## RESEARCH

**Open Access**

# Multi-user video streaming using unequal error protection network coding in wireless networks

Dejan Vukobratović[1*] and Vladimir Stanković[2]

## Abstract

In this article, we investigate a multi-user video streaming system applying unequal error protection (UEP) network coding (NC) for simultaneous real-time exchange of scalable video streams among multiple users. We focus on a simple wireless scenario where users exchange encoded data packets over a common central network node (e.g., a base station or an access point) that aims to capture the fundamental system behaviour. Our goal is to present analytical tools that provide both the decoding probability analysis and the expected delay guarantees for different importance layers of scalable video streams. Using the proposed tools, we offer a simple framework for design and analysis of UEP NC based multi-user video streaming systems and provide examples of system design for video conferencing scenario in broadband wireless cellular networks.

## Introduction

Real-time multi-user (or multi-party) video streaming refers to a scenario where multiple users, interconnected by a common communication network, perform real-time exchange of video streams [1,2]. Each of the users continuously creates its own video stream and is interested in the continuous and real-time recovery of the streams generated by a subset or the set of all the other users. Application examples include video conferencing, multi-view video systems, multi-party peer-to-peer (P2P) video exchange, emerging multimedia-oriented social networking (e.g., "see-what-i-see" applications), etc. However, designing robust and efficient multi-user video streaming systems over wireless networks faces a number of challenges, most notably, the strict delay limits enforced by real-time requirements and time-variable wireless channel conditions responsible for frequent packet losses.

Network coding (NC) is a novel information processing technique applied in network nodes in which, instead of simple forwarding of received data packets, the data packets are combined and resulting network coded packets are transmitted instead. The idea was first introduced for the single-source multicast problem, where it was shown that, unlike routing, it achieves the capacity of the multicast connection [3]. For the single-source multicast problems represented by directed acyclic graphs with unit-capacity error-free edges, the class of linear network codes achieves the multicast connection capacity [4]. Furthermore, random linear codes over sufficiently large finite fields open the way for simple and fully distributed network code design [5]. The random linear coding (RLC) approach is adapted for practical implementation in lossy packet networks [6,7], and suggested in a number of wireless networking applications [8,9].

To increase throughput and improve error resilience, NC has been recently suggested for applications in multimedia streaming [10-14], and in particular, for multi-user video conferencing [15-17]. In [15], which is closest to our work, RLC is investigated for multi-party video conferencing in wireless broadband cellular systems. This study demonstrates that RLC applied within the central node possess a potential to reduce the end-to-end delay, increase throughput and improve the transmission reliability and system fairness.

In this article, we explore analytical tools for the design and analysis of a real-time multi-user video streaming system that applies scalable video coding and unequal error protection (UEP) RLC. We focus on a simple scenario where wireless users exchange video streams over a common central node with the goal of capturing

*Correspondence: dejanv@uns.ac.rs
[1] Department of Power, Electronics and Communication Engineering, University of Novi Sad, Trg D. Obradovića 6, Novi Sad, Serbia
Full list of author information is available at the end of the article

the fundamental system behaviour. This work builds upon our recent theoretical analysis of UEP RLC schemes for erasure channels [18]. Addressing a specific UEP RLC application, the real-time multi-user video streaming, this article extends the layer decoding probability analysis of UEP RLC addressed in previous work to include additional performance measures such as expected decoding delays of different video layers and evolution of the expected received video quality of exchanged scalable video streams over time. Using the presented set of analytical tools, we offer a simple framework for the design and analysis of UEP RLC based real-time multi-user scalable video streaming systems. The framework provides flexible approach for reliable exchange of layered video streams over dynamically changing wireless channels. The application of the proposed framework and the benefits over the standard RLC approach applied in [15] are demonstrated through the distortion-optimized system design examples.

The article is organized as follows. Section "RLC: an overview and UEP extension" provides a background on RLC and its UEP extensions, and provides a decoding performance analysis of the UEP RLC. The proposed multi-user video streaming setup is introduced in Section "Multi-user video streaming using UEP NC". The same section formulates the distortion-based optimization of the multi-user video streaming system based on UEP NC. Selected UEP NC code design examples are discussed in Section "System optimization and results". The article is concluded in Section "Conclusions".

## RLC: an overview and UEP extension
### Background and motivation
For wireless broadcasting, NC is usually motivated by the two-user packet exchange example in Figure 1 (see e.g., [19]). Instead of replicating and independently transmitting each user packet, the central node XORs the incoming user packets and broadcasts a single coded packet. As a result, the number of packet transmissions required for two-way packet exchange between users reduces from four to three.

The two-user example can be extended to multi-user scenarios using opportunistic binary NC (XOR-ing) for
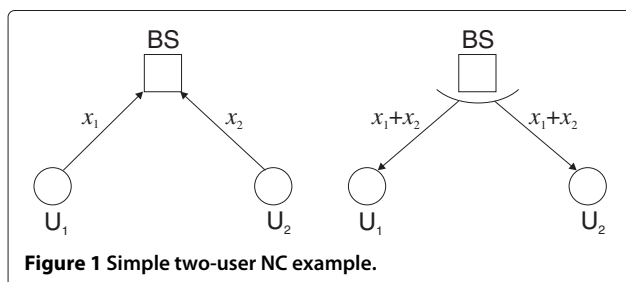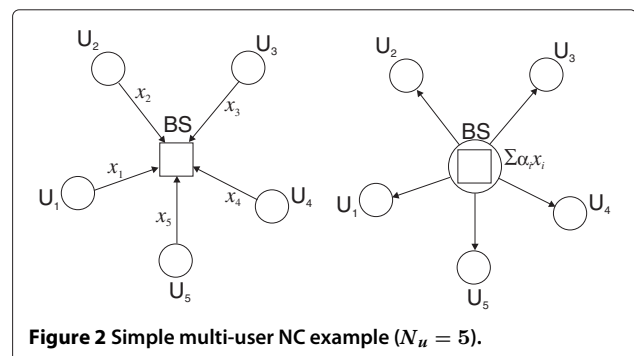
wireless broadcast networks, proposed in [20]. However, the broadcasting node needs to know the buffer content of its neighbours in order to construct the optimal encoded packet. On the other hand, extension to multi-user scenario is possible by applying RLC over received data packets using non-binary finite field coefficents [15]. After $N_u$ users upload their data to the central node, the central node broadcasts random linear combinations of users' packets in a "rateless" fashion, until each user recovers the data packets of other users (Figure 2). A user needs to receive any $N_u - 1$ encoded packets broadcasted by the central node in order to recover other users with high probability, if the finite field size is sufficiently large. In contrast, without NC, an alternative is repeated broadcasting of user packets in a "data carousel" fashion, or managing the one-to-many automatic repeat request (ARQ) mechanism, which is known to suffer a number of drawbacks (e.g., excessive delay and the feedback implosion problem).

In this article, we extend the basic idea of Figure 2 to the case where users exchange scalable coded video streams. Users' messages are organized into layers of different importance, starting with the most important and continuing with progressively less important layers. If, e.g., due to poor channel conditions or low access data rates, a user is unable to fully recover other users, it benefits from recovering as many importance layers of other users' messages starting from the most important layer onwards. For increased protection of more important layers over error-prone wireless links, the advantages of the UEP forward error correction (FEC) coding are demonstrated in a number of research studies [21-23]. In this article, we focus on the UEP RLC to explore the benefits of both rateless UEP FEC coding and NC.

