The Benefits of Cooperation Between the Cloud and Private Data Centers for Multi-rate Video Streaming

Pouya Ostovari*, Jie Wu*, and Abdallah Khreishah†
*Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
†Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102
Email: {ostovari, jiewu}@temple.edu, abdallah.khreishah@njit.edu

***Abstract*—Video streaming is one of the applications with the highest traffic on the Internet. This high traffic leads to a lot of workload on the video servers (data centers), and increases the energy consumption of the servers. Reducing energy consumption becomes more important in the case that the data centers use renewable energy. The cost of these servers changes over time, based on the availability of energy resources such as solar and wind power. As a result, in order to reduce the cost of the servers, we need to use a mechanism to reduce the load on the servers, especially during times of energy cost increases. One way to tackle this problem is to lease some storage clouds that work as proxies during these periods of time, and provide the users with the popular videos through these storage clouds. However, finding the proper time during which the videos should be downloaded, and the efficient amount of storage, is not straightforward. Moreover, the popularity of the videos might change over different times of day, and for different geographic locations. In this paper, we find the optimal cloud lease that results in the minimum cost. For this purpose, we model the problem as a linear programming optimization, which can be solved in polynomial time. We also propose an optimal solution for the case of multi-resolution video coding, in which different users can request and watch the videos with different qualities.**

Keywords*—*Video streaming, renewable energy, storage cloud, multi-resolution video, optimization, video-on-demand, network coding.

I. Introduction

Due to recent advances in technology, which make the Internet more accessible, and the changes in life requirements, people use video streaming widely and more frequently. Recent studies show that video streaming is a dominant form of traffic on the Internet; for example, the YouTube and Netflix servers produce 20-30% of the web traffic on the Internet as shown in recent studies [1], [2]. This video traffic increases the workload on the video servers, and as a result, the energy consumption of the servers. Because of limited fossil fuels resources and global warming, green computing and using renewable energy resources received a lot of attention from the community [3]–[5] in recent years, as to reduce energy consumption and use of the limited energy resources.

Reducing energy consumption becomes more important in the case that the data centers use renewable energy. The cost of these servers changes over time, depending on the availability of renewable energy resources such as solar and wind power. Consequently, we need to use a mechanism to use the resources efficiently and reduce the workload on the servers, especially when the energy cost increases. One efficient way to tackle this problem is to use some helper nodes (proxies) that have been explored [6]–[10]. The helper nodes work in conjunction with the servers to serve the users with the videos [11], [12], which reduces the workload on the servers. The servers provide the portion of the video files that cannot be obtained from the helper nodes. These helper nodes can be considered as storage clouds (proxy clouds) that can be leased for a period of time. In the remainder of the paper, we use the terms proxy clouds and storage clouds interchangeably.

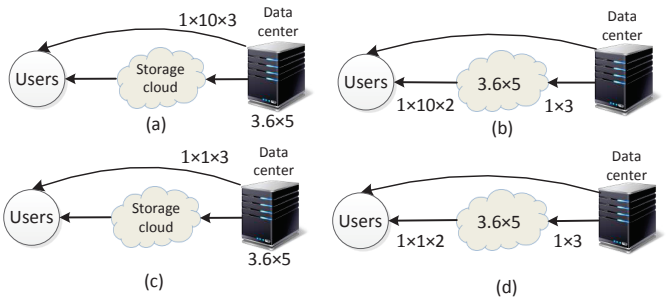


Fig. 1. Motivation example. (a) Direct streaming, expected requests: 10. (b) Indirect streaming, expected requests: 10. (c) Direct streaming, expected requests: 1. (d) Indirect streaming, expected requests: 1.

In this work, we consider a set of video servers that use renewable energy as their primary source of energy. However, the renewable energy sources may not be available, or may not be available in the right quantity; this may increase the energy cost of the servers, as they need to use other power sources. In order to minimize the video streaming cost, the company that owns the video servers can lease some storage clouds, such as Amazon, to help the servers in providing the popular videos to the users. Using this policy, the load on the video servers and their energy decreases. However, the company needs to pay a leasing cost for the storage clouds. Thus, the efficiency of leasing the storage clouds depends on the popularity of the videos and the energy cost of the video servers.

Consider the example in Figure 1(a). We have one server (data center) that use renewable energy as its power source, one storage cloud, and one region of the users. Assume that the available renewable energy is not sufficient for the data center at the current time, and it need to use another source of energy, which increases its cost. For simplicity, assume that we have just one video with a rate and size equal to 1 Mb/s and 3.6 Gb, respectively. Also, the expected number of users that request the video is 10. We assume that the storage energy cost on the server and the cloud are equal to 5 per Mb. Moreover, the bandwidth cost of the server to the users and the cloud is

equal to 3 per Mb. The bandwidth cost from the cloud to the users is equal to 2 per Mb.

We first compute the total cost in the case of direct download from the server, which is shown in Figure 1(a). The expected number of requests is equal to 10; thus, the total bandwidth cost from the server to the users is equal to $1 \times 10 \times 3 = 30$. The storage cost of the server is $3.6 \times 5 = 18$. As a result, the total cost of direct downloading becomes 48. In the case of indirect streaming shown in Figure 1(b), the bandwidth cost of the cloud and the server become $1 \times 10 \times 2 = 20$ and $1 \times 3 = 3$, respectively. Also, the storage cost of the cloud is $3.6 \times 5 = 18$. In this case, the total cost becomes 41. Therefore, indirect streaming is more efficient than direct streaming.

