}-\lambda_2^j-\lambda_4^{il})x_{ji}^{kl} - (x_{ji}^{kl} - y_{ji}^{kl})^2\right]\\
&+ \sum_{\substack{j,i:h_j\in H\\ u_i\in N(h_j)}} \sum_{\substack{k,l:m_k=q_i\\ l\leq c_i}} \lambda_1^{jil} f_j^{kl} \frac{r_k}{e_k} - \sum_{j:h_j\in H} \sum_{\substack{k,l:m_k\in M\\ l<e_k}} \lambda_3^j f_j^{kl} \times \frac{v_k}{e_k}
\end{aligned}$$

By a simple change of variables, the Lagrange function is separable in $\vec{x}$ and $\vec{f}$, and we can rewrite it as:

$$\begin{aligned}
&L(\vec{x},\vec{f},\vec{y},\vec{\lambda}) =\\
&\sum_{\substack{i,k:u_i\in U\\ m_k=q_i}} \sum_{\substack{j,l:h_j\in N(u_i)\\ l\leq c_i}} \left[(1-\lambda_1^{jil}-\lambda_2^j-\lambda_4^{il})x_{ji}^{kl} - (x_{ji}^{kl} - y_{ji}^{kl})^2\right]
\end{aligned} \tag{16}$$

$$+ \sum_{\substack{j,k:h_j\in H\\ m_k\in M}} (\sum_{\substack{i,l:u_i\in N(h_j)\\ l\leq c_i}} \lambda_1^{jil} \times \frac{r_k}{e_k} - \sum_{l:l<e_k} \lambda_3^j \times \frac{v_k}{e_k}) f_j^{kl} \tag{17}$$

Algorithm 1 Calculation of $\vec{f}$ (for helper node h_j)

```
$rem = S_j$, calculate $\gamma_j^{kl} \quad \forall k,l : m_k \in M, l \leq e_k$
for each $f_j^{kl}$ in descending order of $\gamma_j^{kl}$ do
  if $\gamma_j^{kl} > 0$ and $rem > 0$ then
    if $rem > \frac{v_k}{e_k}$ then
      set $f_j^{kl} = 1$, $rem = rem - \frac{v_k}{e_k}$
    else
      set $f_j^{kl} = \frac{rem}{v_k/e_k}$, $rem = 0$
  else
    $f_j^{kl} = 0$, $rem = 0$
```

The objective function of the dual problem is:

$$D(\vec{y},\vec{\lambda}) = \max_{\substack{\vec{x}\geq 0\\ \vec{y}\geq 0}} L(\vec{x},\vec{f},\vec{y},\vec{\lambda})$$

The dual problem itself is $\min_{\lambda\geq 0} D(\vec{y},\vec{\lambda})$. The dual optimization problem can be solved using the gradient method. The updates of the Lagrange variables are listed as follows:

$$\lambda_1^{jil}(t+1) = \left[\lambda_1^{jil}(t) + \alpha(x_{ji}^{kl}(t) - f_j^{kl}(t) \times \frac{r_k}{e_k})\right]^+,$$
$$\forall j,i,k,l : h_j \in H, u_i \in N(h_j), m_k = q_i, l \leq e_k$$

$$\lambda_2^j(t+1) = \left[\lambda_2^j(t) + \alpha \sum_{\substack{i,k:u_i\in N(h_j)\\ m_k=q_i}} \sum_{l\leq c_i} (x_{ji}^{kl}(t) - B_j)\right]^+,$$
$$\forall j : h_j \in H$$

$$\lambda_3^j(t+1) = \left[\lambda_3^j(t) + \alpha\Big(\sum_{k:m_k\in M} \sum_{l:l\leq e_k} (f_j^{kl}(t) \times \frac{v_k}{e_k} - S_j)\Big)\right]^+,$$
$$\forall j : h_j \in H$$

$$\lambda_4^{il}(t+1) = \left[\lambda_4^{il}(t) + \alpha\Big(\sum_{j:h_j\in N(u_i)} (x_{ji}^{kl}(t) - \frac{r_k}{e_k})\Big)\right]^+,$$
$$\forall i,k,l : u_i \in U, m_k = q_i, l \leq c_i$$

where $[.]^+$ denotes the projection on $[0,\infty)$. Also, by setting the first derivative of (16) with respect to $\vec{x}$ equal to zero, the optimal $\vec{x}$ can be calculated as follows:

$$x_{ji}^{kl}(t+1) = \frac{1 - \lambda_1^{jil}(t) - \lambda_2^j(t) - \lambda_4^{il}(t)}{2} + y_{ji}^{kl}(t)$$
$$\forall j,i,k,l : h_j \in H, u_i \in N(h_j), m_k = q_i, l \leq e_k$$

