Robust Wireless Delivery of Scalable Videos using Inter-layer Network Coding

Pouya Ostovari and Jie Wu
Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122

***Abstract*—As the popularity of wireless devices (e.g. smartphones and tablets) and watching videos over the Internet is increasing rapidly, delivering high quality videos to users over wireless links is becoming an important application. One of the main challenges of multicasting in wireless networks to multiple receivers is the diversity of the receivers. In a wireless network, the users have different wireless channel conditions, and as a result, they experience different packet delivery rates. In order to handle these heterogeneous channels, multi-resolution videos are used to deliver videos at multiple quality levels. The recent research on multi-resolution codes show that triangular network coding can increase the quality of the received videos by the users. In this paper, considering the dependencies among different temporal and spatio (resolution) layers of a video, we propose a new two-dimensional triangular network coding that performs network coding between the temporal and spatio layers. Our simulation results show the effectiveness of our two-dimensional coding schemes compared to the previous one-dimensional network coding schemes.**

Keywords*—*Streaming, scalable video coding, unequal error protection, inter-layer network coding, wireless networks.

I. INTRODUCTION

Recently, watching videos over the Internet has become increasingly popular. Recent studies show that the most dominant traffic on the Internet is multimedia streaming. For example, 20-30% of the web traffic on the Internet is from YouTube and Netflix [1]. A large portion of the users that watch videos use wireless devices, e.g. smartphones and tablets. This creates a new challenge in terms of efficiently using the bandwidth resources, e.g. WiFi and 4G, and to deliver a high quality video to the users.

Clearly, unicasting an independent stream to each receiver is not an efficient approach, since it does not take advantage of the broadcast nature of the wireless medium. As a result, video multicasting has recently received a lot of attention. However, the main challenge in video multicasting is in regard to receivers with heterogeneous channel conditions. If the source transmits a single video stream at the lowest bit rate supported by the receiver, the users will experience the video quality of the receiver with the worst channel. On the other hand, if the source transmits at a higher bit rate, some of the users will not be able to watch the video.

In order to solve this problem, *scalable video coding* (SVC) [2] is proposed. In SVC, which is also called *multi-layer codes* [3] or *multi-resolution codes* (MRC), videos are divided into a base layer and enhancement layers. The base layer is the most important layer and is required to watch the video. The enhancement layers can augment the quality of the decoded video. If a user receives more layers, he can watch the video at a higher quality. However, because of the dependency among the layers, the i-th layer is useless without its preceding layers. H.264/MPEG-4 or advanced video coding (AVC) [4] is one of the most commonly used video compression format. H.264/SVC, which is an extension of the H.264/AVC, supports spatial, temporal, and quality scalabilities. Because of the hierarchical dependencies between the video layers, the quality of the received video is greatly affected by packet loss. As a result, providing robustness against packet losses is very critical in multicasting SVC videos.

In order to maximize the number of useful layers that can be retrieved by the users, the work in [5] combines SVC with triangular NC. In triangular NC [10], the coded layer 1 only contains packets of the first layer, and coded layer two consists of the packets of the first two layers. In general, coded layer i is coded using the packets of the first i layers. The idea behind this kind of coding is that, because of the dependency between the layers, layer i is not useful unless layers 1 to $i-1$ are available. Triangular NC allows the retrieval of useful video layers from more combinations of the received transmissions, which improves the number of decoded layers.

In order to improve the number of decodable layers and to increase the quality of the received video, we propose two-dimensional coding schemes. In contrast with the work in [5], which performs the triangular NC on the quality layers, we use triangular NC for coding both the spatio and temporal layers together. The main challenge in combining inter-layer coding with SVC is to find the optimal coding strategy for a given channel condition, and total number of transmissions that can be performed before the deadline of a group of pictures (GoP). We propose several two-dimensional NC schemes, and show their efficiency compared to a one-dimensional NC. Moreover, in order to reduce the searching complexity for the optimal distribution of the transmissions, we propose a theorem for checking the decodability of the coded layers. Using this theorem, an algorithm is proposed that checks the layers that can be decoded without using Gaussian elimination.

The rest of the paper is organized as follows: we introduce related work in Section II. The setting and objective is introduced in Section III. We study different coding schemes in Section IV. In Section V, we propose our search algorithm for finding the optimal coding strategy. Section VI presents the simulation results, and Section VII concludes the paper.