The popularity of the videos has an effect on the efficiency of direct or indirect streaming. Assume that the costs are the same as before, but the expected request for the video is 1. Figure 1(c) shows the costs in the case of direct streaming. The downloading cost of direct streaming becomes $1 \times 1 \times 3 = 3$. The storage cost remains the same, as in the previous case. Thus, the total cost is 21. The downloading costs from the server and cloud in the case of indirect streaming are equal to $1 \times 1 \times 2 = 2$ and $1 \times 3 = 3$, respectively. The storage costs are the same as in the case of 10 requests. Therefore, the total cost becomes equal to 23. Consequently, when the popularity of the video decreases, direct streaming is more efficient than the streaming using cloud. Figure 1(d) shows the costs of indirect streaming.

In this work, we answer the following questions: at a given time, how much storage and bandwidth should be leased from the storage clouds to minimize the total video-on-demand (VoD) streaming cost? Which videos, and which fraction of them, should be provided by the servers to the users directly, and which of them should be served indirectly by the storage clouds? We study video streaming using storage cloud proxies and servers that use renewable energy, and characterize the minimum cost VoD streaming using linear programming. We also propose a linear programming optimization for the case of multi-resolution VoD streaming.

The remainder of this paper is organized as follows: In Section II, we introduce the settings. We provide a background on network coding and propose our linear network coding scheme in Section III. We also motivate the optimal video streaming problem in Section III. We formulate the problem in the case of single and multi-resolution videos as a linear programming in Section IV. We evaluate our proposed methods through comprehensive simulations in Section V. We conclude the paper in Section VI.

II. System Setting and Problem Definition

We consider a set of data centers that provide a set of videos to the users, as shown in Figure 2. The servers are located in different geographic locations, e.g. on different continents. Also, the users are distributed over a set of regions. The servers use renewable energy as their primary power source. As a result, their storage and bandwidth cost changes over time. However, the future costs are not known exactly, and we just know the expected cost at different times of a day. In addition to the data centers, we can lease storage and bandwidth from storage clouds, that work as proxies, to keep the videos and provide them to the users. In this way, we can turn off some part of the data centers and reduce the cost once the renewable sources are not available, and the cost of the servers are high.

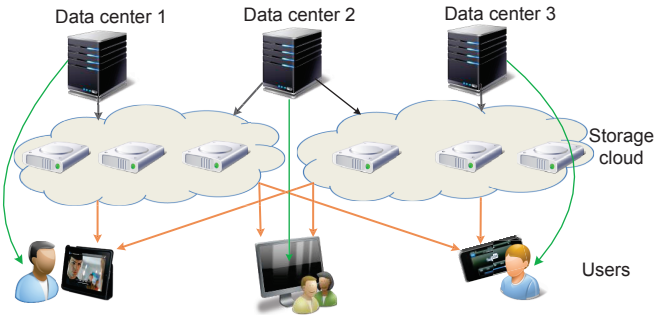


Fig. 2. The system architecture.

We represent the set of storage clouds, servers, users' regions, and videos as H, S, R, and M, respectively. We assume that the expected requests for each video at different times are known (calculating the expected requests is beyond the scope of this paper), and represent the expected requests for video m from region i at time t as $E_{im}(t)$. We also represent the downloading rate of region i over the video m from cloud j and sever k at time t as $x_{jim}(t)$ and $x_{kim}(t)$, respectively. We represent the downloading cost of region i from cloud j and server k at time t as $c_{1ji}(t)$ and $c_{1ki}(t)$, respectively. Moreover, $c_{1kj}(t)$ represents the download cost from server k to storage cloud j. The lease cost of cloud j and the storage cost of server k at time t are represented as $c_{2j}(t)$ and $c_{2k}(t)$, respectively. We show the size and rate of video m as v_m, and r_m, respectively. Table I summarizes the notations used in this paper.

The cost in our model consists of the downloading and storage costs. The storage cost itself is the summation of the leasing cost of the clouds and the cost of the storages on the servers. Our goal is to provide the requested videos to the users with the minimum total cost. For this purpose, we need to find the fraction of the requests that should be served directly by the servers, and the fraction of help that should be provided by the cloud storages.

If we assume that the popularity of the videos, the bandwidth cost, and storage cost of the cloud storages and the servers are fixed, then we just need to calculate the cost of direct transmission of each video from the severs to the different regions, and the cost of indirect service from the storage clouds. Then, the solution with the smaller cost should be selected. However, in our model, the bandwidth and storage cost changes over time, and we just have some estimations about the future possible costs. Moreover, the popularity of the videos are not fixed. As a result, it might be more efficient to download and store some of the videos that are not popular at the current moment, but which we expect to be requested more frequently in the next few hours, since we expect the renewable energy not to be available at that time.

TABLE I. THE SET OF SYMBOLS USED IN THIS PAPER.