Algorithm 1 illustrates the computation of $\vec{f}$. Here, $\gamma_j^{kl} = \sum_{\substack{j,k:h_j\in H\\ m_k\in M}}(\sum_{\substack{i,l:u_i\in N(h_j)\\ l\leq c_i}} \lambda_1^{jil} \times \frac{r_k}{e_k} - \sum_{l:l<e_k} \lambda_3^j \times \frac{v_k}{e_k})$ is the multiplier of f_j^{kl} in Equation (17), and rem is the free space of helper node h_j. The idea here is that, in order to maximize Equation (17), we should give a greater value to the fraction of the videos with a greater γ value. On the other hand, the fraction of videos with a negative γ value should be equal to zero. Therefore, for each helper node, we sort the γ_j^{kl} in

Algorithm 2 Users' Protocol (for user u_i)

```
**Initialization** Send the request and the number of desired layers to the adjacent helper nodes. Set $\lambda_4^{il}(1,0) = 0$
**Iteration Phase** at the $\tau$-th iteration
**for** $t = 0, ..., T-1$ perform the following step sequentially
    send $\lambda_4^{il}(\tau, t+1) = [\lambda_4^{il}(\tau, t) + \alpha(\sum_{j:h_j \in N(u_i)}(x_{ji}^{kl}(\tau, t) - \frac{r_k}{e_k}))]^+ \quad \forall l : l \leq c_i$ to all adjacent helpers.
$\vec{\lambda_4}(\tau + 1, 0) = \vec{\lambda_4}(\tau, T)$
```

Algorithm 3 Helpers' Protocol (for helper node h_j)

```
**Initialization** set $x_{ji}^{kl}(1,0) = 0$, $f_i^{kl}(1,0) = 0$, $\lambda_1^{jil}(1,0) = 0$, $\lambda_2^j(1,0) = 0$, $\lambda_3^j(1,0) = 0$, $y_{ji}^{kl}(1,0) = 0$
**Iteration Phase** at the $\tau$-th iteration
**for** $t = 0, ..., T-1$ perform the following steps sequentially
    $\lambda_1^{jil}(\tau, t+1) = [\lambda_1^{jil}(\tau, t) + \alpha(x_{ji}^{kl}(\tau, t) - f_j^{kl}(\tau, t)\frac{r_k}{e_k})]^+$
    $\lambda_2^j(\tau, t + 1) = [\lambda_2^j(\tau, t) + \alpha\sum_{i:u_i \in N(h_j)} \sum_{l \leq c_i}(x_{ji}^{kl}(\tau, t) - B_j)]^+$
    $\lambda_3^j(\tau, t + 1) = [\lambda_3^{jil}(\tau, t) + \alpha(\sum_{k:m_k \in M} \sum_{l:l \leq e_k}(f_j^{kl}(\tau, t)\frac{v_k}{e_k} - S_j))]^+$
    $x_{ji}^{kl}(\tau, t+1) = \frac{1 - \lambda_1^{jil}(\tau,t) - \lambda_2^j(\tau,t) - \lambda_4^{il}(\tau,t)}{2} + y_{ji}^{kl}(\tau, t)$
    run algorithm 1 to calculate $\vec{f}(\tau, t+1)$
$\vec{y}(\tau+1, 0) = \vec{x}(\tau, T)$, $\vec{x}(\tau+1, 0) = \vec{x}(\tau, T)$, $\vec{\lambda_1}(\tau+1, 0) = \vec{\lambda_1}(\tau, T)$, $\vec{\lambda_2}(\tau+1, 0) = \vec{\lambda_2}(\tau, T)$, $\vec{\lambda_3}(\tau+1, 0) = \vec{\lambda_3}(\tau, T)$
```

descending order of their values, and we start to fill the helper nodes with videos that have a greater γ.

The convergence of our algorithm can be proven using a technique similar to [21]. We omit the proof for brevity, and in our simulation we empirically verify the convergence.

