Computer Networks 95 (2016) 115–126

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Robust wireless transmission of scalable coded videos using two-dimensional network coding

Pouya Ostovari[1,*], Jie Wu[2]

*Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122, United States*

ARTICLE INFO

*Article history:*
Received 9 March 2015
Revised 10 November 2015
Accepted 23 November 2015
Available online 11 December 2015

*Keywords:*
Streaming
Scalable video coding
Unequal error protection
Inter-layer network coding
Triangular network coding
Wireless networks

ABSTRACT

With the rapid increase in the popularity of wireless devices (e.g. smartphones and tablets) and watching videos over the Internet, delivering high quality videos to mobile users over wireless links is becoming an important application. Diversity of the receivers is a main challenge of multicasting in wireless networks, where variant channel conditions of end users lead to different packet delivery rates. In order to handle these heterogeneous channel conditions, multi-resolution videos can be used to deliver videos at multiple quality levels. The recent research studies 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 the concept of two-dimensional triangular network coding that performs network coding between the temporal and spatio layers. We also propose the concept of coding speed, which realizes difference balances between video playback smoothness and resolution. We evaluate our proposed two-dimensional coding schemes through simulations using two video sequences.



## 1. Introduction

Nowadays, watching videos over the Internet has become increasingly popular. Recent studies on the Internet traffic show that the most dominant form of traffic on the Internet is multimedia streaming [2–4]. A large portion of the users that watch videos use wireless devices that are connected to the Internet, 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.

It is clear that in wireless networks, 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. Therefore, 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 address the heterogeneous channel conditions in an efficient way, *scalable video coding* (SVC) [5–7] has been proposed. In SVC, which is also called *multi-layer codes* or *multi-resolution codes* (MRC), videos are divided into a base layer and enhancement layers [8,9]. 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

* Corresponding author. Tel.: +2672692931.
*E-mail addresses:* ostovari@temple.edu (P. Ostovari), jiewu@temple.edu (J. Wu).

[1] This work is supported in part by NSF grants CNS 149860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963.

[2] Part of this work appeared was presented in IEEE WCNC, 2015 [1].

http://dx.doi.org/10.1016/j.comnet.2015.11.022
1389-1286/

a higher quality. However, because of the dependency among the layers, the $i$th enhancement layer is useless without its preceding layers.

H.264/MPEG-4 or advanced video coding (AVC) [10] is a video compression format, and is one of the most commonly used formats for compression and distribution of video content. H.264/SVC [11], 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.

The works in [12–18] provide unequal error protection for the layers, depending on their importance and contributions to the video quality. Network coding (NC) is an effective method for providing error protection against packet losses. In NC, coded packets are a linear combination of the original packets. Assuming that $k$ packets are coded together, a receiver node can decode and retrieve the original packets if it receives $k$ linearly independent packets. Using NC, depending on the delivery rate of the links, the source node transmits a larger number of packets than $k$, so that the destination node receives at least $k$ linearly independent coded packets. The decoding is done using Gaussian elimination for solving a system of linear equations.

In order to increase the number of useful layers that can be retrieved by the users and as a result the quality of the decoded videos, the work in [19] combines SVC with triangular NC [20,21]. In triangular NC, 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 triangular 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.

We propose the concept of two-dimensional triangular network coding to improve the number of decodable layers and the quality of the received videos. In contrast with the work in [19], which performs the triangular network coding only on the quality layers, our two-dimensional video coding performs network coding on 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 the total number of transmissions that can be performed before the deadline of a group of pictures (GoP). We discuss the different possible coding schemes, and propose the concept of coding speed. The coding speed determines the different possible coded layers that the source node can produce. Considering a given coding speed, we propose a search algorithm to find the optimal distribution of the transmissions among different possible coded layers.

In this paper, we have the following contributions. We propose several two-dimensional network coding schemes, and show their efficiency compared to a one-dimensional network coding. Moreover, in order to reduce the searching complexity for the optimal distribution of the transmissions, we propose a remark for checking the decodability of the coded layers. Using this remark, an algorithm is proposed that checks the number of layers that can be decoded without using Gaussian elimination.

The remainder of the paper is organized as follows: we introduce related work on network coding and wireless video streaming in Section 2. The setting and objective is introduced in Section 3. We study different coding schemes in Section 4, and propose the concept of coding speed. In Section 5, we propose our search algorithm for finding the optimal coding strategy. Section 6 presents the simulation results, and Section 7 concludes the paper.

## 2. Related work

The first time *network coding* (NC) [22] was used for wired networks, as to solve the bottleneck problem in the single multicast problem. In [23] a useful algebraic representation of the linear NC problem is proposed. It is shown that if the coefficients of the coded packets are selected randomly, with a high probability the coded packets will be linearly independent [24]. NC can be used to provide robust transmissions in error-prone environments.

Unequal error protection (UEP) has been widely employed on multi-resolution videos. The authors in [25] 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 [14], a performance metric is proposed to measure the importance of the quality and temporal layers, which are based on the error propagation of a packet loss in a given layer. The authors use the hill climbing method and their performance metric to efficiently assign redundancy to different layers.

The advantage of combining multi-resolution coding with network coding has been studied in [12,13]. In [12], the authors propose a video multicast with joint network coding and video interleaving. They put the temporal layers of different GoPs that have the same importance to the same partition, and perform network coding among the layers of the same partition. The amount of FEC assigned to each partition depends on the importance of that partition. The authors in [13] study the cases of non-coding, random linear coding, and triangular coding, and show that the gain in the case of triangular network coding is more than that of the other cases. They also propose a triangular coding scheme for multi-hop networks.

The work in [26] combines ARQ with the triangular coding scheme and rate control. The authors assume that each user requests the desired number of layers from the source node. In [27], the authors apply multi-generation coding, which is similar to triangular coding. In their proposed method, they distribute the redundant transmissions between different coded layers evenly.