Notation	Definition
S/H	The set of servers/storage clouds
R/M	The set of users' regions/videos
T	The total number of time slots
$x_{jim}(t)/x_{kim}(t)$	Upload rate of video m from the storage cloud j/server k to the users in region i-th at time t
$y_{kjm}(t)$	The fraction of video m downloaded from server k to storage cloud j at time t
$y_{kjml}(t)$	The fraction of layer l of video m downloaded from server k to storage cloud j at time t
$f_{jm}(t)$	The fraction of video m stored on cloud j at time t
$f_{jml}(t)$	The fraction of the layer l-th layer of video m stored on storage cloud j at time t
$c_{1ki}(t)$	The cost of download by users in region i from server k at time t
$c_{1kj}(t)$	The cost of download from server k to cloud j at time t
$c_{1ji}(t)$	The cost of download by users in region i from storage cloud j
$c_{2j}(t)/c_{2k}(t)$	The storage cost of cloud j/server k at time t
$w_k(t)/w_j(t)$	The total download cost from server k/storage cloud j at time t
$z_k(t)/z_j(t)$	The total storage cost of server k/cloud j at time t
E_{im}/E_{iml}	The expected number of users from region i that request video m/l-th layer of video m
L	The number of layers of the video

III. NETWORK CODING BACKGROUND AND VIDEO CODING SCHEME

In this section, we first provide a short background on network coding. We then propose the network coding scheme that we use to code the video packets.

A. Network Coding Background

In order to use the resources optimally, we need a mechanism to distribute the packets of the videos on the storage clouds, since depending on the lease cost and the popularity of the videos it might not be efficient enough to store the videos in full. *Network coding* [13], [14] helps simplify the content distribution problem, and to solve it in an efficient way. Network coding [15], [16] is first introduced in [13] for wired networks, as it is shown to solve the bottleneck problem in the single multicast problem. A useful algebraic representation of the linear network coding problem is provided in [17]. The authors in [18] show that selecting the coefficients of the coded packets randomly achieves the capacity asymptotically, with respect to the finite field size.

The coded packets in random linear network coding are random linear combinations of the original packets over a finite field. The general form of the coded packets is $\sum_{i=1}^{n} \alpha_i \times p_i$. Here, p and α are the packets and random coefficients, respectively. The packet p_i can be an original packet or a coded packet. Using random linear network coding, we are able to decode the random coded packets using any set of n linearly independent coded packets. In order to decode the packets, Gaussian elimination for solving a system of linear equations can be used. Random linear network coding can be used in a variety of applications, such as providing reliable transmissions and increasing the throughput of networks. It also simplifies the content placement on storages.

B. Video Coding Scheme

In order to provide a fluid data model and simplify the distribution of the video packets, we perform intra-layer coding inside each video. For this purpose, we partition each video into equal size segments of packets, and linearly code the packets of each segment using random coefficients. Figure 3(a) shows the packets of a video, which are partitioned to a set of segments. Figure 3(b) shows the encoded video. The coefficients are not shown in the figure for simplicity. For example, $p_1+p_2+p_3+p_4$ means $\alpha_1 p_1+\alpha_2 p_2+\alpha_3 p_3+\alpha_4 p_4$, where α_1 to α_4 are random coefficients. Therefore, the encoded packets of each segment in Figure 3 are different.

Intra-layer network coding helps us to simplify the placement of the video packets on the clouds. Without network coding, the placement becomes a hard problem, as we need to decide which packets of each video should be stored on each cloud. Moreover, this determines which packet of each video should be transmitted to a user from the servers, and which of them should be transmitted from the clouds. In contrast, in the case of random linear network coding, each packet has the same contribution and importance; as a result, once a user receives a sufficient number of encoded packets, he can decode them and retrieve the original packets.

We store the packets uniformly from each segment of the video m on a storage cloud. This enables the storage clouds to serve any users watching video m, regardless of their playback time. Using this scheme, in order to store a fraction f of a video on a storage cloud, we store $f \times n$ random linearly coded packets of each segment on the storage cloud. For instance, in order to store half of the coded video in the example, Figure 3(b), on a storage, we store 2 random linear coded packets of each video segment on the storage. Note that the 2 stored coded packets of each segment in Figure 3(c) are different, since they have different random coefficients. We did not show these coefficients for simplicity. Assuming that the rate of video m is r_m, the storage cloud can supply at the rate of $f \times r_m$ to the users that watch video m.

Delivering the coded packets of the current segment from the storage clouds and the servers to a user with different delays might result in a video lag problem. The reason is that the user cannot decode the segment until it receives a sufficient number of coded packets. In order to resolve this problem, each user needs to buffer the received coded packets of the segments and delay the playback of the video for a specific amount of time. Using this approach, the differences of the transmission delays do not result in a playback lag. We do not address computing the buffering time, since that is beyond the scope of our current work.