II. RELATED WORK

Unequal error protection (UEP) has been widely employed on multi-resolution videos. The authors in [6] proposed a UEP scheme by exploiting the unequal importance of the temporal and quality layers. They use a genetic algorithm to distribute the redundancy to different layers. In [7], a performance metric is proposed to measure the importance of the quality and

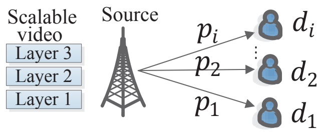


Fig. 1. The system architecture.

temporal layers, which is used in to assign redundancy to different layers. The advantage of combining multi-resolution coding with NC has been studied by [8], [9]. In [8], the authors propose a video multicast with joint NC and video interleaving. They partition the temporal layers of GoPs with equal importance to the same partition, and perform NC among the layers of the same partition. The authors in [9] show that the gain in the case of triangular NC is more than that of the non-coding and random linear coding.

The authors in [5] show that the performance of the previously proposed triangular coding schemes are poor, and they use the estimated number of decodable layers as a measure to find how many layers should be coded to enhance the coding performance. In order to find the optimal triangular coding strategy (transmissions distribution) in the case of multiple users, the authors create a reference table which contains the number of decodable layers for a given delivery rate and triangular coding strategy. They also propose a set of optimization techniques to reduce the time complexity of the search for the optimal solution. These optimizations include transmission distribution granularity, and checking the decodability of a set of packets without using Gaussian elimination.

III. Setting

We consider a single-hop wireless network in Fig. 1, in which a source broadcasts a scalable video to a set of destinations over an erasure channel. We represent the i-th destination as d_i. The delivery rate from the source to d_i is represented as p_i. The transmitted video is coded using H.264/SVC (scalable video coding) video compression standard, which can encode a video into temporal, spatial and quality layers [8], [9]. In this paper we consider temporal and spatio (resolution) layers, and assume that the encoded video contains m and n temporal and spatio layers, respectively. The temporal layers are correspondent to the time sequence frames. In contrast, the spatio video layers are correspondent to the resolution of the video frames. Receiving more layers increases the quality of the decoded video (note that the video quality is different than the quality layers). However, a spatio layer is useless without all of the spatio layers with a smaller index. The same characteristic exists between the temporal layers.

The top picture in Fig. 2(a) shows the temporal layers of a video and their dependencies. The bottom picture depicts the same dependencies in a different way. In this figure, layers l_1 to l_4 represent the temporal layers. As Fig. 2(b) shows, each temporal layer itself contains some spatio layers. The first spatio layer of each temporal layer is the base layer. The next spatio layers are enhancement layers, which can increase the quality of the video. Similar dependencies as those in Fig. 2(a) exist between the enhancement spatio layers of different temporal layers, which are not shown for simplicity.

We represent the total number of transmissions for a GoP that the source can perform as X. The distribution of

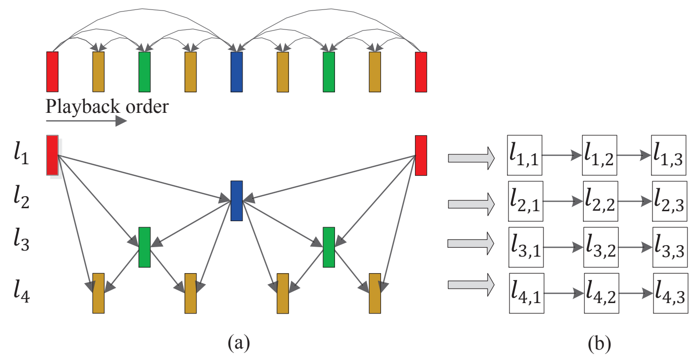


Fig. 2. Scalable video coding with hierarchical B pictures. (a) Temporal layers. (b) Spatio layers.

TABLE I. The set of symbols used in this paper.