### Random linear coding
Let $x = \{x_1, x_2, \ldots, x_K\}$ be a source message that consists of $K$ equal-length source packets. RLC applied over the message $x$ produces encoded packets $\omega$ as random linear combinations of source packets with coefficients randomly selected from a given finite field $GF(2^q)$: $\omega =$



**Figure 1 Simple two-user NC example.**



**Figure 2 Simple multi-user NC example ($N_u = 5$).**

$\sum_{i=1}^{K} \alpha_i \cdot x_i$, where $\alpha_i$ is a randomly selected element of $GF(2^q)$, and $\omega$ is of the same length as source packets. Each encoded packet carries a header information containing global encoding coefficients within the global encoding vector $\boldsymbol{g} = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ [6]. For unicast transmission, to relax the overhead requirements, both the transmitter and the receiver may use the pair of synchronized random number generators (RNG) to produce the same sequence of global encoding coefficients. In this case, only a short RNG seed needs to be conveyed within the packet overhead. The RLC encoding can be repeated at the transmitter in a rateless fashion, until the receiver collects enough encoded packets to decode the source message using the Gaussian elimination (GE) decoding. GE decoding introduces complexity limitations on the message length $K$. However, for real-time interactive multimedia applications, small values of $K$ are acceptable for practical deployment [24,25].
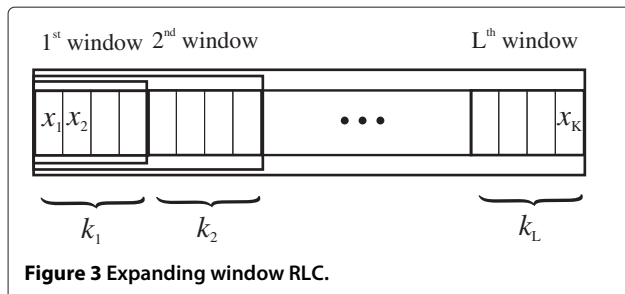
### UEP random linear coding

Let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_K\}$ be a *layered* source message containing $K$ equal-length source packets classified into $L$ importance layers. The source message starts with the most important base layer (BL) and continues with progressively less important enhancement layers (EL). The subset of the source message containing the $l$-th layer is denoted as $\boldsymbol{x}_l$ and contains $k_l$ source packets, where $\sum_{i=1}^{L} k_i = K$. We denote the subset of the source message containing the first $l$ layers as $\boldsymbol{x}_{1:l}$ and the number of source packets in the first $l$ layers is $K_l = \sum_{i=1}^{l} k_i$ (Figure 3).

The UEP RLC scheme called expanding window (EW) RLC was investigated in [18]. The EW RLC introduces a set of EWs over the layered source message following the importance structure of the message. More precisely, for the $L$-layer message, the set of $L$ EWs is defined where the $l$-th EW, $1 \leq l \leq L$, contains the source block subset $\boldsymbol{x}_{1:l}$ (Figure 3).

#### EW RLC encoding

The EW RLC introduces *the probabilistic encoding process* over the set of EWs. For each encoded packet, one of the EWs is first selected using the predefined window selection distribution $\Gamma(\xi) = \sum_{i=1}^{L} \Gamma_i \xi^i$, where $\Gamma_i$ is the probability of selection of the $i$-th window, $\sum_{i=1}^{L} \Gamma_i = 1$. Then, an encoded packet is created by applying RLC only over the selected window. Arbitrarily many EW RLC encoded packets can be produced by independently repeating the encoding process for each encoded packet.

#### EW RLC decoding

A receiver collects correctly received EW RLC coded packets and decodes the source message (or subset of its layers) using the standard GE decoder, as if standard RLC is used. An important difference is that the parts of the layered source message could be decoded even if less than $K$ encoded packets are received.

For more details on the design of EW RLC, we refer the interested reader to [18].

### Performance analysis of EW RLC

In the following, we review a simple upper bound for the set of decoding probabilities $P_{d,l}(N)$ that the content of the $l$-th window, $1 \leq l \leq L$, is recovered at the receiver after receiving any $N$ EW RLC encoded packets. The upper bound is general and holds for any packet-level coding and decoding scheme that applies probabilistic encoding and expanding windowing approach. Moreover, the EW RLC in combination with GE decoding achieves this bound as the field size increases [18].

Let $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$ be a sequence of $N$ received EW RLC encoded packets. For the derivation of upper bounds, $\boldsymbol{y}$ is completely described by the corresponding vector $\boldsymbol{n} = \{n_1, n_2, \ldots, n_L\}$, where $n_l$ denotes the number of received packets obtained by EW RLC coding over the $l$-th window. We denote by $\boldsymbol{y}_l$ (and $\boldsymbol{y}_{1:l}$, respectively) the subset of $\boldsymbol{y}$ containing the set of $n_l$ $\left(N_l = \sum_{i=1}^{l} n_i\right)$ received packets obtained by EW RLC encoding over the $l$-th (the first $l$) window(s).

For a given $\boldsymbol{n}$, we define a set of variables $R_l(\boldsymbol{n}), 1 \leq l \leq L$, using the following recursion:

$$
\begin{aligned}
R_1(\boldsymbol{n}) &= \min(n_1, K_1), \\
R_l(\boldsymbol{n}) &= \min(R_{l-1}(\boldsymbol{n}) + n_l, K_l), 2 \leq l \leq L. \quad (1)
\end{aligned}
$$

Thus any received $\boldsymbol{y}$ can be recursively transformed into $\boldsymbol{R} = \{R_1(\boldsymbol{n}), R_2(\boldsymbol{n}), \ldots, R_L(\boldsymbol{n})\}$.

The values of $R_l(\boldsymbol{n})$ provide an upper bound on the rank of the $N_l \times K$ matrices whose rows are global encoding vectors of $N_l$ received packets in $\boldsymbol{y}_{1:l}$. In other words, $R_l(\boldsymbol{n})$ is the maximum number of source packets in $\boldsymbol{x}_{1:l}$ that can be recovered from $\boldsymbol{y}_{1:l}$. Using $\boldsymbol{R}$ we can simply determine the set of layers of $\boldsymbol{x}$ the receiver can recover after receiving $\boldsymbol{y}$. Namely, $\boldsymbol{x}_{1:l}$ can be fully recovered if $R_l(\boldsymbol{n}) = K_l$. In addition, $\boldsymbol{x}_{1:l}$ can be also recovered if any of the larger



**Figure 3** Expanding window RLC.

windows is recovered, i.e., if $R_m(\boldsymbol{n}) = K_m$ for any $l < m \leq L$, because larger windows contain smaller ones.

Formally, the upper bound on $P_{d,l}(N)$ follows by conditioning on $\boldsymbol{n}$ [a]:

$$P_{d,l}(N|\boldsymbol{n}) =$$
$$= I\left((R_l = K_l) \bigvee_{i=l+1}^{L} \left(\bigwedge_{j=l}^{i-1}(R_j < K_j) \wedge (R_i = K_i)\right)\right),$$
(2)

where $I(\cdot)$ represents an indicator function equal to 1 if its argument, which is a logical expression, is true, otherwise, $I(\cdot) = 0$, and $\bigvee_i()$ ($\bigwedge_j()$) is logical or (and) of a sequence of logical expressions parametrized by $i$ ($j$). Conditioning on $\boldsymbol{n}$ is removed using the prior distribution over $\boldsymbol{n}$:

$$P_{\Gamma(\xi),N}(\boldsymbol{n}) = \frac{N!}{n_1! n_2! \ldots n_L!} \Gamma_1^{n_1} \Gamma_2^{n_2} \ldots \Gamma_L^{n_L}.$$
(3)

Finally, upper bound on $P_{d,l}(N)$ is obtained as:

$$P_{d,l}(N) \leq \sum_{\substack{(n_1,n_2,\ldots,n_L): \\ \sum_{i=1}^{L} n_i = N}} P_{\Gamma(\xi),N}(\boldsymbol{n}) P_{d,l}(N|\boldsymbol{n}).$$
(4)

For a given layered source message, $P_{d,l}(N)$ depends only on the selected window selection distribution $\Gamma(\xi)$.[b] Therefore, designing EW RLC codes with desired $P_{d,l}(N)$ behaviour reduces to the design of appropriate $\Gamma(\xi)$ [18].