IV. VIDEO-ON-DEMAND STREAMING WITH STORAGE CLOUDS

In our model, downloading from the server nodes and the storage clouds has a cost. In the case of direct downloading from the servers, the download cost includes the storage and bandwidth cost of the server to the users. In the case of indirect downloading, the downloading cost includes downloading the videos from the server nodes to the storage clouds, and downloading the video stored on the cloud by the users. Consequently, it might be efficient to pay the cost to download the videos from the servers to the clouds, and the storage lease cost, and then, reuse the videos stored on the clouds several times. However, because of the storage cost of the storage clouds, it might not be efficient to store unpopular

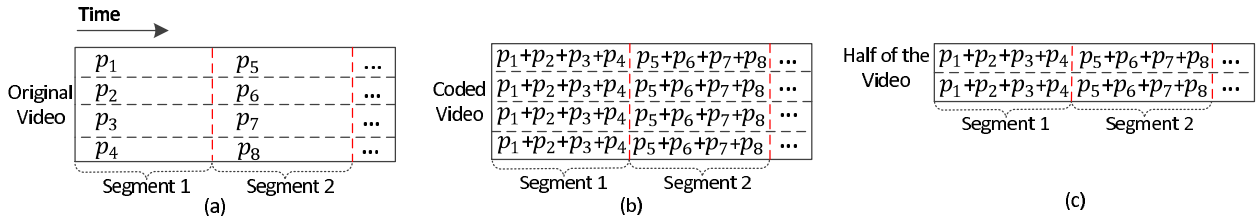


Fig. 3. Network coding scheme. The coefficients are not shown for simplicity; (a) Segmentation of an original video; (b) Linear network coding inside each segment; (c) storing a half portion of the video.

videos on the clouds. Moreover, depending on the costs and the current stored videos on the storage clouds, the economic policy might be the partial storing of some videos. These challenges make the optimization problem hard, even in the case of static networks.

One storage cloud that has the smallest cost might be sufficient to minimize the total streaming cost. However, in our formulations, we consider the general case that multiple clouds can be used. The idea behind using multiple clouds is that, some regions might have some local cloud systems with a smaller access (bandwidth) cost for that region. Moreover, by considering multiple clouds, the optimization becomes more general, and the single cloud is just a simplification of the scheme that we propose. We first formulate the problem of minimum cost video streaming, using storage clouds as a linear programming. We then modify it to the case of multi-resolution video streaming.

A. Single-layer Videos Streaming

We can model this optimization problem in the case of intra-layer coding as in the linear programming in Figure 4. The objective function (1) is minimizing the total cost of downloading the videos, which consists of downloading and storage costs. Here, $w_j(t)$, $w_k(t)$, $z_j(t)$, and $z_k(t)$ represent the total downloading cost and the storage cost of the cloud j and server k at time t, respectively. In the set of Constraints (2), we multiply the cost of downloading from cloud j for different regions by the downloading rate of the links, and the rate of the videos to compute the total downloading cost from storage cloud j. The downloading cost from a server includes the downloading cost to the users and the storage clouds, which are computed in the set of Constraints (3).

In the set of Constraints (4), we multiply the storage lease cost by the fraction of stored video and the size of the video. The summation in Equation (5) is the total storage cost of server k due to the services provided directly to the users. Here, we divide the download rate of video m from the server by the expected number of users in region i to find the average service rate to each user that region. For each server, we take the maximum value of its service rate over different regions to find the fraction of storage that is required to be turned on (For example, if for multiple regions, a server needs to serve half of a video that is stored on 2 storages; one of the storages can be turned off). We then multiply the maximum value by the size of the video and the storage cost.

The fraction of each video stored on a storage cloud node at time t cannot exceed the fraction of the video that is downloaded from the server nodes at time t plus the fraction of stored video at the previous time slot $t-1$, which is represented as the set of Constraints (6). Each cloud cannot provide a video to the users more that the portion of the video that is stored on it, which is represented as the set of Constraints (7). We divide the download rate for each region by the expected number of users served by the cloud j, to compute the average download rate provided for each user in that region. Each user needs to download its requested video at least at the rate of the video to be able to decode it. For this purpose, we use the set of Constraints (8) to make sure that the total download of the expected users in region i that request video m is not less than the rate of video m. The variable $y_{jml}(t)$ is the fraction of video m downloaded by cloud j from server k. As a result, its value should be in the range of $[0, 1]$, as shown in the set of Constraints (9).

Theorem 1: The proposed linear programming in Figure 4 can be solved in a polynomial time.

Proof: In the case that, the number of variables and constraints are a linear function of the input size, the solution of a linear programming optimization can be calculated in polynomial time [19]. Therefore, we need to show that the number of constraints and variables in our proposed optimization are polynomial.

The number of variables $x_{jim}(t)$, $x_{kim}(t)$, and $y_{kjm}(t)$ are equal to $|H| \times |R| \times |M| \times T$, $|S| \times |R| \times |M| \times T$, and $|S| \times |H| \times |M| \times T$, respectively. Moreover, the number of $f_{jm}(t)$ are equal to $|H| \times |M| \times T$. We have T variables w and z for each server and storage cloud. As a result, the number of w and z variables are equal to $2 \times (|S| + |H|) \times T$.

The number of Constraints (2) and (3) are equal to $|H| \times T$ and $|S| \times T$, respectively. Also, we have $|H| \times T$ set of Constraints (4). For each storage cloud, video, and time, we have one Constraint (6), so in total $|H| \times |M| \times T$ constraints. The number of Constraints (7) and (8) are equal to $|R| \times |H| \times |M| \times T$ and $|R| \times |M| \times T$, respectively.