Notation	Definition
p_i	The delivery rate from source to the i-th destination.
l_i	The i-th temporal video layer.
$l_{i,j}$	The j-th spatio layer of the i-th temporal layer.
$L_{i,j}$	The triangular coded layers over $l_{1,1}$ to $l_{i,j}$.
$r_{i,j}$	Number of packets in layer $l_{i,j}$.
X	Total number of transmissions for a GoP.
$x_{i,j}/y_{i,j}$	The number of transmitted/received triangular coded packets over layer $L_{i,j}$.
m/n	The number of temporal/spatio layers.

these transmissions to the layers is shown as $(x_{1,1}, ..., x_{m,n})$. Moreover, the number of received packets for different layers is represented as $(y_{1,1}, ..., y_{m,n})$. We assume that layer $l_{i,j}$ contains $r_{i,j}$ packets. As the different destinations have different wireless channel conditions, they receive different numbers of layers; thus, the quality of the decoded videos at the destinations are different. These qualities not only depend on the number of received layers, but also on which layers have been received. Our objective in this work is to maximize the total quality of the received video by the destinations. In our simulations, we use both the number of decoded layers and the PSNR of the decoded videos to evaluate our methods. The set of symbols used in this paper is summarized in Table I.

IV. Robust Network Coding Schemes

Our video transmission method has two phases. In the first phase, we decide about the NC scheme to code the video layers. In the second phase, the transmissions are distributed among the possible network coded packets of the selected NC scheme such that the quality of the decoded videos at the destinations is maximized. Before discussing about the NC schemes, we provide some preliminary on NC.

A. Background on Network Coding

In this paper, random linear network coding (RLNC) is used to code the video packets. In RLNC [11], coded packets are random linear combinations of the original packets over a finite field. Each coded packet is in the form of $\sum_{i=1}^{k} \alpha_i \times P_i$. Here, α and P are random coefficients and the packets, respectively. Also, k is the number of packets that are coded together. The source transmits random coded packets over the k packets. In order to decode the coded packets, the destinations need to receive k linearly independent coded packets. They can use Gaussian elimination to decode the coded packets.

Using the general form of RLNC for layered videos has a problem. Consider 3 single-packet video layers l_1, l_2, and l_3, which contain packets P_1, P_2, and P_3, respectively. If we perform RLNC on these layers, we should combine these 3 packets together, and the coded packets will be in form of $\alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$. Using RLNC, two cases can happen. If a destination receives 3 linearly independent coded packets, it will be able to decode the packets and retrieve all of the 3 layers. In contrast, in the case of receiving less than 3 linearly independent coded packets, the destination cannot decode and retrieve any packet. In order to solve this problem, triangular NC is proposed. In triangular coding, the i-th coded layer is a linear combination of the packets in the first i layers. Using triangular coding, if a destination receives 2 linearly independent coded packets over packets P_1 and P_2, it can decoded the packets and retrieve layers l_1 and l_2. Moreover, it is possible to receive layer l_1 alone.

B. Network Coding Schemes for Video Transmission

Without losing the generality, we describe the NC schemes in the case of single packet per layer. In Fig. 2, we have 4 temporal and 3 spatio layers. We can reshape the video layers structure in Fig. 2 to Fig. 3(a). Each row and column in Fig. 3(a) represents a spatio and a temporal layer, respectively. Also, the figure shows the dependencies between the layers. For example, layer $l_{2,2}$ is dependent on layer $l_{1,2}$ and $l_{2,1}$ directly. Moreover, layer $l_{2,2}$ is indirectly dependent on layer $l_{1,1}$. It can be inferred from Fig. 3(a) that layer $l_{a,b}$ is not useful without layers $l_{i,j} : \forall 1 \le i \le a, 1 \le j \le b$. For single-layer videos, there is only one triangular coding scheme, as described in the previous section. However, for two-dimensional layered videos, there are several ways to code the layers. In the following sections we discuss the possible coding approaches.

1) Canonical Triangular Coding: The work in [5], performs triangular NC on the quality layers. In canonical triangular coding, the k-th coded layer is a linear coded layer over the first k quality layers of all of the temporal layers. In other words, the coded layer k is in the form of $\sum_{i=1}^{m} \sum_{j=1}^{k} \alpha \times l_{i,j}$. In this paper, we consider spatio layers instead of the quality layers. As a result, if we apply this method in our setting, we should replace quality layers with spatio layers.

2) Vertical Triangular Coding: In the case that we want to give more priority to the spatio layers, we can first perform the triangular coding scheme on the spatio layers of the first temporal layer, as shown in Figs. 3(a)-(c). In this method, the first coded layer contains only layer $l_{1,1}$. The second coded layer is coded over layers $l_{1,1}$ and $l_{1,2}$. In general, the first n coded layers are in the form of $\sum_{j=1}^{k} \alpha \times l_{1,j},\ \forall k : 1 \le k \le n$.