The authors in [19] show that the performance of the previously proposed triangular inter-layer 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

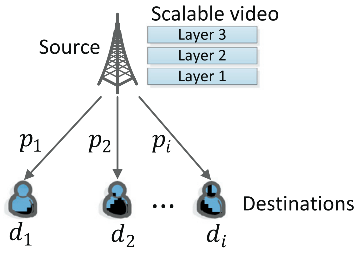



**Fig. 1.** The system setting.

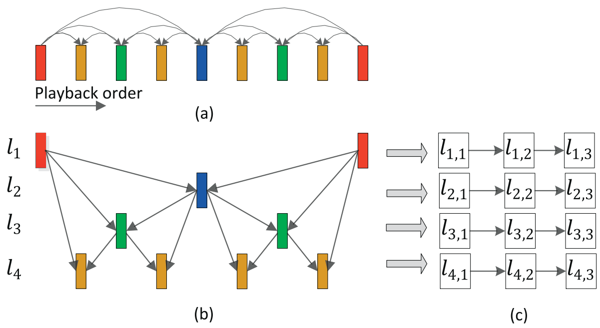



**Fig. 2.** Scalable video coding with hierarchical B pictures. (a) and (b): Temporal layers. (c) Spatio layers.

complexity of the search for the optimal solution. These optimizations include controlling the number of packets that should be assigned to different layers in each round of the algorithm run (granularity), and checking the decodability of a set of packets without using Gaussian elimination. Using the reference table, the source node can search for the optimal solution given the delivery rates of the receivers.

## 3. Setting

We consider a single-hop wireless network shown in Fig. 1, in which a source node multicasts a scalable video to a set of destinations over erasure wireless channels. 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 [12,13]. In this paper we consider temporal and spatio (resolution) video 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 of the videos. In contrast, the spatio video layers are correspondent to the resolution of the video frames. Receiving more spatio and temporal layers increases the quality of the decoded video (note that the video quality is different than the quality layers). However, a spatio layer is almost useless without having access to all of the spatio layers with a smaller index. The same characteristic exists between the temporal video layers.

Fig. 2(a) shows the temporal layers of a video and the dependencies between them. Fig. 2(b) depicts the same dependencies in a different way. In this figure, layers $l_1$ to $l_4$ represent the temporal layers. As Fig. 2(c) shows, each

**Table 1**
The set of symbols used in this paper.

| Notation | Definition |
|---|---|
| $d_i$ | The $i$th destination node. |
| $p_i$ | The delivery rate of the link between the source and the $i$th destination node. |
| $l_i$ | 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. |
| $x_{i,j}$ | The number of transmitted triangular coded packets over layer $L_{i,j}$. |
| $y_{i,j}$ | The number of received triangular coded packets over layer $L_{i,j}$. |
| $m$ | The number of temporal layers. |
| $n$ | The number of spatio layers. |
| $\alpha$ | Random coefficient for the linear coded packets. |

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) and (b) 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 node can perform as $X$. The distribution of these transmissions to the layers is shown as $(x_{1,1}, \ldots, x_{m,n})$. Moreover, the number of received packets for different layers is represented as $(y_{1,1}, \ldots, y_{m,n})$. We assume that layer $l_{i,j}$ contains $r_{i,j}$ packets.

As the different users (destinations) have different wireless channel conditions (delivery rates), they receive different numbers of layers. Consequently, the quality of the decoded videos at the destination nodes 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 destination nodes. 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 1.

## 4. Robust network coding schemes

Our video transmission method has two phases. In the first phase, we decide about the NC scheme that should be used to code the video layers. In the second phase, the total number of transmissions are distributed among the possible network coded packets of the selected NC scheme. Before discussing about the NC schemes on video layers, we provide some preliminary on NC.

### 4.1. Background on network coding

In this paper, random linear network coding (RLNC) is used to code the video packets. In RLNC, coded packets are random linear combinations of the original packets over a finite field of size $q$. Each coded packet is in the form of $\sum_{i=1}^{k} \alpha_i \times P_i$. Here, $\alpha_i$ and $P_i$ are random coefficients and the packets, respectively. The coefficients are selected with a uniform distribution over a finite field. Also, $k$ is the number

$$\text{a}\quad \begin{cases} P_1 \\ P_2 \\ P_3 \end{cases} \qquad \text{b}\quad \begin{cases} P_1 \\ P_1 + P_2 \\ P_1 + P_2 + P_3 \end{cases}$$

**Fig. 3.** (a) The 3 original packets. (b) The 3 triangular coded packets. The coefficients are not shown for simplicity.

of packets that are coded together. Instead of transmitting the original packets, the source node generates and transmits random coded packets over the $k$ packets. In order to decode the coded packets and retrieve the original packets, the destination nodes 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 node receives 3 linearly independent coded packets, it will be able to decode 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. As a result, none of the layers will be decodable. It means that using RLNC, there is no way to receive a subset of the layers. The probability of receiving a certain number of linearly independent coded packet is impacted by the value of $q$.

In order to solve this problem, triangular network coding is proposed. In triangular coding, the $i$-th coded layer is a linear combination of the packets in the first $i$ layers. The triangular coded packets of the previous example are shown in Fig. 3. The coefficients are not shown for simplicity. Using triangular coding, if a destination node 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.

The triangular NC can be easily generalized to the case of multiple packets per layer. Assume that layer $l_i$ contains $r_i$ packets. In order to code layers $l_1$ to $l_i$, we code all of the packets in these layers. For example, the coded packets over layer $l_1$ are random linear combination of the packets in $l_1$. As a result, in order to retrieve layer $l_1$, $r_1$ coded packets over this layer are required. Moreover, in order to code layers $l_1$ and $l_2$, we code the packets of these 2 layers together.