We can define two iterative levels for the distributed algorithm [19]. In the inner loop, we fix the auxiliary variables $\vec{y}$ and update $\vec{x}$, $\vec{f}$, and $\vec{\lambda}$, for T times. We run the outer loop τ times, in which we set $\vec{y}(\tau + 1, 0) = \vec{x}(\tau, T)$. The users' and helpers' policies are shown in Algorithms 2 and 3, respectively.

VI. Simulation Results

We compare our proposed methods with the proposed no-layer VoD method in [3]. The authors formulate the VoD problem as LP, and propose a distributed solution for it. We refer to both their LP an distributed solution as DIST method. We also study the convergence of the proposed distributed method under the static and dynamic cases.

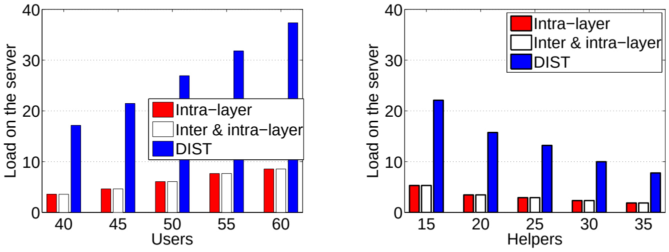
We assume that the popularity of the videos and the number of subscribed layers by each user are uniformly distributed. The range of a video's rate, size, storage capacity, bandwidth capacity, and number of adjacent helpers to each user are randomly chosen in the ranges shown in Table II.

A. Performance

We evaluate the methods on 100 random topologies, and use the average output of the simulation for plots of this section.

TABLE II
The ranges of the parameters in the simulations.

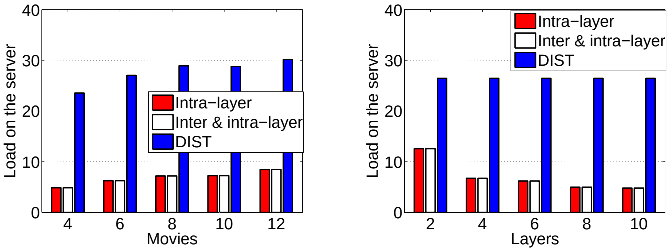
Video's rate	Video's size	Bandwidth capacity	Storage capacity	Num. of adjacent helpers to a user
[1,2] kbps	[0.5,2] MB	[2,4] kbps	[0.5,2] MB	[1,3]



(a) Number of helpers: 20

(b) Number of users: 40

Fig. 7. Server's load, VoD. Number of videos: 5; number of layers: 5.



(a) Number of layers: 5

(b) Number of videos: 5

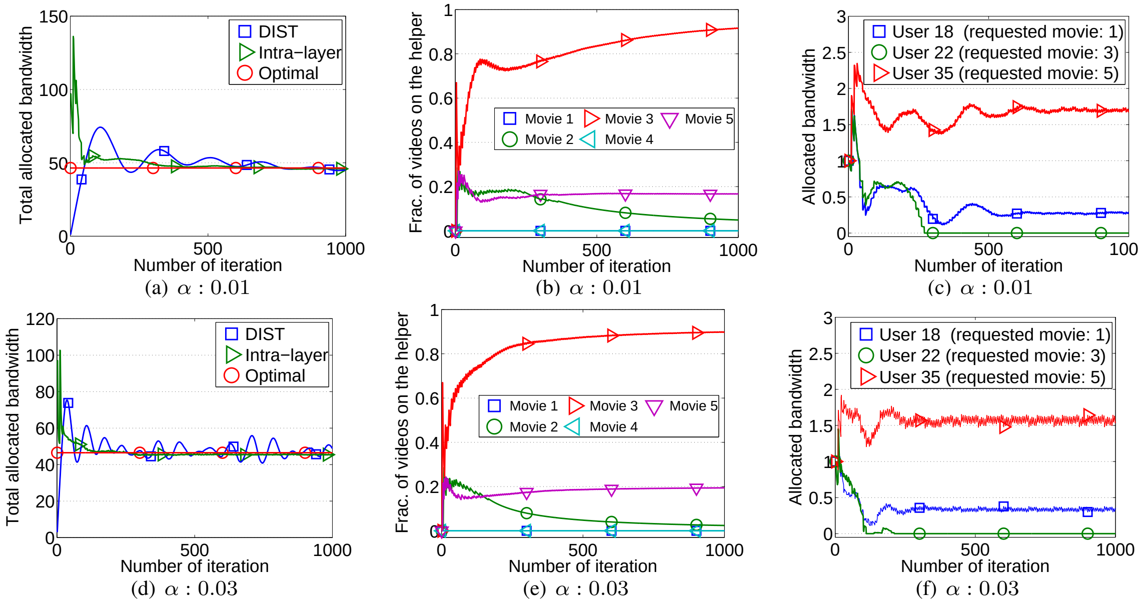
Fig. 8. Server's load, VoD. Number of users: 50; number of helpers: 20.