## Multi-user video streaming using UEP NC
### System model
In this article, we propose the UEP NC as a core component of a real-time multi-user video streaming system. For simplicity, we focus on a wireless cellular network example where $N_u$ mobile users ($U_1, U_2, \ldots, U_{N_u}$) participate in a multi-party video conferencing session over a common base station (BS) within a single cell (Figure 4).

The presented results are applicable in similar scenarios, e.g., if instead of a single BS users connect to different BSs mutually interconnected by high-speed links (e.g., fiber optic) or if instead of a cellular network we observe a Wi-Fi access point.

In our scenario, every user continuously segments its own video stream into groups of frames (GOF), where each GOF contains $N_{\text{gof}}$ frames, and compresses every GOF using a scalable video coder. For each user $U_i$ and each compressed GOF, the output of the video coder is a layered source message $\boldsymbol{x}^{(i)}$ that contains $K^{(i)}$ source packets, each of length $b$ bits, organized into $L$ layers, where the $l$-th layer contains $k_l^{(i)}$ packets. The values of $b$ and $L$ are the same across all users whereas for each user $U_i$ and each GOF, the values $K^{(i)}$ and $\{k_1^{(i)}, k_2^{(i)}, \ldots, k_L^{(i)}\}$ are in general different.

Video streaming among users in a session may be observed as a GOF-by-GOF exchange process. A single GOF exchange period repeats every $T_{\text{gof}} = N_{\text{gof}}/N_{\text{fps}}$ seconds, where $N_{\text{fps}}$ is the number of frames per second of the video stream, and both $N_{\text{gof}}$ and $N_{\text{fps}}$ are equal across all users. For simplicity, we assume that GOF periods are aligned among different users, i.e., the messages $\boldsymbol{x}^{(i)}$ are synchronously generated by all the users. For every GOF period, the goal of each user is to share its own and collect other users GOFs, or at least as many of their layers starting from the beginning onwards, within a strict delay limits.

During each GOF period, the data exchange process can be divided into two phases. In the first, *upload phase*, users simultaneously upload their data to the BS, and in the second, *broadcast phase*, the BS broadcasts the received data to all the users. We assume that orthogonal channels are allocated between each user and the BS, and for the broadcast transmission by the BS, allowing for simultaneous transmission on all channels. Each wireless link is modeled as a synchronous time-slotted packet-erasure link where the fixed size encoded packets of length $b$ bits are transmitted using a fixed transmission data rate $R$ and erasure probability $\epsilon$. We assume
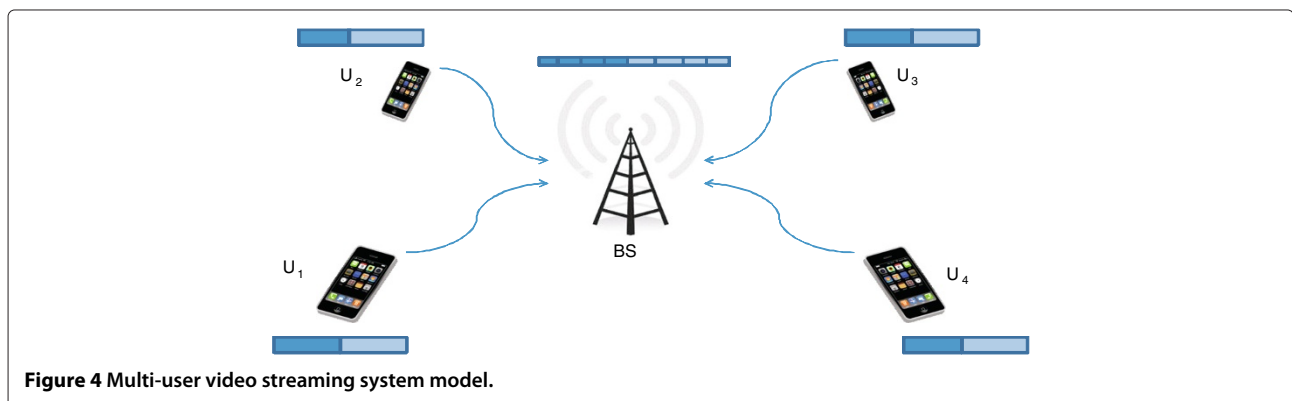


**Figure 4 Multi-user video streaming system model.**

that the pair $(R, \epsilon)$ is in general different on different wireless links, it remains fixed during the transmission period $T_{\text{gof}}$ of a single GOF, and may change between different GOF transmissions. The time slot duration $T_p = b/R$ represents a time required for a single encoded packet transmission.[c]

In the upload phase, to protect data from erasures, the user $U_i$ encodes the source message $x^{(i)}$ using the EW RLC scheme defined by a window selection distribution $\Gamma^{(i)}(\xi)$ and streams the encoded packets using the user rate $R^{(i)}$. The BS recovers the users' messages using $N_u$ independent GE decoders dedicated to different users. After decoding as many user layers as possible, the BS creates its own source message $x^{(\text{BS})}$ that contains all or a subset of users' message layers. For example, if all $N_u$ users are completely recovered, the message $x^{(\text{BS})}$ is of length $K^{(\text{BS})} = \sum_{i=1}^{N_u} K^{(i)}$ packets and contains $L$ layers. The $l$-th layer $x_l^{(\text{BS})}$ comprises the total of $k_l^{(\text{BS})} = \sum_{i=1}^{N_u} k_l^{(i)}$ packets from the $l$-th layer data of all $N_u$ user messages, $x_l^{(\text{BS})} = \{x_l^{(1)}, x_l^{(2)}, \ldots, x_l^{(N_u)}\}$, as illustrated in Figure 4 for the two-layer scenario. In general, the BS may create the $l$-th layer $x_l^{(\text{BS})}$ from the $l$-th layers of a subset of users.

In the broadcast phase, the BS applies the EW RLC scheme defined by a window selection distribution $\Gamma^{(\text{BS})}(\xi)$ over $x^{(\text{BS})}$ and broadcasts the stream of encoded packets using the broadcast rate $R^{(\text{BS})}$. Each user recovers the BS message $x^{(\text{BS})}$ using the GE decoder where, prior to decoding, the user cancels out its own packets from the received encoded packets. We note that the two phases may overlap in time, i.e., the BS may start broadcasting a subset of (already recovered) layers of $x^{(\text{BS})}$ before the upload phase of all users is completed.

### Single-link analysis: decoding and delay performance

To analyze the system in Figure 4, we focus on a single-link transmission of UEP RLC coded layered message between any transmitter-receiver pair during a fixed time period $T$. Instead of layer decoding probabilities after fixed number of received packets, $P_{d,l}(N)$ (Section "Performance analysis of EW RLC"), we shift our interest to layer decoding probabilities after fixed transmission time, $P_{d,l}(T)$. However, unlike $P_{d,l}(N)$, deriving $P_{d,l}(T)$ requires introduction of a packet-level channel model.

During the period of duration $T$, the transmission process consists of $N_T = \lfloor RT/b \rfloor$ encoded packet transmissions (packet slots). We redefine the received sequence $y = \{y_1, y_2, \ldots, y_{N_T}\}$ to describe the outcome of all $N_T$ transmissions, where $y_i$ may represent either the received encoded packet or a lost (erased) packet. The received sequence $y$ can be described by vector $n = \{n_1, n_2, \ldots, n_L, n_e\}$, where, as before, $n_i$ is the number of received encoded packets obtained by encoding over the

$i$-th EW, $N = \sum_{i=1}^{L} n_i \leq N_T$ is the number of (correctly) received encoded packets during the interval $T$, and $n_e = N_T - N$ is the number of erased packets.