Figs. 3(a)-(c) show the first 3 coded layers using the vertical coding scheme. As depicted in Fig. 3(d), after coding the spatio layers of the first temporal layer, we perform triangular coding on the temporal layers. The next coded layers contains layers $l_{1,1}$ to $l_{3,3}$ (Fig. 3(g)), and the last coded layer contains all of the layers as shown in Fig. 3(h).

3) Horizontal Triangular Coding: This scheme is the reverse of the vertical triangular coding scheme. We first apply triangular coding on the temporal layers of the first spatio layer. Then, for the second spatio layer, we code all of the temporal layers of the first and second spatio layer together. This process is repeated for the other spatio layers. In contrast with the vertical coding scheme, horizontal coding gives more priority to the temporal layers than the spatio layers.

4) Diagonal and Zigzag Triangular Coding: Diagonal scheme gives the same priority to the temporal and spatio layers. Figs. 3(e)-(h) show our diagonal triangular coding scheme. The first layer is $l_{1,1}$, which is not coded with the other layers. The second layer contains layers $l_{1,1}$, $l_{1,2}$, $l_{2,1}$, and $l_{2,2}$. In the same way, the third coded layer is coded over layers $l_{i,j} : 1 \le i, j \le 3$. If n and m are not equal, the shape of the layers becomes non-square as depicted in Fig. 3(h). In this case, after reaching the last spatio or temporal layer, depending on whether n or m is smaller, we just increase the index of the other dimension in the next coded packets. Assuming that n is smaller than m, the general form of the coded packets is: $\sum_{i,j \in [1,k]} \alpha_{i,j} l_{i,j}, \forall 1 \le k \le n$ and $\sum_{i \in [1,k], j \in [1,n]} \alpha_{i,j} l_{i,j}, \forall n < k \le m$. In Zigzag coding, we first increase i and code packets $l_{1,1}$ to $l_{i+1,j}$. In the next possible coding, we increase j and code packets $l_{1,1}$ to $l_{i+1,j+1}$.

Receiving more layers increases the video watching experience, and reduces video distortion. However, if we want to have a comparison between the temporal and spatio layers, we can consider receiving more temporal layers as a smoother playback of the videos, and more spatio layers as frames with a higher resolution. Therefore, the horizontal triangular coding provides a smooth playback, and the vertical triangular coding gives more priority to the resolution of the frames. Moreover, the diagonal triangular coding gives the same importance to the resolution and the smoothness of the videos.

V. Optimal Transmissions Distribution

After deciding on the coding scheme, we need to distribute the transmissions among the coded layers of that coding scheme. Consider a video with 4 temporal and 3 spatio layers that is coded using a horizontal coding scheme. Using the horizontal coding scheme, we will have 6 coded layers $L_{1,1}$ to $L_{4,1}$, $L_{4,2}$, and $L_{4,3}$. Considering the limited time and bandwidth, the question is that how many times the source node needs to transmit the packets of each coded layer. Similar to the work in [5], in order to find the optimal solution, we check all of the possible distributions and select the distribution that maximizes the total gain. With the purpose of reducing the time complexity of checking all of the possible distributions, we can assign the transmissions to different layers with granularity g.

In our method, we generate a reference (look-up) table which shows the decoding probability of each layer for each possible distribution of the transmissions and a delivery rate. The reference table needs to be generated just once, and the source can use it to find the optimal distribution in any delivery rate scenario. Having the reference table, the source can easily search the reference table to find the best distribution scheme in the case of multiple destinations with different delivery rates. In the following, we first propose an algorithm for creating a reference table in the case of single packet per layer. We then extend the algorithm to the case of multiple packets per layer.