In Fig. 7(a), we compare the load on the central server. Each video contains 5 layers, and the number of requested layers by each user is randomly chosen in the range of $[1, 5]$. The other parameters are shown in Table II. The figure shows that the result of the joint inter- and intra-layer coding is almost the same as the intra-layer coding. The server's load in our methods is up to 75% less than that of the DIST approach.

In our next experiment, we study the effect that the number of helper nodes has on the server's load. It is clear that more helper nodes can provide more portions of the videos, due to more available capacity and bandwidth resources. As a result, the server's load in all of the methods decreases as we increase the number of helper nodes, as illustrated in Fig. 7(b).

Figs. 8(a) and (b) depict the effect of the number of videos and layers of the server's load. The simulations parameters are chosen randomly in the ranges shown in Table II. The server's load of the methods increases as we increase the number of videos. This is because, as we increase the number of choices, the number of common requests decreases. As a result, the helper nodes need to store more videos, which is not feasible due to the storage limitations. More layers give the users the choice to select videos with a lower quality, which decreases the load on the server as shown in Fig. 8(b). In this figure, the server's load is almost fixed in the DIST method, since DIST is a no-layer approach.

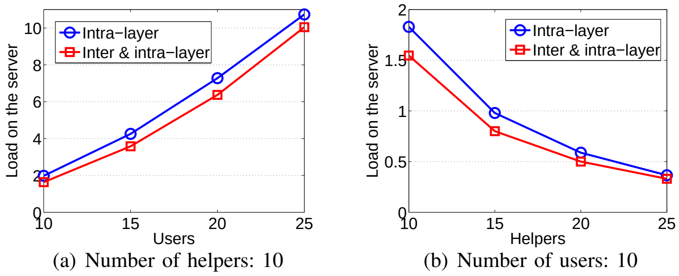
As we stated in the introduction, there are cases where the inter-layer NC reduces the server's load. However, Figs. 7 and 8 show that the server's load using joint NC and just intra-layer coding are very close. In order to study the benefit of inter-layer coding, we repeat the first experiment with a single video to eliminate competition between the users with different video



(a) $\alpha : 0.01$ (b) $\alpha : 0.01$ (c) $\alpha : 0.01$

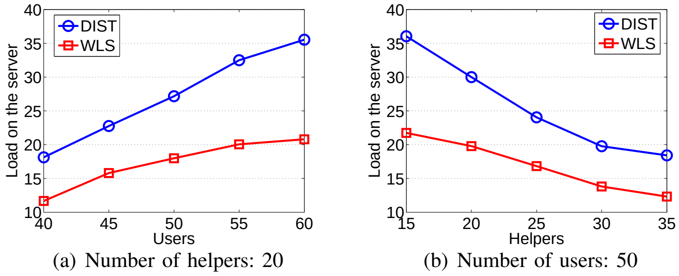
(d) $\alpha : 0.03$ (e) $\alpha : 0.03$ (f) $\alpha : 0.03$

Fig. 11. VoD. Convergence of the proposed distributed method to the optimal solution in a static network case. The numbers of users, helpers, and videos are equal to 50, 20, and 5, respectively. (a) and (d): Total allocated bandwidth to the users. (b) and (e): The fraction of each video on helper h_5. (c) and (f): The allocated bandwidth from helper h_5 to its adjacent users.



(a) Number of helpers: 10 (b) Number of users: 10

Fig. 9. The advantage of joint inter- & intra-layer coding over intra-layer coding. VoD. Number of layers: 4.



(a) Number of helpers: 20 (b) Number of users: 50

Fig. 10. Live streaming. Number of videos: 5; number of layers: 1.