### 4.2. Network coding schemes for video transmission

Without loss of 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 and simplify the layer dependencies to Fig. 4(a). Each row and column in Fig. 4(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. 4(a) that layer $l_{a,b}$ is not useful unless we have access to layers $l_{i,j}: \forall 1 \le i \le a,\ 1 \le j \le b$. In the case of one-dimensional layered 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 choose which layers to code together. In the following sections we discuss the possible coding approaches.

#### 4.2.1. Canonical triangular coding

The work in [19], 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.

#### 4.2.2. 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 Fig. 4(a)–(c). In this method, the first coded layer contains only layer $l_{1,1}$. The second coded layer is coded over layers $l_{1,1}$

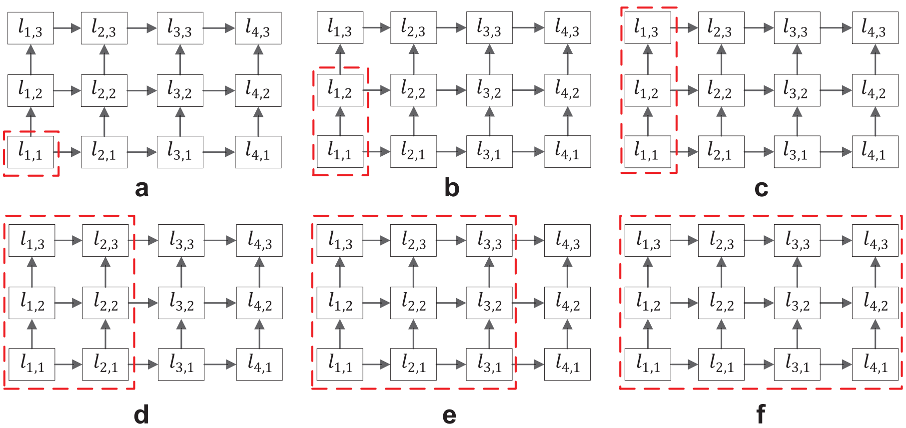



**Fig. 4.** Vertical triangular network coding scheme.

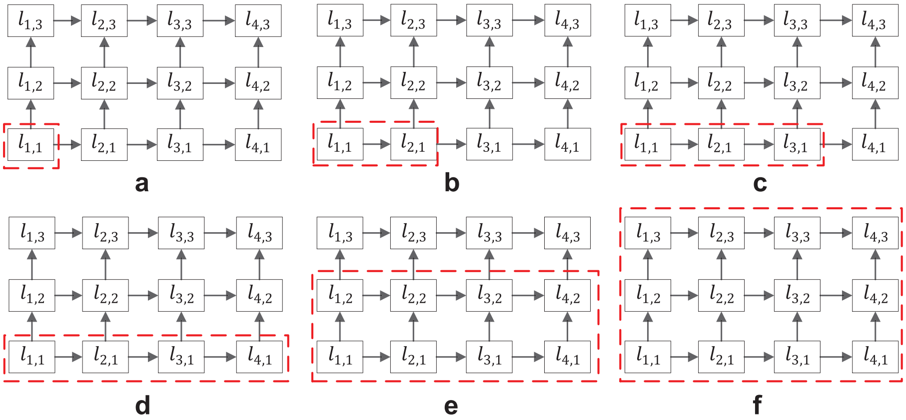



**Fig. 5.** Horizontal triangular network coding scheme.

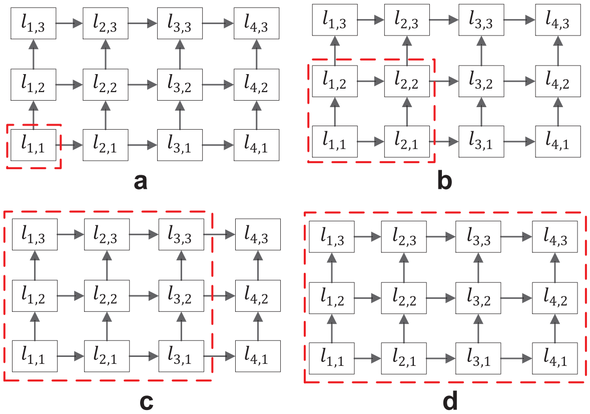



**Fig. 6.** Diagonal triangular network coding scheme.

and $l_{1,2}$. In general, the first $n$ coded layers are in the form of $\sum_{j=1}^{k} \alpha \times l_{1,j},\ \forall k : 1 \le k \le n$.

Fig. 4(a)–(c) show the first 3 coded layers using the vertical coding scheme. As depicted in Fig. 4(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. 4(e)), and the last coded layer contains all of the layers as shown in Fig. 4(f).

### 4.2.3. 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. The set of possible codings in the case of 4 temporal and 3 spatio layers is shown in Fig. 5.

### 4.2.4. Diagonal triangular coding

This scheme gives the same priority to the temporal and spatio layers. Figs. 6(a)–(d) 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. 6(d). 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:

$$\sum_{i,j\in[1,k]} \alpha_{i,j} l_{i,j}, \quad \forall\, 1 \le k \le n$$

$$\sum_{i\in[1,k],\, j\in[1,n]} \alpha_{i,j} l_{i,j}, \quad \forall\, n < k \le m$$

### 4.2.5. General two-dimensional triangular coding

The coded layers in the general triangular coding are the union of the above schemes. In this scheme, any layer $l_{i,j}$ is coded with the layers $l_{1,1}$ to $l_{i,j}$. For example, as Fig. 4(b) shows, layer $l_{1,2}$ is coded with layer $l_{1,1}$. Moreover, in Fig. 6(b), layers $l_{1,1}$ to $l_{2,2}$ are coded together. If we want to code layer $l_{3,3}$, we code all of the layers with $i$ and $j$ in the range of 1 and 3. The idea behind this coding scheme is that because of the dependencies between the layers, layer $l_{i,j}$ is not useful unless the layers with smaller $i$ and $j$ indices are available to the user. In this coding scheme, we have $mn$ possible coded layers.