A. Avoiding Gaussian Elimination

We denote a transmission distribution as $(x_{1,1}, ..., x_{m,n})$, and the number of received transmissions by a destination as

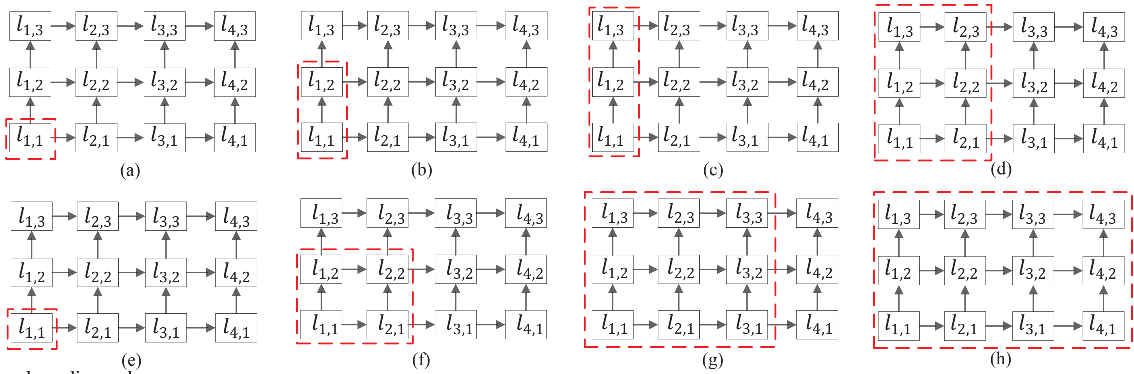


Fig. 3. Network coding schemes.

$(y_{1,1}, ..., y_{m,n})$. In order to check the probability of decoding each layer for a given transmission distributions, we need to check which layers can be decoded for each possible outcome (transmission receptions). For example, assuming that the transmissions of 2 coded layers $L_{1,1}$ and $L_{2,1}$ is equal to $x = (1, 1)$, the possible outcomes are $y = (0, 0)$, $y = (1, 0)$, $y = (0, 1)$, and $y = (1, 1)$. One approach to check the layers that can be decoded is using Gaussian elimination. In order to reduce the time complexity of reference table creation, we want to avoid Gaussian elimination. In [5], a theorem is proposed to calculate the number of decodable layers for a given reception outcome, without Gaussian elimination for the case of one-dimensional layered-videos. We first prove the theorem, as it gives insight for a theorem in the case of two-dimensional triangular coding and its proof.

Theorem 1: Under a one-dimensional triangular coding and a reception outcome $(y_1, ..., y_m)$, the user can decode layers up to l_a if and only if $\sum_{i=h}^{a} y_i \geq a-h+1,\ \forall\ h : 1 \leq h \leq a$.

Proof: Assume that we have a coded packets over layers l_1 to l_a with a rank equal to a, which means that layers l_1 to l_a can be decoded. Also, assume that for a given h, we have $\sum_{i=h}^{a} y_i < a-h+1$. As a result, the number of coded packets over layers l_1 to l_{h-1} is equal to $\sum_{i=1}^{h-1} y_i > a-(a-h+1) = h-1$. Consequently, we have at least h coded packets over the first $h-1$ layers. It means that these coded packet cannot be linearly independent, which contradicts the assumption. ∎

For example, the client can decode layers l_4 and its predecessor layers l_1, l_2, and l_3, if and only if $y_4 \geq 1$, $y_4 + y_3 \geq 2$, $y_4 + y_3 + y_2 \geq 3$, and $y_4 + y_3 + y_2 + y_1 \geq 4$.

Theorem 2: Under any two-dimensional triangular coding scheme, for a reception outcome $(y_{1,1}, ..., y_{m,n})$, the user can decode layers up to $l_{a,b}$ if and only if:

$$\sum_{i=h}^{a} \sum_{j=k}^{b} y_{i,j} \geq (a-h+1)(b-k+1) \quad \forall h, k{:}1 \leq h \leq a, 1 \leq k \leq b \tag{1}$$

Proof: Assume that we have ab coded packets over layers $l_{1,1}$ to $l_{a,b}$ with a rank equal to ab. In other words, layers $l_{1,1}$ to $l_{a,b}$ are decodable. Moreover, assume that h and k are the largest indices such that $\sum_{i=h}^{a} \sum_{j=k}^{b} y_{i,j} < (a-h+1)(b-k+1)$ As a result, the number of coded packets over the layers with smaller indices that h or k is equal to:

$$\sum_{i=1}^{h-1} \sum_{j=1}^{b} + \sum_{i=1}^{a} \sum_{j=1}^{k-1} > ab - (a-h+1)(b-k+1)$$
$$= ak-a+hb-hk+i-b+k-1 = a(k-1)+b(h-1) - (h-1)(k-1)$$

On the other hand, the number of packets involved in these coded layers is equal to $a(k-1)+b(h-1)-(h-1)(k-1)$. Thus, the number of coded packets is greater than the number of original packets. It means that these coded packets cannot be linearly independent, which contradicts the assumption. ∎

As an instance, the client can decode layers $l_{2,2}$ and its predecessor layers $l_{1,1}$, $l_{1,2}$, and $l_{2,1}$, if and only if $y_{2,2} \geq 1$, $y_{2,2}+y_{1,2} \geq 2$, $y_{2,2}+y_{2,1} \geq 2$, and $y_{2,2}+y_{1,2}+y_{2,1}+y_{1,1} \geq 4$.