The set of Constraints (5) can be converted to the following equivalent linear form:

$$z'_{km}(t) \geq \frac{x_{kim}(t)}{E_{im}(t)}, \quad \forall k, t, m, i : k \in S, i \in R, t \in [1, T]$$

$$z_k(t) \geq \sum_{m \in M} z'_{km}(t) v_m c_{2k}(t), \quad \forall k, t : k \in S, t \in [1, T]$$

where, $z'_{km}(t)$ is an axillary variable for converting the max operation in Constraint (5) to a linear form. Consequently, the number of variables $z'_{km}(t)$ are equal to $|S| \times |M| \times |T|$. Moreover, the number of above two set of constraints are equal to $|S| \times |M| \times |R| \times T$ and $|S| \times T$. ■

$$\min \sum_{t=1}^{T} \Big[\sum_{j\in H} [w_j(t) + z_j(t)] + \sum_{k\in S} [w_k(t) + z_k(t)] \Big] \tag{1}$$

$$s.t.\ w_j(t) \geq \sum_{i\in R} \sum_{m\in M} x_{jim}(t) r_m c_{1ji}(t), \quad \forall j,t : j \in H, t \in [1,T] \tag{2}$$

$$w_k(t) \geq \sum_{i\in R} \sum_{m\in M} x_{kim}(t) r_m c_{1ki}(t) + \sum_{j\in H} \sum_{m\in M} y_{kjm}(t) v_m c_{1kj}(t), \quad \forall k,t : k \in S, t \in [1,T] \tag{3}$$

$$z_j(t) \geq \sum_{m\in M} f_{jm}(t) v_m c_{2j}(t), \quad \forall j,t : j \in H, t \in [1,T] \tag{4}$$

$$z_k(t) \geq \sum_{m\in M} \max_{i\in R} \Big(\frac{x_{kim}(t)}{E_{im}(t)}\Big) v_m c_{2k}(t), \quad \forall k,t : k \in S, t \in [1,T] \tag{5}$$

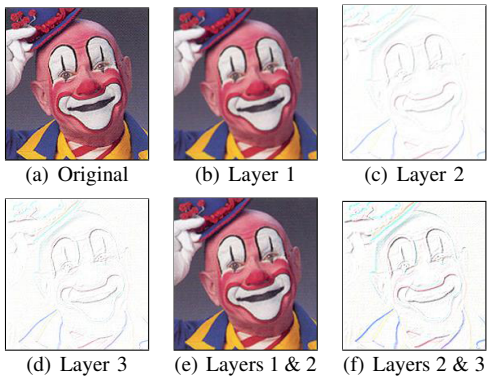
$$f_{jm}(t) - f_{jm}(t-1) \leq \sum_{k=1}^{|S|} y_{kjm}(t), \quad \forall j,m,t : j \in H, m \in M, t \in [1,T] \tag{6}$$

$$\frac{x_{jim}(t)}{E_{im}(t)} \leq f_{jm}(t) r_m, \ \forall i,j,m,t : j \in H, i \in R, m \in M, t \in [1,T] \tag{7}$$

$$\sum_{k\in S} x_{kim}(t) + \sum_{j\in H} x_{jim}(t) \geq E_{im}(t) r_m, \quad \forall i,m,t : i \in R, m \in M, t \in [1,T] \tag{8}$$

$$0 \leq y_{jm}(t) \leq 1, \quad \forall j,m,t : j \in H, m \in M, , t \in [1,T] \tag{9}$$

Fig. 4. Optimization in the case of single-layer videos.



(a) Original (b) Layer 1 (c) Layer 2

(d) Layer 3 (e) Layers 1 & 2 (f) Layers 2 & 3

Fig. 5. Multi-resolution video with 3 layers.

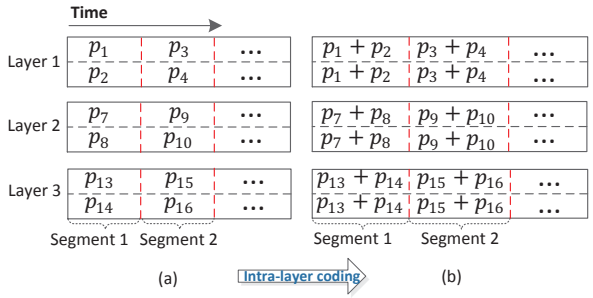


Fig. 6. Network coding in the case of multi-resolution videos. The coefficients are not shown for simplicity; (a) A multi-resolution video with 3 layers; (b) intra-layer linear network coding among the packets of each layer.

B. Multi-resolution Video Streaming

In this section, we extend the proposed optimization to the case of multi-resolution video coding [20]–[22]. In *multi-resolution codes* (MRC), videos are typically divided into a base layer and a set of enhancement layers [21], [23]. The base layer (layer 1, or in some references, layer 0) is required to watch the video, but the enhancement layers augment the quality of the video streaming. Having access to more layers increases the quality of the video; however, the i-th enhancement layer is almost not useful unless all of the enhancement layers with a smaller index are provided. The reason we say almost useless is that the output of the decoding contains just some shadows, without the required details. In Figure 5(a), an original image is shown, and Figures 5(b)-(d) show the constructed layers from this image. The first layer is the most important layer, which is required to watch the video. Figures 5(c) and (d) depict that layers 2 or 3 cannot be used without all of the layers with a smaller index. Figure 5(f) shows that decoding layers 2 and 3 together, without layer 1, is almost useless, as well. Combining layer 2 to and layer 1 together increases the quality of the image, as illustrated in Figure 5(e).