For a fixed $T$, the number of received encoded packets $N$ is dependent on the underlying channel packet-loss model. For simplicity, we assume a packet erasure channel model that erases encoded packets independently with erasure probability $\epsilon$. To obtain $P_{d,l}(T)$, we use conditioning over $n$:

$$P_{d,l}(T) = \sum_{n} P_{\Gamma(\xi),\epsilon}(n) P_{d,l}(T|n), \tag{5}$$

where $P_{\Gamma(\xi),\epsilon}(n)$ is slightly altered version of (3) that accounts for $n_e$ erased packet events:

$$P_{\Gamma(x),\epsilon}(n) = \frac{N_T!}{n_1! n_2! \ldots n_L! n_e!} \cdot \tag{6}$$
$$\cdot [\Gamma_1(1-\epsilon)]^{n_1} [\Gamma_2(1-\epsilon)]^{n_2} \ldots [\Gamma_L(1-\epsilon)]^{n_L} \cdot (\epsilon)^{n_e}.$$

$P_{d,l}(T|n)$ can be obtained as the layer decoding probability $P_{d,l}(N|n)$ after $N$ received encoded packets (Section "Performance analysis of EW RLC", Equations (1)–(2)) since, by knowing $n$, we directly obtain $N = \sum_{i=1}^{L} n_i$.

The knowledge of $P_{d,l}(T)$ implicitly provides information on the decoding delay of the $l$-th message layer. For example, one can search for minimal transmission period $T_{\text{th}}^{(l)}$ such that the $l$-th message layer decoding probability $P_{d,l}(T_{\text{th}}^{(l)}) > P_{\text{th}}^{(l)}$, where $P_{\text{th}}^{(l)}$ is the threshold decoding probability set in advance. For more explicit delay information, one can obtain the probability distribution $p_{d,l}(N_T)$ that the $l$-th message layer is recovered after exactly the $N_T$-th time slot (and cannot be recovered before):

$$p_{d,l}(N_T) = \tag{7}$$
$$= \prod_{i=1}^{N_T-1} (1 - P_{d,l}(T = i \cdot T_p)) \cdot P_{d,l}(T = N_T \cdot T_p),$$

and $T_p = b/R$. The expected delay $\mathbb{E}_l[N_T]$ for recovery of the $l$-th message layer is:

$$\mathbb{E}_l[N_T] = \sum_{N_T=1}^{\infty} N_T \cdot p_{d,l}(N_T). \tag{8}$$

The expected delay of recovery of the complete source message $\mathbb{E}[N_T]$ for the EW RLC scheme is equal to the recovery delay of the last $L$-th layer: $\mathbb{E}[N_T] = \mathbb{E}_L[N_T]$. $\mathbb{E}_l[N_T]$ is expressed in terms of the number of time slots, but it can be easily converted into absolute time values as $\mathbb{E}_l[T] = \mathbb{E}[N_T] \cdot T_p$.

**Example 1.** *Let $x$ be a layered source message containing $K = 60$ source packets of size $b = 3,200$ bits (400 bytes) divided into $L = 2$ layers: the BL containing $k_1 = 20$ packets and the EL containing $k_2 = 40$ packets. The EW RLC*
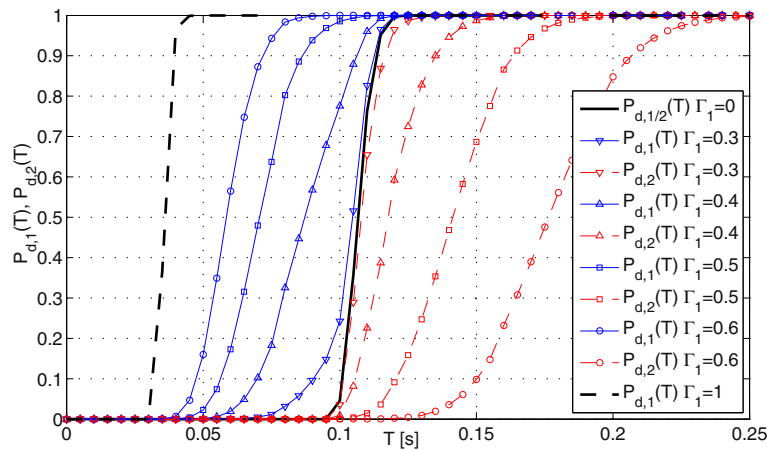
**Figure 5** $P_{d,1}(T)$ and $P_{d,2}(T)$ for EW RLC over the range of $\Gamma_1$ values.

scheme defined by $\Gamma(\xi) = \Gamma_1 \xi + (1 - \Gamma_1)\xi^2$ is applied over $\boldsymbol{x}$ producing continuous stream of 400-bytes long encoded packets. The wireless link towards the receiver is modeled as a synchronous rate $R = 2\,Mbit/s$ link with packet erasure probability $\epsilon = 0.1$. Figure 5 presents the evolution of layer decoding probabilities $P_{d,1}(T)$ and $P_{d,2}(T)$ over time $T$ at the receiver for the range of $\Gamma_1$ values. As a baseline scheme, we start from the middle solid curve for $\Gamma_1 = 0$, representing standard RLC applied over the whole message, which results in simultaneous recovery of both message layers. By increasing $\Gamma_1$, we obtain the UEP effect of the EW RLC scheme, where solid curves for the $P_{d,1}(T)$ gradually shift to the left, i.e., towards earlier recovery of the most important data. The extreme case of $\Gamma_1 = 1$ results in the earliest recovery of the most important part represented by the leftmost dashed curve. The increase in $\Gamma_1$ comes at the price of delayed decoding of less important layer $P_{d,2}(T)$. Figure 6 presents the expected layer decoding

delays $\mathbb{E}_1[T]$ and $\mathbb{E}_2[T]$ as a function of $\Gamma_1$. Note that significant decrease of $\mathbb{E}_1[T]$ with the increase in $\Gamma_1$ initially comes with a relatively small loss in $\mathbb{E}_2[T]$. For example, for $\Gamma_1 = 0.5$, $\mathbb{E}_1[T]$ drops from 105 to 62 ms ($-41\%$) while $\mathbb{E}_2[T]$ rises from 105 to 126 ms ($+20\%$).

The single-link analysis can be extended to the scenario where the transmitter changes the applied EW RLC code during the transmission (i.e., switches between different $\Gamma(\xi)$). As an example, we derive $P_{d,l}(T)$ for $T > T_1$, given that the transmitter has applied the EW RLC defined by $\Gamma^a(\xi)$ during $0 \le t \le T_1$, and the EW RLC defined by $\Gamma^b(\xi)$ for $t > T_1$. For the link parameters $R$ and $\epsilon$, the transmitter sends $N_1 = \lfloor RT_1/b \rfloor$ encoded packets using $\Gamma^a(\xi)$ and $N_2 = \lfloor R(T - T_1)/b \rfloor$ encoded packets using $\Gamma^b(\xi)$. The received sequence $\boldsymbol{y} = [\boldsymbol{y}_1 \boldsymbol{y}_2]$ is a concatenation of two sequences $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ of length $N_1$ and $N_2$, respectively. It can be described by the vector $\boldsymbol{n} = \boldsymbol{n_1} \boxplus \boldsymbol{n_2}$, where the vectors $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ represent the description of
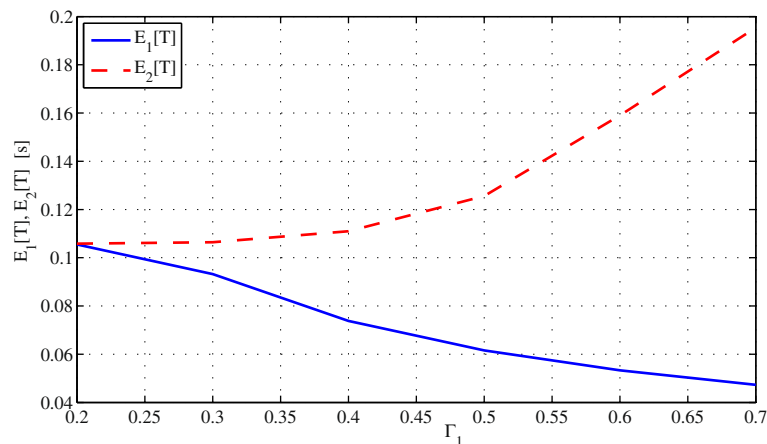