It is shown in [19] that finding the optimal distribution of the transmissions to the different layers is a difficult problem, even in the case of a one-dimensional scalable video. Therefore, it is not a straightforward task to find the optimal distribution in the general triangular coding. Due to too many coding possibilities in the case of general two-dimensional coding (specifically $mn$), the time complexity of finding the optimal distribution of the transmissions is high. As a result, we avoid this type of coding in our work.

### 4.2.6. Coding speed

In order to make the diagonal coding scheme more general, and at the same time keep the complexity of the method low, we introduce the concept of *coding speed* (CS). Coding speed is a triple $CS = (z_1, z_2, z_3)$. Here, $z_1$ shows the number of spatio layers that should be added to the coded layer

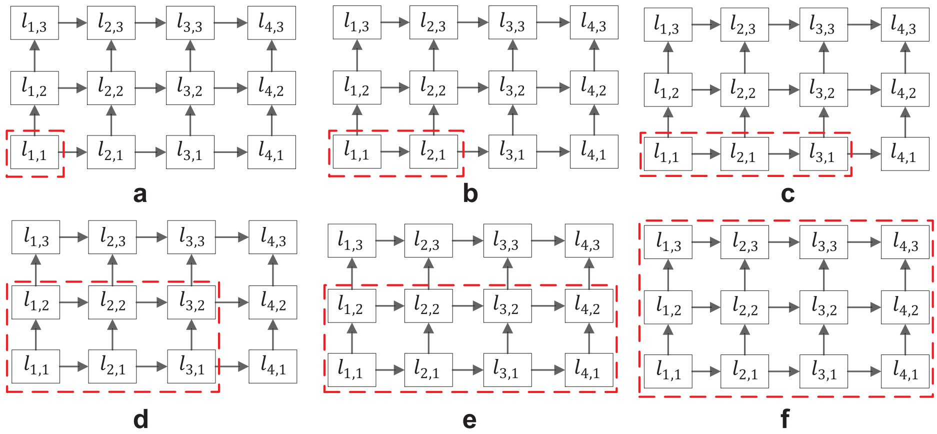



**Fig. 7.** Coding speed illustration, $CS = (1, 2, 1)$.

before a temporal layer be added. In the same way, $z_2$ represents the number of temporal layers that should be added to the coded layer before a spatio layer be added. Also, $z_3$ shows whether the coding direction is first horizontal or vertical. In more details, $s_3 = 0$ and $s_3 = 1$ mean that the coding direction is first vertical and horizontal, respectively. The possible codings with coding speed $CS = (1, 2, 1)$ are shown in Fig. 7. Here, $z_3 = 1$ means that the coding starts in the horizontal direction. Also, $z_1 = 1$ and $z_2 = 2$ mean that, after adding 2 temporal layers (horizontal), 1 spatio layer (vertical) will be added to the coding set. This process will be repeated until we reach the last temporal and spatio layers.

As shown in Fig. 7(a), the first coded layer only includes layer $l_{1,1}$. We need to first add 2 temporal layers to the coding set, and then add one spatio layer to the coding set. As shown in Fig. 7(b) and (c), layers $l_{2,1}$ and $l_{3,1}$ are added to the coding set. In Fig. 7(d), we add layer $l_{3,2}$ to the coding set. In order to add layer $l_{3,2}$, we also need to include layers $l_{1,2}$ and $l_{2,2}$. We perform the coding as an atomic procedure, which means, in each step, we just increase the height or length of the coding set.

We refer to this scheme as speed triangular coding (STC). The horizontal and vertical triangular coding schemes are subsets of the STC method with a coding speed equal to $CS = (m-1, n-1, 1)$ and $CS = (m-1, n-1, 0)$, respectively. The optimal CS depends on the video coding scheme and the rates of the layers. For example, in the cases that the temporal layers have more of an impact on the video quality than the spatio layers, e.g. videos that the similarity between their frames is low, more speed can enhance the quality of the video played by the users. In contrast, if the spatio layers are more important than the temporal layers, a speed less than 1 might be better.

### 4.3. Comparing the coding schemes

It is clear that receiving more layers increases the video watching experience. In other words, receiving more layers 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. Because of time complexity necessary to find the optimal solution using the general form of two-dimensional network coding is high, we do not use it in our work.

Referring to the above discussion, we can set up coding speed by performing a trade-off between playback smoothness and resolution of the video, or by considering the overall distortion. Coding speeds with $z_3 = 1$ and a large $z_2$ gives more priority to the smoothness. Also, coding speed with $z_3 = 0$ and large $z_1$ gives more importance to the resolution. In the extreme cases, the coding becomes similar to the vertical and horizontal triangular coding. Moreover, in the case of a coding speed equal to $CS = (1, 1, 1)$ or $CS = (1, 1, 0)$, the importance of the resolution and smoothness are the same. From an overall distortion perspective, we should try to minimize the distortion of the video.

## 5. 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 $L_{1,1}$ $L_{2,1}$, $L_{3,1}$, $L_{4,1}$, $L_{4,2}$, 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 [19], 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. It should be noted that the reference table needs to be generated just once, and the source node can use it to find the optimal distribution in any delivery rate scenario. Having the reference table, the source node 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 sections, 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.