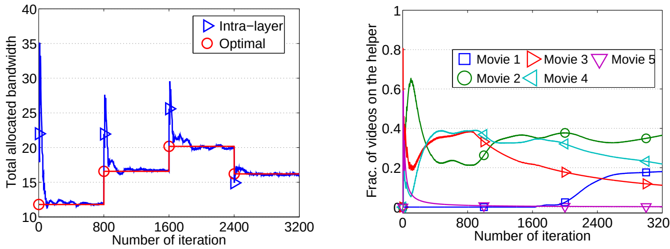
requests. The helpers' bandwidths are in the range of $[5, 10]$, and the video size, video rate, and the storage capacities are set to 4, 4, and 1. Also, the degree of each user is in the range of $[1, 4]$, and we set the number of requested layers of each user to its degree. Fig. 9(a) shows that the server's load using joint coding is up to 17% less than that of the intra-layer coding method. Fig. 9(b) shows that, as we increase the number of helpers, the difference between the methods decreases, which is due to the availability of a high percentage of the video through the helpers in both methods. Based on our observation, we can find that when the users compete in receiving different videos and the bandwidth is the bottleneck, inter-layer NC cannot increase the available content to the users.

Figs. 10(a) and (b) show the comparison between the server's load in the DIST and WLS (wireless live streaming) methods. The experiment parameters are chosen randomly in the ranges shown in Table II. In the case of LS, the playback time of the users that watch the same video are synchronous. Thus, in the WLS method, the helpers do not assign a separate bandwidth to the users that watch the same video, which results in providing more portions of the videos through the helper nodes. As a result, the server load in the WLS method is less than that of the DIST method. In Fig. 10(a), the slope of DIST is more than that of the WLS, which means that the helper nodes do not have enough free bandwidth to support more users. On the other hand, in Fig. 10(b) WLS has less slope than the DIST method since, even in the case of 15 helper nodes, the users receive a large portion of their requests.

B. *Convergence*

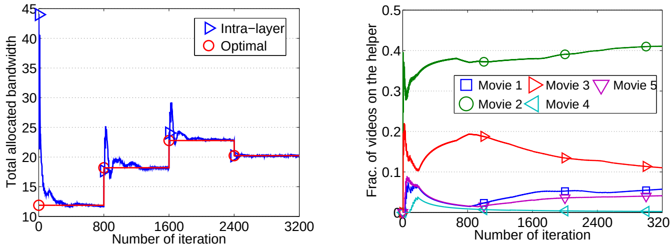
In this section, we study the convergence of our distributed solution under both the static and dynamic cases.

1) Static System: We evaluate the convergence of the proposed distributed storage and bandwidth allocation algorithm in Fig. 11. In this figure, the numbers of users, helpers, and videos are equal to 50, 20, and 5, respectively. In order to have a fair comparison, we set the number of video layers to 1. The optimal solution is computed off-line for comparison. It is clear in Fig. 11(a) that the proposed distributed solution converges to the optimal solution very fast; however, the convergence speed of the DIST approach is less than our approach. Moreover, the DIST method oscillates around the optimal solution. Fig. 11(b) depicts the convergence of a particular helper's (helper h_5) storage allocation. The allocated storage for videos 2 and 4 goes to zero, since these videos are not requested by the adjacent users of this helper node. The convergence of the allocated bandwidth from helper h_5 to its



(a) (b)

Fig. 12. Convergence of the proposed distributed method to the optimal solution in the case of dynamic users. Numbers of helpers: 10; Number of videos: 5. (a) Total allocated bandwidth. (b) Frac. of videos on helper h_8.



(a) (b)

Fig. 13. Convergence of the proposed distributed method to the optimal solution in the case of dynamic helpers. Numbers of users: 20; Numbers of videos: 5. (a) Total allocated bandwidth. (b) Frac. of videos on helper h_3.

adjacent users is shown in Fig. 11(c).

We repeat the previous experiment by increasing the step size α from 0.01 to 0.03. The results are shown in Figs. 11(d), (e), and (f). By comparing Figs. 11(a) and (d), it can be inferred that our distributed method converges faster to the optimal solution as we increase the step size. Moreover, even with a greater α, our method does not oscillate. On the other hand, the DIST method's oscillation increases rapidly as we increase the step size. Figs. 11(e) and (f) illustrate the bandwidth and storage allocation of helper h_5, respectively.

2) Dynamic System: The objective of this section is to show that our distributed approach automatically adapts to the system dynamics. As a result, the users and the helper nodes only need to run the distributed algorithm, regardless of the changes in the system.