Algorithm 1 Distribution Algorithm for Diagonal Coding

```
// dist(x, i, j, X)
if i ≤ m or j ≤ n then
    for k = 1 to X do
        x_{i,j} = k
        if i ≤ m and j ≤ n then
            dist(i + 1, j + 1, X − k)
        else if i ≤ m then
            dist(i + 1, j, X − k)
            else dist(i, j + 1, X − k)
    else RTC(x)
```

B. Reference Table Creation

In order to create the reference table, we first produce all the possible distributions of the X transmissions given a specific coding scheme. The distribution algorithm for diagonal coding is shown in Algorithm 1. In order to distribute X transmissions, we call the algorithm with parameters X, i=1, and j=1. The algorithm recursively calls itself until i and j reach m and n, respectively. During each run, the remaining transmissions are assigned to each valid triangular coding. After producing a possible distribution, the reference creation algorithm (RTC) is called to calculate the successful decoding probability of each layer and to fill the reference table.

In order to find the successful decoding probability of a layer, we need to check if a given layer is decodable in the different possible delivery outcomes. The RTC algorithm receives a given distribution x and checks the possible outcomes. It first sets the decodability of each layer to 1. It then

Algorithm 2 Reference Table Creation ($RTC(x)$)

```
for each possible outcome y out of x transmissions do
    for i = 1 to m do
        for j = 1 to n do dec(i, j) = 1
    for a = 1 to m do
        for b = 1 to n do
            for h = 1 to a, k = 1 to b do
                if Σ_{i=h}^{a} Σ_{j=k}^{b} y_{i,j} < (a−h+1)(b−k+1) then
                dec(a, b) = 0
    for i = 1 to m do
        for i = 1 to n do
            if dec(i, j) = 1 then a = i, b = j
    for p = 0.05 to 1, step = 0.05 do
        q = prob(receiving y out of x) transmissions
        RT(a, b, p) = RT(a, b, p) + q
```

checks Theorem 2 for each layer to find which layers are not decodable under each delivery outcome. Then, the RTC algorithm calculates the probability of that specific outcome happening, and add this probability to the cell of the reference table RT that is correspondent to the decodable layer with the largest indices. The algorithm calculates this probability in terms of different packet delivery rates. In this paper, we consider 0.05 as the granularity of the packet delivery rates. The details of the RTC algorithm are shown in Algorithm 2.

C. Extension to Varying Number of Packets per Layer

In the previous sections, we assumed that each layer contains a single packet. However, the different layers of the videos might be encoded at different bitrates, and as a result, contain different number of packets. Theorem 2 and the proposed algorithm can be easily extended to the case of multiple packets per layer. We just need to change the decoding condition in Theorem 2. Considering $r_{i,j}$ packets in layer $l_{i,j}$, the decoding condition in Theorem 2 becomes:

$$\sum_{i=h}^{a}\sum_{j=k}^{b} y_{i,j} \geq \sum_{i=h}^{a}\sum_{j=k}^{b} r_{i,j} \quad \forall\, h,k : 1 \leq h \leq a, 1 \leq k \leq b$$

VI. Evaluation

We evaluate our methods by comparing them with the Percy method [5]. In Percy, the triangular coding is performed among the spatio layers (in [5], quality layers are mentioned, but Percy can be applied on spatio layers instead on the quality layers). Thus, the first coded layer contains layers $l_{1,1}$ to $l_{m,1}$. The next coded layers are similar to those of the horizontal coding. In order to evaluate the methods, we implemented a simulator in the MATLAB environment. Moreover, we use JSVM reference software for encoding and decoding the video and measuring the PSNR of the decoded videos.