### 5.1. Avoiding Gaussian elimination

We denote a transmission distribution as $(x_{1,1}, \ldots, x_{m,n})$, and the number of received transmissions by a destination node as $(y_{1,1}, \ldots, y_{m,n})$. In order to check the probability of successful 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 [19], a remark 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. There is a proof for the remark in [19]; however, we prove it here for two reasons. First, it gives an insight for a remark in the case of two-dimensional triangular coding and its proof. Moreover, our proof provides more details than those in [19]. Here, we assume a large field size, e.g. $q \geq 2^8$. As a result, as shown in [28], with a high probability, the generated random linear network coded layers are linearly independent. For a small field size $q$, the coded layers might not be linearly independent.

**Remark 1.** Under a one-dimensional triangular coding and a reception outcome $(y_1, \ldots, y_m)$, the user can decode layers up to $l_a$ if and only if:

$$\sum_{i=h}^{a} y_i \geq a - h + 1, \ \forall \, h : 1 \leq h \leq a \qquad (1)$$

**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 $y_4 + y_3 + y_2 + y_1 \geq 4$.

The Remark 1 gives us an insight for the decodability conditions in the case of two-dimensional coding:

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

$$\sum_{i=h}^{a} \sum_{j=k}^{b} y_{i,j} \geq (a - h + 1)(b - k + 1)$$
$$\forall \, h, k : 1 \leq h \leq a, 1 \leq k \leq b \qquad (2)$$

**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:

$$\begin{aligned} \sum_{i=1}^{h-1} \sum_{j=1}^{b} + \sum_{i=1}^{a} \sum_{j=1}^{k-1} &> ab - (a - h + 1)(b - k + 1) \\ &= ak - a + hb - hk + i - b + k - 1 \\ &= a(k - 1) + b(h - 1) - (h - 1)(k - 1) \end{aligned} \qquad (3)$$

On the other hand, the number of packets involved in these coded layers is equal to $a(k - 1) + b(h - 1) - (h - 1)(k - 1)$. As a result, the number of coded packets on these original layers 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 $y_{2,2} + y_{1,2} + y_{2,1} + y_{1,1} \geq 4$.

### 5.2. 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 among the layer, we should 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 of the algorithm, the remaining transmissions are assigned to each valid triangular coding. After producing a possible distribution, the reference creation algorithm