Using multi-resolution videos has three advantages. Firstly, the users can select a lower video quality once they have a video lag problem because of a connection problem or bandwidth limitation. Secondly, the users can save their 4G data if they have download limitations. Thirdly, switching to a lower quality reduces the load on the server and the storage clouds, which reduces the total cost of the system.

Our coding scheme is similar to that of the single layer video. The only difference is that each video has several layers, and each layer has its own segmentation. The coding is performed among the packets of the same segment and the same layer, which is called intra-layer network coding. Consider the 3-layers video in Figure 6(a). Each segment of each layer contains 2 packets. For instance, the packets in the

$$\min \sum_{t=1}^{T} \big[\sum_{j\in H} [w_j(t) + z_j(t)] + \sum_{k\in S} [w_k(t) + z_k(t)] \big] \tag{10}$$

$$s.t.\ w_j(t) \geq \sum_{i\in R} \sum_{m\in M} \sum_{l=1}^{L} r_{ml} x_{jiml}(t) c_{1ji}(t), \quad \forall j, t : j \in H, t \in [1,T] \tag{11}$$

$$w_k(t) \geq \sum_{i\in R} \sum_{m\in M} \sum_{l=1}^{L} x_{kiml}(t) r_m c_{1ki}(t) + \sum_{j\in H} \sum_{m\in M} \sum_{l=1}^{L} y_{kjml}(t) v_{ml} c_{1kj}(t), \quad \forall k, t : k \in S, t \in [1,T] \tag{12}$$

$$z_j(t) \geq \sum_{m\in M} \sum_{l=1}^{L} f_{jml}(t) v_{ml} c_{2j}(t), \quad \forall j, t : j \in H, t \in [1,T] \tag{13}$$

$$z_k(t) \geq \sum_{m\in M} \sum_{l=1}^{L} \max_{i\in R} \big(x_{kiml}(t) / \big[\sum_{l'=l}^{L} E_{iml'}(t) \big] \big) v_{ml} c_{2k}(t), \forall k, t :\in S, t \in [1,T] \tag{14}$$

$$f_{jml}(t) - f_{jml}(t-1) \leq \sum_{k=1}^{|S|} y_{kjml}(t) \quad \forall j, m, t : j \in H, m \in M, t \in [1,T] \tag{15}$$

$$x_{jiml}(t) / \big[\sum_{l'=l}^{L} E_{iml'}(t) \big] \leq f_{jml}(t) r_{ml}, \ \forall i, j, m, t : j \in H, i \in R, m \in M, t \in [1,T] \tag{16}$$

$$\sum_{k\in S} x_{kiml}(t) + \sum_{j\in H} x_{jiml}(t) \geq r_m \sum_{l'=l}^{L} E_{iml'}(t) \quad \forall i, m, l, t : i \in R, m \in M, 1 \leq l \leq L, t \in [1,T] \tag{17}$$

$$0 \leq y_{jml}(t) \leq 1 \quad \forall j, m, l, t : j \in H, m \in M, 1 \leq l \leq L, t \in [1,T] \tag{18}$$

Fig. 7. Optimization in the case of multi-resolution videos.

first segment of layer 1 are p_1 and p_2. As a result, the coded packets over the first segment of layer 1 are in the form of $\alpha_1 p_1 + \alpha_2 p_2$, where α_1 and α_2 are random coefficients. The encoded layered video is shown in Figure 6(b). Note that we did not show the coefficients in Figure 6(b) to simplify the example. In this example, in order to store half of a video layer, we can store one random linear coded packet of each segment of that layer. In this paper, we do not consider intra-layer network coding. The reason is that, in our previous work [24] we found that, in practice, the effect of inter-layer network coding is marginal. As a result, because of more computational complexity of joint inter- and intra- layer network coding, we avoid it in this work.

We assume that each layer of the video m has a constant streaming rate r_m and size v_m, and each video contains L video layers. We extend the variables used to formulate the single-layer video streaming to the case of multi-resolution videos, as follows. Variables $x_{jiml}(t)$ and $x_{kiml}(t)$ are the downloading rate from storage cloud j and server k to the users in region i over the l-th layer video m at time slot t, respectively. We use variables $y_{kjml}(t)$ and $f_{jml}(t)$ to represent the fraction of downloaded layer l of video m at time slot t from server k to cloud j and the fraction of the video layer stored on the storage cloud, respectively. The variable E_{iml} represents the expected number of users in region i that requested the first l layers of video m. Moreover, we assume that the size of the l-th layer of video m and its rate are equal to v_{ml} and r_{ml}, respectively. The download and storage costs are the same as in the previous section.

The optimization problem in the case of intra-layer coding can be modeled as the linear programming in Figure 7. The objective function is similar to (1) and the the set of constraints (15) are similar to (6). The difference between the optimization in Figure 7 and Figure 4 is that each video in 7 has several layers. In the set of Constraints (14), in order to find the average number of users that need to receive layer l, we divide $x_{kiml}(t)$ by set of users that their number of requested layers is more than or equal to l layers. The reason is than, as mentioned before, the video layers have a prefix format. As a result, the users that request $l' \geq l$ layers need to receive video layer l to decode the layers 1 to l'. For the same reason we divide $x_{kiml}(t)$ by the summation of the expected users that request l to L number of layers in the set of Constraints (16).