**Figure 6** $E_1[T]$ and $E_2[T]$ for EW RLC over the range of $\Gamma_1$ values.
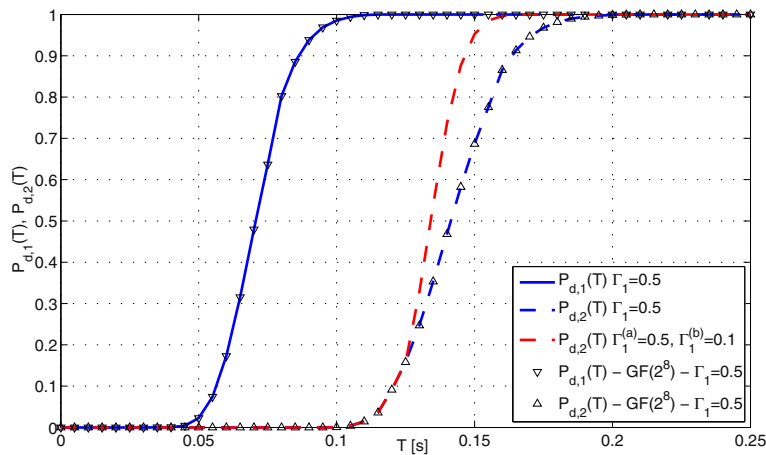
**Figure 7** $P_{d,1}(T)$ and $P_{d,2}(T)$ for EW RLC which changes $\Gamma_1^{(a)} = 0.5$ to $\Gamma_1^{(b)} = 0.1$ at the time instant $T_1 = 0.125$.

$y_1$ and $y_2$ respectively (as defined earlier), and $\boxplus$ is the component-wise sum of two equal-length vectors. Since $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ follow probability distributions $P_{\Gamma^{(a)}(x),\epsilon}(\boldsymbol{n_1})$ and $P_{\Gamma^{(b)}(x),\epsilon}(\boldsymbol{n_2})$ given by Equation (6), the layer decoding probabilities are obtained as in (5):

$$P_{d,l}(T) = \tag{9}$$
$$= \sum_{\boldsymbol{n_1}} \sum_{\boldsymbol{n_2}} P_{\Gamma^{(a)}(x),\epsilon}(\boldsymbol{n_1}) P_{\Gamma^{(b)}(x),\epsilon}(\boldsymbol{n_2}) \cdot P_{d,l}(T|\boldsymbol{n}),$$

where $P_{d,l}(T|\boldsymbol{n})$ follows from $P_{d,l}(N = \sum_{i=1}^{L} n_i|\boldsymbol{n})$.

**Example 2.** *We continue Example 1 by investigating the evolution of $P_{d,1}(T)$ and $P_{d,2}(T)$ over time $T$ at the receiver if the transmitter applies $\Gamma^a(\xi) = 0.5\xi + 0.5\xi^2$ for the first $T_1 = 125\,ms$, and then changes to $\Gamma^b(\xi) = 0.1\xi + 0.9\xi^2$ (the remaining parameters are the same as in the previous example). Figure 7 compares the case where $\Gamma^a(\xi)$ changes to $\Gamma^b(\xi)$ with the case where $\Gamma^a(\xi)$ is used throughout the*

*transmission. Figure illustrates that the $\Gamma(\xi)$ change has no effect on $P_{d,1}(T)$ as it comes too late ($P_{d,1}(T_1 = 0.125) = 1$), whereas the improvement of $P_{d,2}(T)$ for $T > T_1$ is notable due to the increase in the second window selection probability from 0.5 to 0.9, which points out to possible adaptive (e.g., feedback-driven) updates of $\Gamma(\xi)$ during transmission. In addition, Figure 8 demonstrates that the upper bound expressions for $P_{d,l}(T)$ used in this article match very well the exact calculation of $P_{d,l}(T)$ for a finite field size $GF(2^8)$ (markers) [18].*

Finally, analytical results presented above can be easily extended to Gilbert–Elliot erasure channel model with two states: the good and the bad state. This follows from the fact that the probability distribution of the number $n_e$ of erasures over $N_T$ consecutive realizations of the channel (i.e., over time interval $T$) is known (e.g., see [26]). The remaining $N_T - n_e$ non-erased channel realizations deliver encoded symbols from different EWs according to the multinomial distribution law (3).
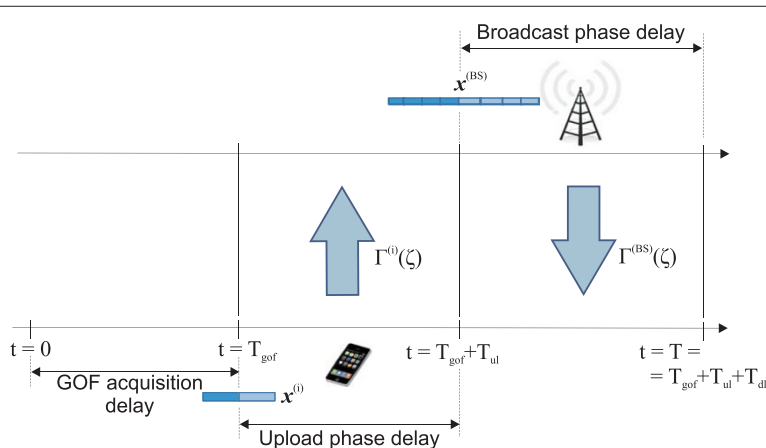


**Figure 8 Time-diagram of different transmission phases in the system model.**

## Distortion-optimal system design

In this section, we apply the single-link anaysis to analyze the multi-user video conferencing setup introduced in Section "System model". Our goal is to formulate the system design problem that leads to the EW RLC code design providing distortion-optimal system performance. As a distortion measure, we use peak signal to noise ratio (PSNR) as a standard video quality metric following directly from the mean square error (MSE) distortion measure. We use the terms video quality and distortion interchangeably while refering to video quality (PSNR) measure.

We focus on a single message (GOF) exchange cycle among the system users. After the initial delay of $t = T_{\text{gof}}$ needed for each user to acquire and compress $N_{\text{gof}}$ frames of video (assuming negligible compression delay), the users start the upload phase. During the upload phase, the BS waits to receive enough encoded packets to recover the users' messages or as many of their layers with sufficiently high probability. The upload phase duration $T_{ul}$ is upper bounded by the GOF period duration $T_{\text{gof}}$ as, after this period expires, users are supplied with a new set of compressed messages, which marks the beginning of the upload phase of the next message exchange cycle. From the set of recovered layers, the BS creates its own message $x^{(\text{BS})}$ which is broadcasted back to the users over the broadcast downlink channel during the broadcast phase of duration $T_{dl}$. To simplify the analysis, we assume the upload and the broadcast phase do not overlap, i.e., after the upload phase of duration $T_{ul}$, the users stop transmitting and start listening the BS for the following period of duration $T_{dl}$. This analysis provides guaranteed (lower-bound) performance for the overlapping phases case, as we discuss later.