**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(1, 1, x)
```

$$\begin{bmatrix} q_{1,1,0.95} & q_{1,2,0.95} & \cdots & q_{1,n,0.95} \\ q_{2,1,0.95} & q_{2,2,0.95} & \cdots & q_{2,n,0.95} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m,1,0.95} & q_{m,2,0.95} & \cdots & q_{m,n,0.95} \end{bmatrix}$$

**Fig. 8.** Example of reference table. Delivery rate p = 0.95.

(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 creates the possible delivery outcomes. It first sets the decodability of each layer to 1. It then checks Remark 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 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.

**Algorithm 2** Reference Table Creation (RTC).

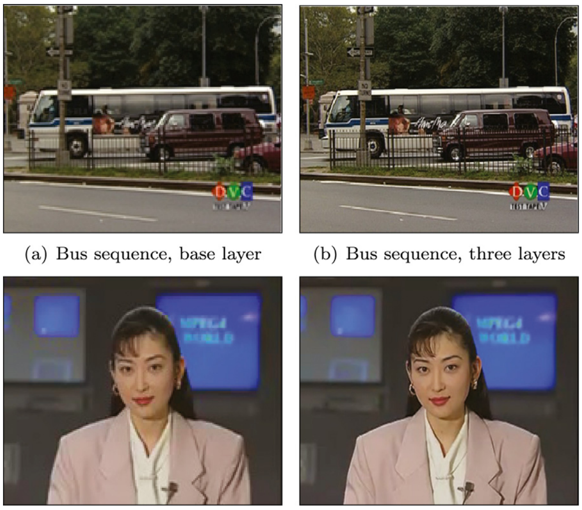
**for** each outcome $y$ **do**
  **for** $i = 1$ to $m$ **do**
    **for** $j = 1$ to $n$ **do**
      $dec(i, j) = 1$
  **for** $a = 1$ to $m$ **do**
    **for** $b = 1$ to $n$ **do**
      **for** $h = 1$ to $a$, $k = 1$ to $b$ **do**
        **if** $\sum_{i=h}^{a} \sum_{j=k}^{b} y_{i,j} < (a-h+1)(b-k+1)$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

Fig. 8 shows a part of a three-dimensional reference table, which is created for a given transmission strategy $x$. The figure depicts the successful decoding probability of each layer in the case of a delivery rate equal to 0.95. In this figure, we represent the successful decoding probability of layer $L_{i,j}$ for delivery rate $p$ as $q_{i,j,p}$. For each delivery rate, there will be a matrix similar to that in Fig. 8.

### 5.3. Extension to varying number of packets per layer

In the previous sections and the proposed algorithm, 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. Remark 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 Remark 2 and use it in the algorithm. Considering $r_{i,j}$ packets in layer $l_{i,j}$, the decoding condition in Remark 2 becomes:

$$\sum_{i=h}^{a} \sum_{j=k}^{b} y_{i,j} \geq \sum_{i=h}^{a} \sum_{j=k}^{b} r_{i,j}$$
$$\forall\, h, k : 1 \leq h \leq a, 1 \leq k \leq b \tag{4}$$



(a) Bus sequence, base layer (b) Bus sequence, three layers

(c) Akiyo sequence, base layer (d) Akiyo sequence, three layers

**Fig. 9.** Decoded video frames from Bus [30] and Akiyo [31] video sequences. (a) and (c) Decoded video frames using the base spatio layer. (b) and (d) Decoded video frames using all of the 3 spatio layers.

## 6. Evaluation

In this section we evaluate our proposed methods by comparing them with the Percy method [19]. In Percy, the triangular coding is performed among the spatio layers (in [19], quality layers are mentioned; however, Percy can be applied on spatio layers instead on the quality layers). Therefore, 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. Different from their work, we consider spatio and temporal layers. In order to evaluate the methods, we implemented a simulator in the MATLAB environment. Moreover, we use JSVM reference software [29] for encoding and decoding videos and measuring the PSNR of the decoded videos.

### 6.1. Simulation setting

We use the Bus [30] and Akiyo [31] video sequences in our evaluations, which are shown in Fig. 9. The resolution and the frame rate of the videos that we use are equal to 352 × 288 pixels and 30 frames per second, respectively. We partition the videos 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. Fig. 9(a) and (b) depict the decoded videos frames using the base layer and the 3 spatio layers of the Bus video sequence.

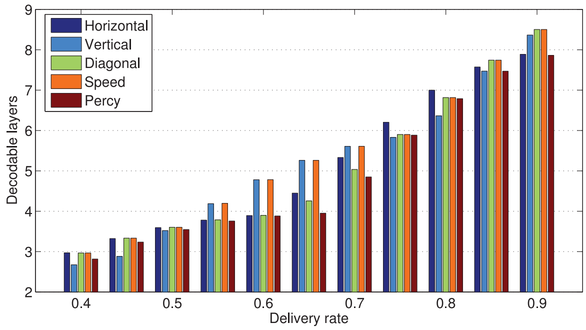



**Fig. 10.** Number of decodable layers in the case of single user for any video. $m = 4,\ n = 3,\ X = 12,\ CS = 1$.

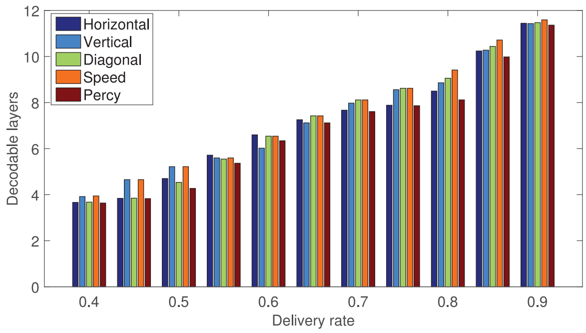



**Fig. 11.** Number of decodable layers in the case of single user for any video. $m = 4,\ n = 3,\ X = 15,\ CS = 1$.

**Table 2**
PSNR of the decoded layers of the Bus video sequence [30].

| 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 |

**Table 3**
PSNR of the decoded layers of the Akiyo video sequence [31].

| n/m | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 35.58 | 35.96 | 36.17 | 38.74 |
| 2 | 37.83 | 38.78 | 39.49 | 45.24 |
| 3 | 39.24 | 40.85 | 42.38 | 99.99 |

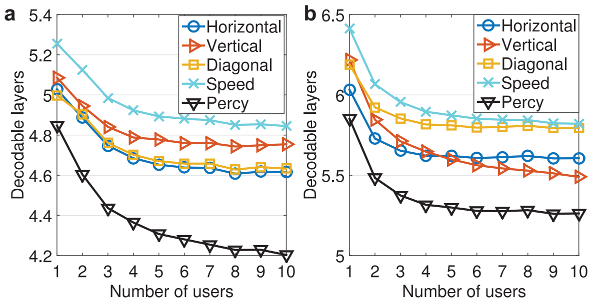



**Fig. 12.** Number of decodable layers in the case of multiple users for any video. $m = 4,\ n = 3$. (a) $X = 12$. (b) $X = 14$.

We evaluate the methods in terms of number of decodable layers and PSNR. We calculate the PSNR of the decoded videos using the PSNR Static 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 of the Bus and Akiyo video sequence are shown in Tables 2 and 3, respectively.

In the simulations, we set $CS = (1, 1, 1)$ for the speed coding method. We run the simulations for 1000 random delivery rates, and show the average output in the plots.

### *6.2. Simulation results*

#### *6.2.1. Number of decodable layers*

In the first experiment, we measure the number of decodable layers in the case of single user in Fig. 10. 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 of the vertical approach. Fig. 10 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. In this plot, the total number of transmissions is equal to the number of packets in the video. The idea behind selecting this low number of transmissions is to measure the quality of the received videos in the case of limited bandwidth.

We repeat the previous experiment in the case of 15 packet transmissions. The result is depicted in Fig. 11. The figure shows that in the case of a delivery rate equal to 0.9, the number of received videos in different methods becomes very close to each other. The reason is the high delivery rate and high redundancy level. However, for a moderate delivery rate, the number of received layers using our methods is still more than that of the Percy method.