We use the Bus video trace in our evaluation, which is shown in Fig. 4. The resolution and the frame rate of the video that we use are equal to 352×288 pixels and 30 frames per second, respectively. We partition the video to 4 and 3 temporal and spatio layers. The resolution of the spatio layers 1, 2, and 3 are equal to 176×144, 320×240, and 352×288, respectively. Figs. 4(a) and (b) depict the decoded videos using the base layer and the 3 spatio layers.

(a) (b)

Fig. 4. (a) Decoded video using the base spatio layer. (b) Decoded video using all of the 3 spatio layers.

TABLE II. PSNR OF THE DECODED LAYERS.

n \ m	1	2	3	4
1	31.24	32.85	34.30	35.62
2	31.72	34.12	36.97	40.6
3	39.51	49.4	67.11	99

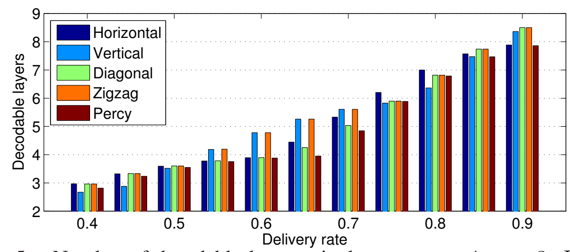
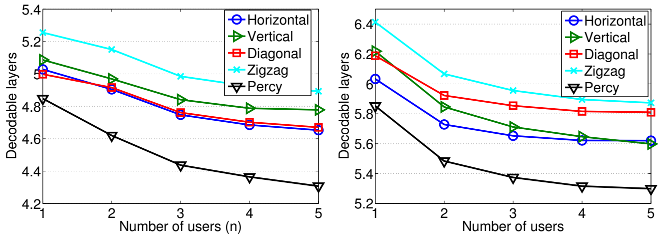


Fig. 5. Number of decodable layers, single user. $m = 4$, $n = 3$, $X = 12$.

We evaluate the methods in terms of number of decodable layers and PSNR. We calculate the PSNR of the decoded videos using the PSNRStatic function provided in JSVM. We use the decoded video using all of the layers as the reference to calculate the PSNR. Before we calculate the PSNR of a decoded video, we upsample the decoded video to make its frame rate and resolution consistent with that of the reference video. The PSNR of the decoded videos for different layers are shown in Table II. We run the simulations for 1000 random delivery rates, and show the average output in the plots.

1) Number of Decodable Layers: In the first experiment, we measure the number of decodable layers in the case of single user in Fig. 5. We assume single packet per layer, and set the total number of transmissions to 12. As expected, the figure shows that the number of decodable layers increases as the delivery rate of the link increases. Moreover, the number of decodable layers of the horizontal coding is always more than that of the Percy method. The reason is that the coding in the Percy method is a subset of the coding possibilities in the horizontal approach. However, for some of delivery rates, the Percy method delivers more layers that that of the vertical approach. Fig. 5 shows that the number of decodable layers in the horizontal and vertical methods are up to 12% and 33% more than that of the Percy method, respectively.

Fig. 6(a) shows the average number of decodable layers for different numbers of users. The total number of transmissions in this experiment is set to 12, and the reliability of the links are randomly chosen in the range of $[0.4, 0.9]$. As the figure illustrates, the average number of decodable layers by the users decreases as we increase the number of users. The reason is that, as we increase the number of users, the diversity of the channels' delivery rates increases. In this case, it is probable



(a) (b)

Fig. 6. Number of decodable layers in the case of multiple users. $m = 4$, $n = 3$. (a) $X = 12$. (b) $X = 14$.

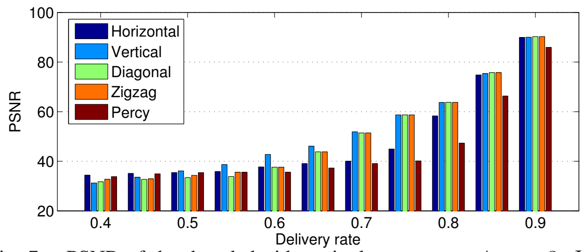


Fig. 7. PSNR of the decoded video, single user. $m = 4$, $n = 3$, $X = 18$.