We are interested in the system design that maximizes the total average received video quality at the user terminals after a given target system delay $T = T_{\text{gof}} + T_{ul} + T_{dl}$. Note that, as $T_{\text{gof}}$ is constant and $T$ is given in advance, it follows that the system design should optimally balance between the $T_{ul}$ and $T_{dl}$ ($T_{ul} + T_{dl} = T - T_{\text{gof}} = $ const.) The time diagram of the system model, ignoring the propagation and data processing delays, is illustrated in Figure 8.

### Upload phase

The upload phase is represented by $N_u$ parallel and independent single-link transmission processes, each characterized by different message/layer sizes $(K^{(i)}, \{k_1^{(i)}, k_2^{(i)}, \ldots, k_L^{(i)}\})$ and channel state pairs $(R^{(i)}, \epsilon^{(i)})$. Assuming that the user knows[d] $(R^{(i)}, \epsilon^{(i)})$ and that the value of $T_{ul}$ is fixed in advance by the BS, the set of layer decoding probabilities $P_{d,l}^{(i-BS)}(T_{ul})$ of the $i$-th user message at the BS can be calculated for any selected EW RLC parameter $\Gamma^{(i)}(\xi)$. In the following, we focus on a simple user upload strategy where the user applies standard RLC over the largest window $l$ such that the decoding probability $P_{d,l}^{(i-BS)}(T_{ul}) > P_{\text{th}} \sim 1$ (if such exists), where $P_{\text{th}}$ is a (close to one) value of threshold probability selected in advance. More formally, the $i$-th user will apply RLC only over the $l^{(i)}$-th window, where $l^{(i)}$ is obtained as:

$$l^{(i)} = \max \left\{ l : P_{d,l}^{(i-BS)}(T_{ul}) > P_{\text{th}} \right\}. \tag{10}$$

Note that applying RLC only over the $l^{(i)}$-th window is equivalent to the special case of applying UEP RLC with the window selection distribution $\Gamma^{(i)}(\xi) = \xi^{l^{(i)}}$ (i.e., the one which places probability one on the $l^{(i)}$-th window).

Overall, the set of $N_u$ users will upload the subset of their layers, jointly described by vector $\boldsymbol{l} = \{l^{(1)}, l^{(2)}, \ldots, l^{(N_u)}\}$, within the upload phase of duration $T_{ul}$. The probability $P_{\text{th}}$ can be selected so as to keep the overall probability $P_{d,\boldsymbol{l}}^{(\text{BS})}(T_{ul}) \geq P_{\text{th}}^{N_u}$ that the BS will recover the set of users' layers described by $\boldsymbol{l}$ during the upload phase of duration $T_{ul}$ sufficiently high.

### Broadcast phase

During the broadcast phase, the BS applies the EW RLC code defined by $\Gamma^{(\text{BS})}(\xi)$ over the BS message $x^{(\text{BS})}(\boldsymbol{l})$, which is determined by the set of uploaded user layers $\boldsymbol{l}$. From $\boldsymbol{l}$, one can easily obtain the BS message size parameters $(K^{(\text{BS})}, \{k_1^{(\text{BS})}, k_2^{(\text{BS})}, \ldots, k_L^{(\text{BS})}\})$. The broadcast phase can be also analyzed using the single-link analysis applied on the parameters of the broadcast transmission, as seen by each of the system users. In other words, given $\Gamma^{(\text{BS})}(\xi)$, $x^{(\text{BS})}(\boldsymbol{l})$ and the BS-to-user-$i$ (BS-$i$) transmission link parameters $(R^{(\text{BS})}, \{\epsilon^{\text{BS}-i}\}_{1 \leq i \leq N_u})$, the single-link analysis provides the set of layer decoding probabilities $P_{d,l}^{(\text{BS}-i)}(T_{dl}), 1 \leq l \leq L$, describing the $i$-th user capability to recover the layers of the BS message $x^{(\text{BS})}(\boldsymbol{l})$ after the broadcast phase. Thus the broadcast phase reduces to the EW RLC design problem for multicast/broadcast setup that aims to simultanously satisfy heterogeneous user link conditions ($\{\epsilon^{\text{BS}-i}\}_{1 \leq i \leq N_u}$). This problem has been recently addressed for expanding window fountain (EWF) code design in video multicast setup [27], however, with the difference that in this article, instead of broadcasting a single stream, the BS simultaneously broadcasts a mixture of $N_u$ user streams.

Each user simultaneously receives $N_u - 1$ video streams originating at the remaining system users. The average received video quality $D^{(i)}$ perceived by the $i$-th user is obtained by averaging over the received video qualities of all $N_u - 1$ video streams:

$$D^{(i)} = \frac{1}{N_u - 1} \sum_{j=1, j \neq i}^{N_u} D_j^{(i)}, \tag{11}$$

where $D_j^{(i)}$ is the average received PSNR of the $j$-th user video content as perceived by the $i$-th user. $D_j^{(i)}$ can be obtained by combining the results of the upload and the broadcast phase analysis:

$$D_j^{(i)} = P_{d,\boldsymbol{l}}^{(\mathrm{BS})}(T_{ul}) \sum_{l=1}^{l(j)} P_{d,1:l}^{(\mathrm{BS}-i)}(T_{dl}) \cdot D_{j,1:l}, \qquad (12)$$

where the sum is taken over the set of $l(j) \leq L$ layers of the $j$-th user included in $\boldsymbol{x}^{(\mathrm{BS})}(\boldsymbol{l})$. In the above expression, $P_{d,1:l}^{(\mathrm{BS}-i)}(T_{dl})$ is the probability that exactly the first $l$ layers of the BS message $\boldsymbol{x}^{(\mathrm{BS})}(\boldsymbol{l})$ are recovered at the user $i$:

$$P_{d,1:l}^{(\mathrm{BS}-i)}(T_{dl}) = \begin{cases} 1 - P_{d,1}^{(\mathrm{BS}-i)}(T_{dl}), & l = 0 \\ P_{d,l}^{(\mathrm{BS}-i)}(T_{dl}) \cdot \\ \quad \cdot (1 - P_{d,l+1}^{(\mathrm{BS}-i)}(T_{dl})), & 0 < l < L \\ P_{d,l}^{(\mathrm{BS}-i)}(T_{dl}) & l = L, \end{cases} \qquad (13)$$

and $D_{j,1:l}$ is the average received PSNR of the $j$-th user video content after recovery of the first $l$ layers (averaged over all the frames of the transmitted GOF). Finally, the average received PSNR, averaged across all the users of the multi-user video streaming session, is equal:

$$D = \frac{1}{N_u} \sum_{i=1}^{N_u} D^{(i)}. \qquad (14)$$

### *System parameters and design*

From (11)–(14), by factoring out $P_{d,\boldsymbol{l}}^{(\mathrm{BS})}(T_{ul})$, we note that the distortion-optimized system design allows for independent design of the upload and the broadcast phase, given the duration $T_{ul}$ and decoding probability threshold $P_{\mathrm{th}}$ are fixed. In other words, by fixing and informing the users on the values of $T_{ul}$ and $P_{\mathrm{th}}$, the set of layers $\boldsymbol{l}(T_{ul})$ that can be reliably uploaded by users in the upload phase can be determined by corresponding users. Consequently, the BS message $\boldsymbol{x}^{(\mathrm{BS})}(\boldsymbol{l})$ and $T_{dl} = T - T_{\mathrm{gof}} - T_{ul}$ is also determined, which reduces the broadcast phase design to optimization of the EW RLC code parameter $\Gamma^{(\mathrm{BS})}(\xi)$ such that the average received video quality $D$ is maximized after the target system delay $T$.