Theorem 2: The proposed linear programming in Figure 7 can be solved in polynomial time.

Proof: The proof is similar to that in Theorem 1. The only difference is that the number of some variables and constraints is multiplied by the number of layers L. As a result, the number of variables and constraints is still a linear function of the input size. ■

V. Simulations

In this section, we compare the proposed VoD streaming using storage clouds with VoD streaming without using storage clouds. We first report our result in the case of single-layer videos. Then, we present the simulation result for multi-layer videos.

A. Simulation Setting

In order to evaluate our proposed method, we developed a simulator in the MATLAB environment. For the case of streaming without storage clouds, we modify our proposed method by removing the variables and constraints that correspond to the storage clouds. The constraints that are removed include Constraints (2), (4), (6), (7), and (9). Moreover, w_j and z_j are removed from the objective function (1). The last change is that we no longer have the second summations in Constraints (3) and (8).

We run the simulations on 100 settings, in which the expected number of requests for each video, storage, and bandwidth costs are chosen randomly. The ranges of the random numbers are mentioned for each figure in the next sections. The number of servers, clouds, regions are set to 3. The number of videos in the simulations are set to 30. The rate and the size of each video is set randomly in the range of $[1, 2]$ Mb/s and $[3.6, 7.2]$ Gb, respectively. The reason for not choosing a larger number of videos is that it just increases the run time of the simulations, and it does not have any effect on the performance rate of our cloud-based streaming compared to the direct streaming scheme.

B. Single-Layer Video Streaming

In the first experiment, we evaluate the effect that the storage cost has on the total VoD streaming cost in Figure 8(a). The expected requests for each video is randomly set to a number in the range of $[1, 10]$. The bandwidth cost from the servers to the users, servers to the cloud, and cloud to the users are chosen in the ranges of $[2, 5]$. The storage cost of the servers and the clouds are shown in the x-axis of the figure. As it is expected, increasing the storage cost of the clouds and the servers increases the total cost of both of the direct and cloud-based streaming. Figure 8(a) shows that the total cost of the direct streaming is more than 20% more than that of the cloud-based streaming.

We measure the effect of cloud bandwidth cost on the total streaming cost in Figure 8(b). The storage cost of the clouds and servers are set in the range of $[2, 5]$. Also, the bandwidth cost from the servers to the users and servers to the clouds are set randomly in the range of $[2, 5]$. As the bandwidth cost of the clouds increases, the total streaming cost of the cloud-based streaming method becomes closer to that of streaming without cloud. However, the total cost of the cloud-based streaming never exceeds that of the direct streaming, since even in the cloud-based streaming, we can provide the videos to the users directly from the servers in the case that the total cost of the cloud-based streaming is more than that of the direct streaming. In Figure 8(b), the streaming cost of the direct streaming is up to 50% more than that of the cloud-based streaming.

In the next experiment, we study the effect that the number of expected requests has on the total cost of our proposed method. We set the storage cost of the clouds and servers in the range of $[2, 5]$. The bandwidth cost from the servers to the users, servers to the clouds, and clouds to the users are chosen in the ranges of $[1, 8]$, $[5, 8]$, and $[2, 5]$, respectively. Figure 9(a) shows that the total streaming cost of both of the methods increase as the expected requests increase. Moreover,

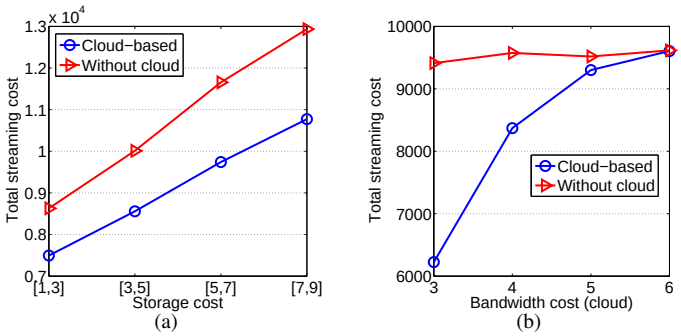


Fig. 8. The total VoD streaming cost in the case of single-layer videos. Number of servers, clouds, and regions is equal to 3. $C_{1ki} \in [2, 5]$, $C_{1kj} \in [2, 5]$, $E_{im} \in [1, 10]$. (a) $C_{1ji} \in [2, 5]$. (b) $C_{2k} \in [2, 5]$, $C_{2j} \in [2, 5]$.

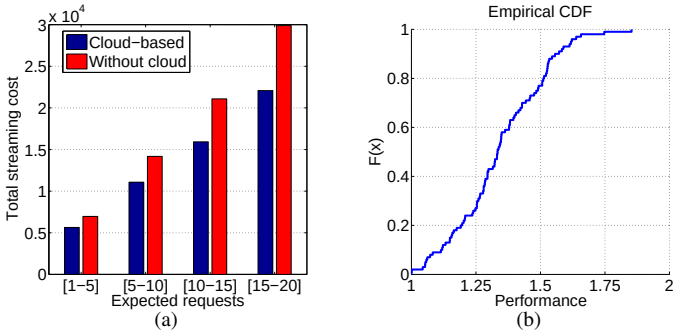


Fig. 9. Single-layer video streaming. Number of servers, clouds, and regions is equal to 3, $C_{1ki} \in [1, 8]$, $C_{1kj} \in [5, 8]$, $C_{1ji} \in [2, 5]$, $C_{2k} \in [2, 5]$, $C_{2j} \in [2, 5]$. (a) Effect of expected number of requests on the total cost. (b) Empirical CDF of the performance of cloud-based streaming over streaming without cloud, $E_{im} \in [15, 20]$.