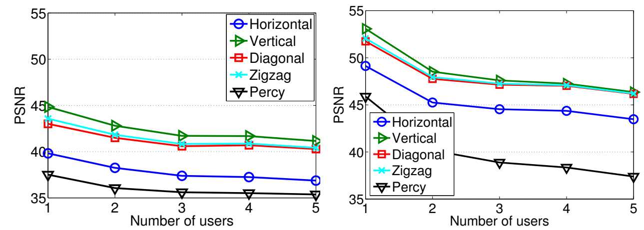
that a good distribution choice for a user is not appropriate for another user. Therefore, we cannot satisfy all of the users at the same time. It can be inferred from the figure that the number of decodable layers in the Zigzag coding method is up to 14% more than that of the Percy method.

We increase the total number of transmissions to 14 and repeat the previous experiment. The result is shown in Fig. 6(b). It is clear that a higher number of transmissions results in more decodable layers. The figure illustrates that the Zigzag coding scheme and the Percy method has the highest and lowest number of decodable layers, respectively.

2) PSNR: In the previous experiments, we just checked the number of decodable layers. However, the relation of number of decodable layers and the quality of the video is not always linear. Therefore, we also measure the PSNR of the decoded videos. In Fig. 7, the PSNR of the decoded videos in the case of single user are shown for different delivery rates. The figure shows that the PSNR of the decoded video increases as the delivery rate of the link increases. Fig. 7 depicts that the PSNR of the vertical, diagonal, and the Zigzag coding methods are up to 45% more than that of the Percy method.

Fig. 8(a) shows the average PSNR of the decoded videos in the case of multiple users. The number of users varies from 1 to 5. Moreover, the reliability of the links are randomly chosen in the range of $[0.4, 0.9]$. The Fig. shows that the PSNR of the vertical coding method is up to 20% more than that of the Percy method. Moreover, the average PSNR decreases as the number of users increases. As mentioned before, the reason is that as we increase the number of users, the diversity of the channels' delivery rate increases, which results in a decrease in the average number of layers delivered to the users.

In the next experiment, we increase the number of transmissions to 20, and repeat the previous experiment. The result is shown in Fig. 8(b). As the number of transmissions increases, the gap between our coding methods decreases, which is due to the larger number of received layers in all of the methods. However, the PSNR of the Percy method is still much less than that in our methods. As Table II shows, in the video



(a) (b)

Fig. 8. PSNR of the decoded video in the multiple users. $m = 4$, $n = 3$. (a) $p \in [0.4, 0.6]$. (b) $p \in [0.6, 0.8]$.

trace that we use, the spatio layers have more of an impact on the PSNR. However, the Percy method cannot deliver partial temporal layers, and cannot assign more transmissions to the spatio layers than the temporal layers, which results in less PSNR compared to our methods.

VII. Conclusion

As a result of increasing popularity of video streaming, wireless video multicasting is becoming an important application. However, the diversity of users' channel conditions is a challenge to video multicasting. Multi-resolution video coding is an efficient method to address this challenge. In this work, we propose a two-dimensional NC method to increase the users' experience. In our coding scheme, we combine both temporal and spatio layers together, and we introduce a search algorithm to find the optimal distribution of the transmissions to different coded layers. Through simulations results, we show the effectiveness of our two-dimensional coding scheme, as compared to the one-dimensional NC scheme.

References

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

[2] M. Effros, Universal multiresolution source codes, IEEE Transactions on Information Theory 47 (6) (2001) 2113–2129.

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

[4] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the h. 264/AVC standard, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1103–1120.

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

[6] H. Ha, C. Yim, Two-dimensional channel coding scheme for MCTF-based scalable video coding, Y. Wang and T. Fang and L-P. Chau and K. Yap 9 (1) (2007) 37–45.

[7] H. Ha, C. Yim, Layer-weighted unequal error protection for scalable video coding extension of h. 264/AVC, IEEE Transactions on Consumer Electronics 54 (2) (2008) 736–744.

[8] H. Wang, S. Xiao, C. Kuo, Robust video multicast with joint network coding and video interleaving, Journal of Visual Communication and Image Representation 21 (2) (2010) 77–88.

[9] H. Wang, S. Xiao, C. Kuo, Random linear network coding with ladder-shaped global coding matrix for robust video transmission, Journal of Visual Communication and Image Representation 22 (3) (2011) 203–212.

[10] P. Ostovari, A. Khreishah, J. Wu, Cache content placement using triangular network coding, in: IEEE WCNC, 2013.

[11] P. Ostovari, J. Wu, A. Khreishah, Network Coding Techniques for Wireless and Sensor Networks, Springer, 2013.