Overall, for the distortion-optimized system design, the BS should optimally balance between the upload and the broadcast phase by selecting appropriate $T_{ul}$, appropriate threshold probability $P_{\mathrm{th}}$, and optimally satisfy heterogeneous user requirements by selecting optimized $\Gamma^{(\mathrm{BS})}(\xi)$. The optimal solution weights between the number of layers that could be uploaded to the BS with reliability $P_{\mathrm{th}}$ after $T_{ul}$ and their quality of reconstruction at the set of heterogeneous users after $T_{dl}$.

## System optimization and results
### System optimization

For the system model and distortion-optimized design discussed above, the system optimization process is performed centrally, e.g., at the video conference server collocated with the central BS node. Given the parameters of all the user messages $(K^{(i)}, \{k_1^{(i)}, k_2^{(i)}, \dots, k_L^{(i)}\})$, uplink channel conditions $(R^{(i)}, \epsilon^i)$ and the broadcast channel conditions $(R^{(\mathrm{BS})}, \{\epsilon^{\mathrm{BS}-i}\}_{1 \leq i \leq N_u})$, the BS should provide the duration of the upload phase $T_{ul}$, the threshold probability $P_{\mathrm{th}}$ and the EW RLC code design parameter $\Gamma^{(\mathrm{BS})}(\xi)$, such that the average received PSNR $D$ is maximized after the target system delay $T$. In other words, the BS solves the following problem:

$$\max_{T_{ul}, P_{\mathrm{th}}, \Gamma^{(\mathrm{BS})}(\xi)} D, \qquad (15)$$

where $0 \leq T_{ul} \leq \min\{T_{\mathrm{gof}}, T - T_{\mathrm{gof}}\}$ and for $\Gamma^{(\mathrm{BS})}(\xi)$ we have $0 \leq \Gamma_l^{(\mathrm{BS})} \leq 1, 1 \leq l \leq L$ and $\sum_{l=1}^{L} \Gamma_l^{(\mathrm{BS})} = 1$.

Assuming that the BS knows the channel conditions (e.g., by measurements and user reporting), it still needs to know the user message parameters $(K^{(i)}, \{k_1^{(i)}, k_2^{(i)}, \dots, k_L^{(i)}\})$ to be able to perform the above optimization. Since these data cannot be obtained instantaneously at the BS, to avoid delays, we assume that the BS uses information available from recent GOF exchanges (e.g., the last GOF or the average over last several GOFs). This way, the BS is able to perform system optimization prior to the start of the upload phase and to broadcast the required parameters $T_{ul}$ and $P_{\mathrm{th}}$ back to the users. The users then determine the number of layers $l^{(i)}$ to upload to the BS and start the upload phase.

In general, the complexity of calculation of the set of layer decoding probabilities in Sections "Performance analysis of EW RLC" and 2 grows exponentially, due to an exponential number of terms in sums given in Equations (4) and (5), as $K$ and $L$ grows. However, in practical applications, the calculations are tractable due to the fact that $K$, $L$ and $N_u$ are usually small. For example, $K$ is already bounded by GE decoding complexity and should not exceed $K \sim 100$; the number of scalable video layers is typically small, e.g., $L < 5$; and for comfortable use of real-time multi-user video conferencing system, $N_u$ should also be small, e.g., $N_u < 5$. (note that $N_u$ can be larger as long as each user displays only a small subset of active user streams). With the restrictions on $K$, $L$ and $N_u$, the optimization problem can be evaluated at the BS side server with acceptable delay. Alternatively, the BS may run optimization less frequently then on a GOF-by-GOF basis, using accumulated averages of channel conditions and GOF message lenghts and periodically update the users and the BS transmitter with the new values of $(T_{ul}, P_{\mathrm{th}})$ and $\Gamma^{(\mathrm{BS})}(\xi)$, respectively.

**Table 1 Parameters of H.264/SVC sequences**
**($L = 2, N_{\text{gof}} = 4$)**

| Sequence/layers | Number of packets $b = 3,200$ [bits] | Bit rate [kbps] | Y-PSNR [dB] |
|---|---|---|---|
| Stefan BL | $k_1^{(1)} = 20$ | 476.68 | 28.44 |
| Stefan BL + EL | $K^{(1)} = 60$ | 1432.56 | 34.53 |
| Foreman BL | $k_1^{(2)} = 12$ | 282.68 | 33.62 |
| Foreman BL + EL | $K^{(2)} = 42$ | 992.56 | 38.63 |
| News BL | $k_1^{(3)} = 16$ | 379.68 | 33.47 |
| News BL + EL | $K^{(3)} = 40$ | 948.56 | 38.36 |
| Coast BL | $k_1^{(4)} = 20$ | 474.68 | 30.32 |
| Coast BL + EL | $K^{(4)} = 64$ | 1522.56 | 34.69 |

### Design examples

The multi-user video streaming system design proposed in this article is illustrated using numerical examples.

**Example 3.** *In this example, we present a distortion-optimized UEP NC solution for the multi-user video conferencing system with $N_u = 4$ users (Figure 4). We assume users perform real-time exchange of H.264/SVC compressed CIF Stefan, Foreman, News and Coastguard sequences ($352 \times 288$, $N_{fps} = 30$), each user sharing a*

different video sequence. Users encode the sequences into $L = 2$ quality layers (BL and one EL) using the coarse-grain scalable (CGS) coding feature. The GOF size is set to very low value of $N_{gof} = 4$ in order to reduce the start-up coding delay $T_{gof} = 4/30 = 133$ ms to the acceptable value. The parameters of the obtained layered source messages, after H.264/SVC compression and averaged over the frames of a sample GOF we use for optimization, are given in Table 1.

For the uplink channel parameters, for each user, we select rate values around 2 Mbps and erasure probabilities in the range $\epsilon = 0.05 - 0.15$: ($R^{(1)} = 1.5$ Mbps, $\epsilon^1 = 0.07$), ($R^{(2)} = 1.8$ Mbps, $\epsilon^2 = 0.15$), ($R^{(3)} = 2.3$ Mbps, $\epsilon^3 = 0.05$) and ($R^{(4)} = 1.5$ Mbps, $\epsilon^4 = 0.12$), to account for the variations in particular uplink conditions. The BS broadcast rate is set to $R^{(BS)} = 6$ Mbps and, for simplicity, the broadcast erasure rates towards each user are set equal to the erasure rates of the corresponding uplink channels, i.e., $\epsilon^{BS-i} = \epsilon^i$.