We study the effect of changing the number of users to the system in Fig. 12. For this purpose, we add 5 users at each of iterations 800 and 1600, and we randomly connect them to [1,3] helper nodes. We also remove 5 users at iteration 2400. The initial number of users is 10, and there are 10 helper nodes in the system. We set the number of videos to 5. The optimal solution is computed off-line for comparison. Fig. 12(a) shows that the total allocated bandwidth of the optimal solution changes as we add or remove users, and the distributed solution converges to the optimal result. We depict the fraction of stored videos on a helper h_8 in Fig. 12(b).

We repeat the previous simulation for the case of dynamic helper nodes. We set the numbers of users, helpers, and videos to 20, 6, and 5, respectively. We add 3 new helpers at iterations 800 and 1600, and remove 3 helpers at iteration 2400. Figs. 13(a) and (b) show that the proposed distributed method adapts to the changes in the dynamic case.

VII. Conclusion

We study the problem of utilizing helper nodes to minimize the load on the central video servers. For this purpose, we formulate the problem as a linear programming optimization problem. This is done by using joint inter- and intra-layer NC, and through an empirical study, we found the cases that joint coding reduces the server's load. We also solve the proposed optimization in a distributed way. We evaluate the convergence and the gain of our distributed approach by comprehensive simulations. Our future work is to study the overhead of introducing the helper nodes and unreliability of the links.

References

[1] A. Finamore, M. Mellia, M. Munafò, R. Torres, and S. Rao, "Youtube everywhere: impact of device and infrastructure synergies on user experience," in *ACM IMC*, 2011, pp. 345–360.

[2] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *ACM SIGCOMM*, 2010, pp. 75–86.

[3] H. Hao, M. Chen, A. Parekh, and K. Ramchandran, "A distributed multichannel demand-adaptive P2P VoD system with optimized caching and neighbor-selection," in *SPIE*, 2011.

[4] J. Wang, C. Yeo, V. Prabhakaran, and K. Ramchandran, "On the role of helpers in peer-to-peer file download systems: Design, analysis and simulation," in *IPTPS*, 2007.

[5] J. Wang and K. Ramchandran, "Enhancing peer-to-peer live multicast quality using helpers," in *IEEE ICIP*, 2008, pp. 2300–2303.

[6] H. Zhang, J. Wang, M. Chen, and K. Ramchandran, "Scaling peer-to-peer video-on-demand systems using helpers," in *IEEE ICIP*, 2009, pp. 3053–3056.

[7] Y. He and L. Guan, "Improving the streaming capacity in P2P VoD systems with helpers," in *IEEE ICME*, 2009, pp. 790–793.

[8] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *ACM CCR*, 1996, pp. 117–130.

[9] M. Kim, D. Lucani, X. Shi, F. Zhao, and M. Médard, "Network coding for multi-resolution multicast," in *IEEE INFOCOM*, 2010, pp. 1–9.

[10] M. Effros, "Universal multiresolution source codes," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2113–2129, 2001.

[11] M. Shao, S. Dumitrescu, and X. Wu, "Layered multicast with inter-layer network coding for multimedia streaming," *IEEE Transactions on Multimedia*, vol. 13, no. 99, pp. 353–365, 2011.

[12] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[13] P. Ostovari, J. Wu, and A. Khreishah, "Network coding techniques for wireless and sensor networks," in *The Art of Wireless Sensor Networks*, H. M. Ammari, Ed. Springer, 2013.

[14] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[15] S. Pawar, S. Rouayheb, H. Zhang, K. Lee, and K. Ramchandran, "Codes for a distributed caching based video-on-demand system," in *ACSSC*, 2011.

[16] Y. He and L. Guan, "A new polynomial-time algorithm for linear programming," in *ACM STOC*, 1984, pp. 302–311.

[17] D. Koutsonikolas, Y. Hu, C. Wang, M. Comer, and A. Mohamed, "Efficient online wifi delivery of layered-coding media using inter-layer network coding," in *IEEE ICDCS*, 2011, pp. 237–247.

[18] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ (USA); Prentice Hall Inc., 1989.

[19] X. Lin and N. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 5, no. 51, pp. 766–781, 2006.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ Press, 2004.

[21] A. Khreishah, C.-C. Wang, and N. B. Shroff, "Optimization based rate control for communication networks with inter-session network coding," in *IEEE INFOCOM*, 2008, pp. 81–85.