the efficiency of using storage clouds increases as the expected number of requests increases. The reason is that, the download cost of cloud-based streaming consists of bandwidth cost from servers to the clouds and clouds to the users. As a result, even if the bandwidth cost of the clouds is less than the bandwidth cost of the servers to the user, indirect downloading might not be efficient in the case of few expected requests for a video. In contrast, once the popularity of the videos increases, it becomes more beneficial to pay the cost of downloading the videos from the servers to the clouds once, and then use the less expensive links (if exist) of the clouds to provide the videos to the users.

Figure 9(b) shows the empirical CDF of the cloud-based streaming over streaming without cloud. We define the performance as the cost of streaming without cloud divided by the cost of cloud-based streaming. The figure shows that in 50% of the cases, the performance of cloud-based streaming is between 1.35 and 1.85. Moreover, in 20% of the cases, the performance of our approach is between 1.50 and 1.85.

C. Multi-Resolution Video Streaming

For the case of video streaming without storage clouds, we modify the proposed linear programming by removing the variables and constraints related to the storage clouds. For this goal, we remove Constraints (11), (13), (15), (16), and (18). Furthermore, w_j and z_j are removed from the objective

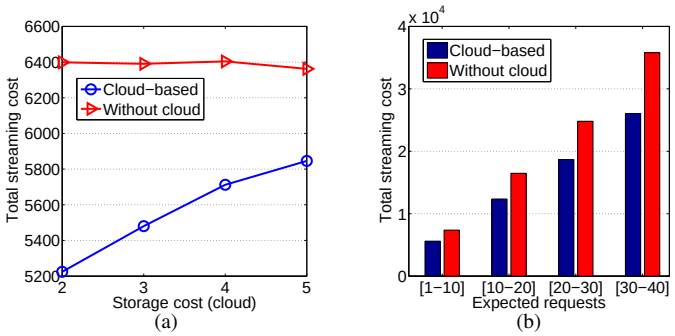


Fig. 10. Multi-resolution video streaming. Number of servers, clouds, and regions is equal to 3. $C_{1ki} \in [2,5]$, $C_{2k} \in [2,5]$. (a) $E_{im} \in [1,10]$, $C_{1kj} \in [2,5]$, $C_{1ji} \in [2,5]$. (b) $C_{1kj} \in [1,8]$, $C_{1ji} \in [5,8]$, $C_{2j} \in [2,5]$.

function (1). Finally, we do not have the second summations in Constraints (12) and (17). In the simulations with multiple-resolution videos, we set the number of layers to 3.

In Figure 10(a), we evaluate the effect of cloud storage cost on the total streaming cost. The setting is shown in the caption of the figure. It is clear that the storage cost of the clouds do not have any affect on the total cost of direct streaming. The figure depicts that as storage cost of the clouds increases, the gap between the direct streaming and cloud-based streaming decreases. In this figure, the total cost of the direct streaming is up to 23% more than that of the streaming using the help of the clouds.

Figure 10(b) shows the effect of expected number of requests on the total cost. Similar to Figure 9(b), the total cost of both direct streaming and cloud-streaming methods increase as the expected number of requests increases. Moreover, the efficiency of the cloud-based dreaming increases as we increase the expected number of requests.

VI. Conclusion

One form of application on the Internet with a high traffic is video streaming. With the increase in the energy demand of the data centers that provide the video files to the users, the importance of using renewable and green energy is increasing. In order to minimize the energy cost of the data centers, which are geographically distributed all over the world, we need to try to reduce the load on the servers at the time durations to which the green sources of energy, such as sun and wind, are not available. One way to achieve this goal is to lease storage clouds and store the popular videos on these clouds. In this paper, we study the problem of cloud leasing in order to minimize the total video streaming cost. We model the problem as an optimization problem, which becomes a linear programming problem in the case of linear energy cost functions. We extend our solution to the case of multi-resolution video coding, which provides the users with videos in different quality levels, based upon requests.

Acknowledgment

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

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] Í. Goiri, R. Beauchea, K. Le, T. Nguyen, M. Haque, J. Guitart, J. Torres, and R. Bianchini, "Greenslot: scheduling energy consumption in green datacenters," in *ACM STOC*, 2011, pp. 302–311.

[4] C. Li, A. Qouneh, and T. Li, "Characterizing and analyzing renewable energy driven data centers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 1, pp. 323–324, 2011.

[5] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang, and Y. Chen, "Greencloud: a new architecture for green data center," in *ACM ICAC*, 2009, pp. 29–38.

[6] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," *Arxiv preprint arXiv:1109.4179*, 2011.

[7] 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.

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

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[16] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM*, 2006.

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

[18] 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.

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

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

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

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

[23] 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.

[24] P. Ostovari, A. Khreishah, and J. Wu, "Multi-layer video streaming with helper nodes using network coding," in *IEEE MASS*, 2013.