Fig. 12(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]. In other words, the reliability is distributed uniformly in the specified range. 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 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 speed coding method is up to 14% more than that of the Percy method.

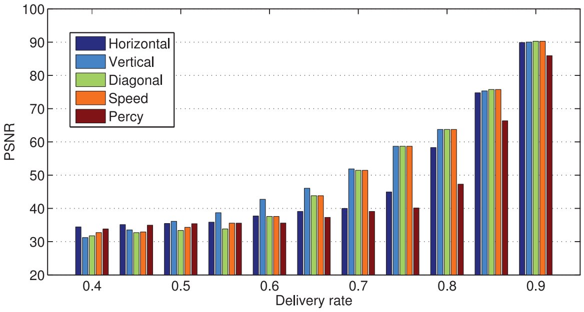



**Fig. 13.** Bus video sequence [30]. PSNR of the decoded video in the case of single user. $m = 4,\ n = 3,\ X = 20,\ CS = 1$.

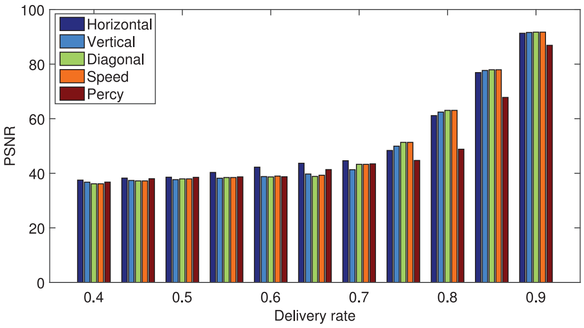



**Fig. 14.** Akiyo video sequence [31]. PSNR of the decoded video in the case of single user. $m = 4,\ n = 3,\ X = 20,\ CS = 1$.

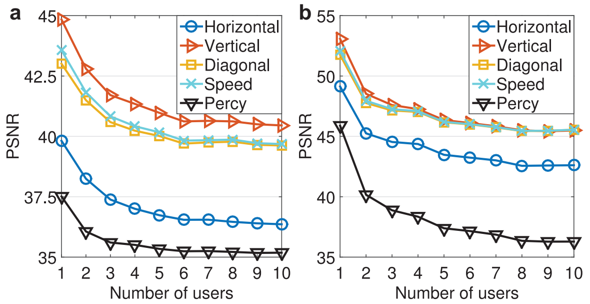



**Fig. 15.** Bus video sequence [30]. PSNR of the decoded video in the case of multiple users. Bus video sequence. $m = 4,\ n = 3$. (a) $X = 18$. (b) $X = 20$.

We increase the total number of transmissions to 14 and repeat the previous experiment. The result is shown in Fig. 12(b). It is clear that a higher number of transmissions results in more decodable layers. The figure illustrates that the speed coding scheme and the Percy method has the highest and lowest number of decodable layers, respectively.

*6.2.2. 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. 13, the PSNR of the decoded videos in the case of single user are shown for different delivery rates. The Bus video sequence contains 18 packets, and the number of transmitted packets is equal to 20. The figure shows that the PSNR of the decoded video increases as the delivery rate of the link increases. Fig. 13 depicts that the PSNR of the vertical, diagonal, and the speed coding methods are up to 45% more than that of the Percy method.

In Fig. 14, we repeat our last experiment on Akiyo video sequence. The figure shows that our proposed methods can deliver a higher video quality than the Percy method in terms of PSNR. By comparing Figs. 13 and 14, we can say that our methods are more efficient in the case of Bus video sequence than the Akiyo sequence. That is because of the different video characteristic. The Bus video sequence is more dynamic than the Akiyo method. As a result, losing some temporal or quality layers has a less impact on the total PSNR of the decoded video. In other words, the Akiyo video is naturally more robust against losses than the Bus video trace. As a result, even with an inappropriate protection method, the quality of the decoded video is higher than that of the Bus video.

Fig. 15(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 figure 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. 15(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 2 shows, in the Bus video sequence, 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.

We repeat the previous experiment on Akiyo video sequence. As Fig. 16(a) shows, our proposed horizontal triangular coding results in the highest PSNR compared to the other methods. This is in contrast with the case of Bus video trace, in which the performance of horizontal triangular coding was less that of the vertical, diagonal, and speed coding methods (Fig. 15(a)). This can be justified by the difference in the characteristics of the Bus and Akiyo video sequences. As the number of users increase in Fig. 16(a), the average PSNR decreases. However, the slop of this average PSNR loss decrease as we the number of users becomes more.

In Fig. 16(b), we change the number of transmission from 18 to 20. in this figure, the performance of the horizontal coding method is less than that of the Diagonal as speed

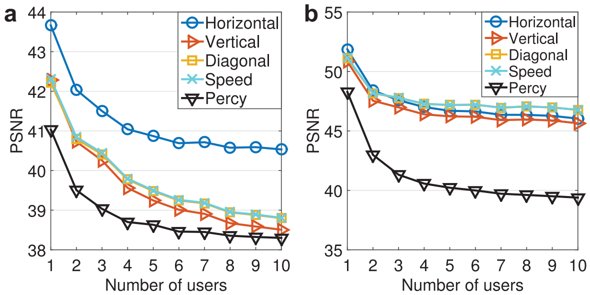



**Fig. 16.** Akiyo video sequence [31]. PSNR of the decoded video in the case of multiple users. $m = 4$, $n = 3$. (a) $X = 18$. (b) $X = 20$.

coding methods. From Fig. 16(a) and (b) we find that a coding method which is appropriate for a specific number of transmissions might not be efficient for the other number of transmissions.

### 6.3. Discussion and summary

The presented simulation results suggest that the number of decodable layers in our proposed coding schemes are almost always more than that of the Percy method, which is due to more coding possibilities in our schemes. In the case of single destination, depending on the delivery rate of the link, different coding methods might be optimal. However, in general, the number of decodable layers of the speed coding method is more than that of the other methods. Also, the Percy method has the lowest number of decodable layers and PSNR among the compared methods.

Comparing the number of decodable layers and the PSNR values of the different methods, we can find that the relation between the number of decodable layers and PSNR is not linear. For instance, in the case of multiple users, the speed method has the largest number of decodable layers among the evaluated methods. However, for Bust video sequence, its PSNR is less than that of the vertical coding method.

## 7. Conclusion

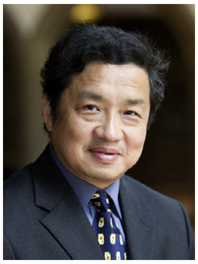
As a result of the rapid increase in the popularity of wireless devices, such as smartphones and tablets, and video streaming over Internet, wireless video multicasting is becoming an important application. However, the diversity of users' channel conditions is a challenge to video multicasting to multiple receivers. Multi-resolution video coding, in which a video is divided into a set of base layer and enhancement layers, is an efficient method to address this challenge. In the previous works, it has been shown that a triangular network coding scheme can increase the quality of the received videos by the users. In this work, we propose a two-dimensional network coding method to further 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 network coding scheme. Our future work is to analyze the effect of spatio and temporal layers on the video quality, and to find a mechanism to select the appropriate coding strategy based on the characteristics of the videos.

## References


[1] P. Ostovari, J. Wu, Robust wireless delivery of scalable videos using inter-layer network coding, in: Proceedings of IEEE WCNC, 2015, pp. 2191–2196.
[2] A. Finamore, M. Mellia, M. Munafò, R. Torres, S. Rao, Youtube everywhere: impact of device and infrastructure synergies on user experience, in: Proceedings of ACM IMC, 2011, pp. 345–360.
[3] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, F. Jahanian, Internet inter-domain traffic, in: Proceedings of ACM SIGCOMM, 2010, pp. 75–86.
[4] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019 White Paper, 2014.
[5] S. McCanne, V. Jacobson, M. Vetterli, Receiver-driven layered multicast, in: Proceedings of ACM CCR, 1996, pp. 117–130.
[6] M. Kim, D. Lucani, X. Shi, F. Zhao, M. Médard, Network coding for multi-resolution multicast, in: IEEE INFOCOM, 2010, pp. 1–9.
[7] M. Effros, Universal multiresolution source codes, IEEE Trans. Inf. Theory 47 (6) (2001) 2113–2129.
[8] M. Shao, S. Dumitrescu, X. Wu, Layered multicast with inter-layer network coding for multimedia streaming, IEEE Trans. Multimed. 13 (99) (2011) 353–365.
[9] E. Magli, M. Wang, P. Frossard, A. Markopoulou, Network coding meets multimedia: a review, IEEE Trans. Multim. 15 (5) (2013) 1195–1212.
[10] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the h. 264/AVC standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (2007a) 1103–1120.
[11] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the h. 264/avc standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (2007b) 1103–1120.
[12] H. Wang, S. Xiao, C. Kuo, Robust video multicast with joint network coding and video interleaving, J. Vis. Commun. Image Represent. 21 (2) (2010) 77–88.
[13] H. Wang, S. Xiao, C. Kuo, Random linear network coding with ladder-shaped global coding matrix for robust video transmission, J. Vis. Commun. Image Represent. 22 (3) (2011) 203–212.
[14] H. Ha, C. Yim, Layer-weighted unequal error protection for scalable video coding extension of h. 264/AVC, IEEE Trans. Consum. Electron. 54 (2) (2008) 736–744.
[15] Y. Sun, X. Zhang, F. Tang, S. Fowler, H. Cui, X. Dong, Layer-aware unequal error protection for scalable h. 264 video robust transmission over packet lossy networks, in: NBiS, 2011, pp. 628–633.
[16] A. Tassi, I. Chatzigeorgiou, D. Vukobratovic, Resource-allocation frameworks for network-coded layered multimedia multicast services, IEEE J. Sel. Areas in Commun. 33 (2) (2015a) 141–155.
[17] A. Tassi, C. Khirallah, D. Vukobratovic, F. Chiti, J. Thompson, R. Fantacci, Resource allocation strategies for network-coded video broadcasting services over lte-advanced, IEEE Trans. Veh. Technol. 64 (5) (2015b) 2186–2192.
[18] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, Z. Xiong, Resource allocation strategies for network-coded video broadcasting services over lte-advanced, IEEE Trans. Veh. Technol 64 (5) (2009) 2186–2192.
[19] 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.
[20] D. Wang, J. Liu, Q. Zhang, F. Chen, Cooperative coding and caching for streaming data in multihop wireless networks, EURASIP J. Wireless Commun. Netw. 2010 (10) (2010) 2:1–2:13.
[21] P. Ostovari, J. Wu, A. Khreishah, N.B. Shroff, Scalable video streaming with helper nodes using random linear network coding, IEEE/ACM Trans. Netw. PP (99) (2015) 1–14.
[22] R. Ahlswede, N. Cai, S. Li, R. Yeung, Network information flow, IEEE Trans. Inform. Theory 46 (4) (2000) 1204–1216.
[23] R. Koetter, M. Medard, An algebraic approach to network coding, IEEE/ACM Trans. Netw. 11 (5) (2003) 782–795.
[24] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, IEEE Trans. Inform. Theory 52 (10) (2006) 4413–4430.
[25] Y. Wang, T. Fang, L-P Chau, K. Yap, Two-dimensional channel coding scheme for MCTF-based scalable video coding, IEEE Trans. Multimed. 9 (1) (2007) 37–45.

[26] F. Soldo, A. Markopoulou, A.L. Toledo, On the performance of network coding in multi-resolution wireless video streaming, in: IEEE NetCod, 2010, pp. 1–6.
[27] M. Halloush, H. Radha, Practical network coding for scalable video in error prone networks, in: IEEE PCS, 2009, pp. 2107–2114.
[28] T. Ho, R. Koetter, M. Medard, D. Karger, M. Effros, The benefits of coding over routing in a randomized setting, in: IEEE ISIT, 2003.
[29] J. Reichel, H. Schwarz, M. Wien, Joint Scalable Video Model JSVM-6, Technical Report, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Geneva, Switzerland, 2006.
[30] Video sequences, http://trace.eas.asu.edu/yuv/bus/
[31] Video sequences, http://trace.eas.asu.edu/yuv/akiyo/

**Pouya Ostovari** is s an assistant professor in the Department of Computer & Information Sciences at Temple University. He received his B.S. degree in computer engineering from Shahid Beheshti Univeristy, and the M.S. degree from Amirkabir University of Technology (Tehran Polytechnic). Also, he received his Ph.D. degree in computer & information sciences from Temple University in 2015. His research interests include wireless networks, sensor networks, and distributed systems.



**Jie Wu** is the Associate Vice Provost for International Affairs at Temple University. He also serves as the Chair and Laura H. Carnell professor in the Department of Computer and Information Sciences. Prior to joining Tempe University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.