Given the system parameters above, we seek for the optimal system parameters ($T_{ul}, \Gamma^{(BS)}(\xi)$) such that the average received PSNR D across all system users is maximized after the target delay $T = 250$ ms. For simplicity, we fix $P_{th} = 0.99$. The solution is illustrated in Figure 9 where average PSNR is plotted as a (two-dimensional) function of ($T_{ul}, \Gamma_1^{(BS)}$). The system achieves the best
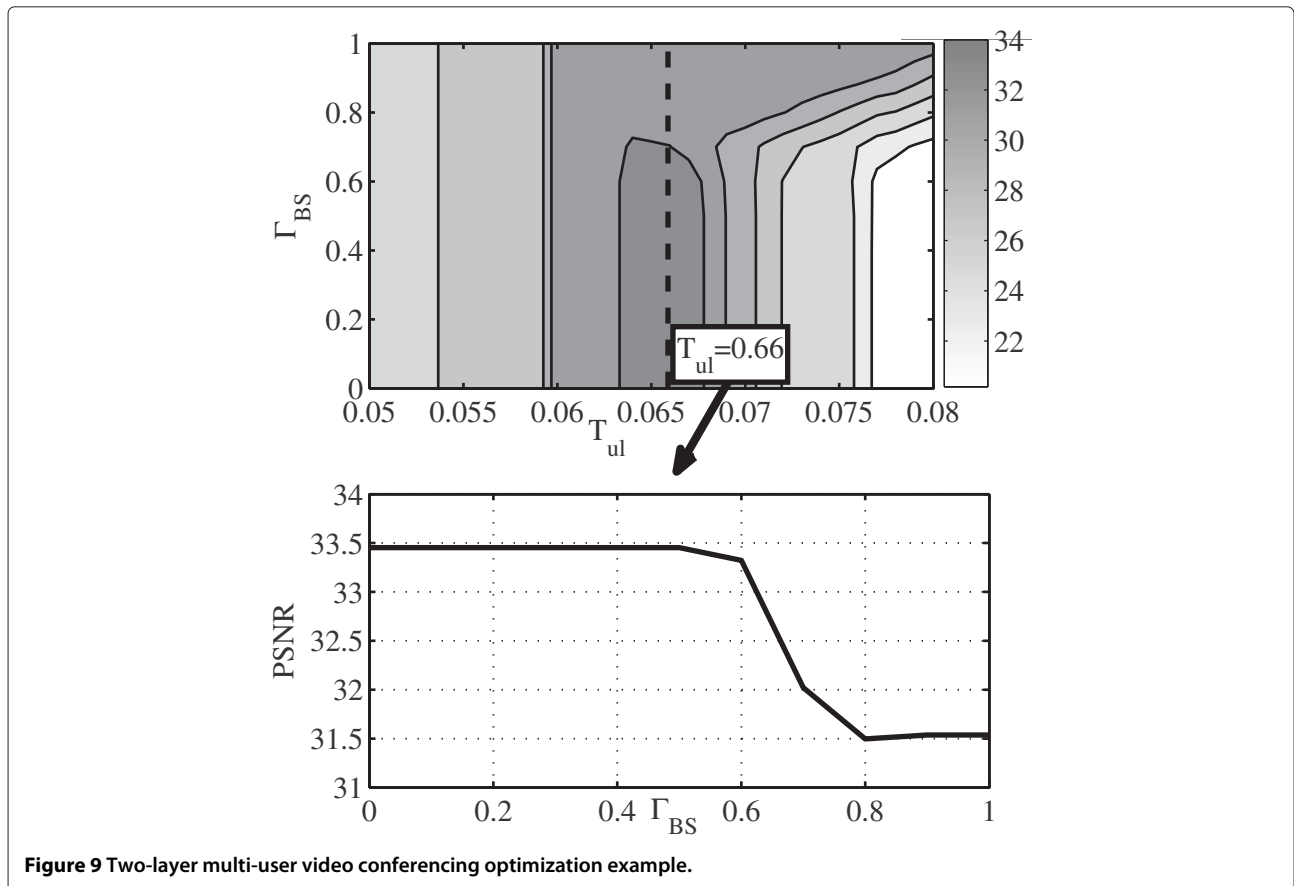


**Figure 9 Two-layer multi-user video conferencing optimization example.**

**Table 2 Expected delays in Example 1**

| Upload transmission | $\mathbb{E}_l[T]$ [ms] | Broadcast transmission | $\mathbb{E}_l[T]$ [ms] |
|---|---|---|---|
| $U_1$-BS | $\mathbb{E}_1[T] = 45.80$ | BS-$U_1$ | $\mathbb{E}_2[T] = 41.29$ |
| $U_2$-BS | $\mathbb{E}_1[T] = 25.09$ | BS-$U_2$ | $\mathbb{E}_2[T] = 50.20$ |
| $U_3$-BS | $\mathbb{E}_2[T] = 58.58$ | BS-$U_3$ | $\mathbb{E}_2[T] = 29.20$ |
| $U_4$-BS | $\mathbb{E}_1[T] = 48.48$ | BS-$U_4$ | $\mathbb{E}_2[T] = 43.36$ |

average performance for $T_{ul} = 66$ ms where all users are able to share at least their BL, while user 3 is able to upload both layers to the BS, i.e., $l = (1, 1, 2, 1)$. For optimal $T_{ul} = 66$ ms, a separate (lower) graph shows the system performance for different EW RLC codes at the BS. Although the maximum of D is achieved over the range of first window selection probabilities $\Gamma_1^{(BS)}$, it is favourable to select as large $\Gamma_1^{(BS)}$ as possible to reduce the decoding delay for the first layer, while still maintaining high probability of recovery of the second layer of $x^{(BS)}$ at all users.[e]

Table 2 illustrates the average decoding delays for upload and broadcast phase transmisssions for the set of uploaded layers $l = (1, 1, 2, 1)$ and the solution point $(T_{ul}, \Gamma_1^{(BS)})$=(0.066,0). We can easily note that the sum of maximum delays experienced during the upload/broadcast phase closely satisfies the delay limits imposed by the system: $\mathbb{E}_2^{(2-BS)}[T] + \mathbb{E}_2^{(BS-3)}[T] = 108.78 < 117 = T - T_{gof}$, where maximum upload delay is below selected upload duration $\mathbb{E}_2^{(2-BS)}[T] = 58.58 < 66 = T_{ul}$. This points out to the possibility of approximated system design using expected delay calculations.

**Example 4.** *Additional flexibility in the system design is obtained if the users compress their video streams into larger number of layers. In this example, we observe the performance of the distortion-optimized system design for the same transmission parameters as in the previous example, but where the layered source message is compressed into $L = 4$ quality layers (see Table 3 for the message parameters). The system performs optimally for $T_{ul} = 64$ ms where users are able to upload the set of layers $l = (2, 3, 3, 2)$, where $P_{th} = 0.99$ is assumed fixed. The EW RLC broadcast phase parameters that achieve the optimal value $D = 34.88$ are for the window selection distribution $\Gamma^{(BS)}(\xi) = 0.5\xi + 0.5\xi^3$. We note that the gain obtained in average system distortion D is not large, due to the fact that compressing video into larger number of layers introduces small performance penalties, but the system flexibility reflected through better layer resolution provides more options for the system design process.*

**Decode-and-broadcast versus buffer-and-broadcast**
In the proposed system, we apply *decode-and-broadcast* operation in central multi-user video streaming point: the

uploading streams are firstly decoded and then broadcasted within the non-overlapping broadcasting stage. Clearly, this approach simplifies applications of our analytical tools and enables simple and elegant system design, however, improvements are possible if the broadcast phase is initialized before the incoming user messages are completely recovered. A possible improvements are shortly commented below.

*Layer-by-layer decode-and-broadcast*
Let us assume the upload phase where a general UEP RLC is applied instead of the specific RLC case that encodes the largest window decodable within $T_{ul}$. In this case, the unequal recovery time (URT) property enables the central point to decode user layers sequentially over time, starting from the BL onwards [18]. Thus the central point is able to produce encoded packets as soon as the BL of the message $x^{(BS)}$ is decoded and include additional layers as soon as they become available while updating the broadcast EW RLC code parameter $\Gamma^{(BS)}(\xi)$ "on the fly," as illustrated in Example 2. We note that this scenario introduces a trade-off between increase in the upload delays of higher layers and decrease in the beginning of the broadcast phase, which has to be balanced by the optimal solution. Unfortunately, the distortion optimized system design for this scenario would result in tedious optimization problem, which is why we leave it out of consideration. However, we note that expected delay analysis, similar to the one presented in Table 2, could

**Table 3 Parameters of H.264/SVC sequences ($L = 4, N_{gof} = 4$)**

| Sequence/ layers | Number of packets $b = 3,200$ [bits] | Bit rate [kbps] | Y-PSNR [dB] |
|---|---|---|---|
| Stefan BL | $K_1^{(1)}$=15 | 356.22 | 25.89 |
| Stefan BL + EL | $K_2^{(1)}$=24 | 567.46 | 28.15 |
| Stefan BL + 2EL | $K_3^{(1)}$=40 | 951.18 | 30.65 |
| Stefan BL + 3EL | $K_4^{(1)}$=64 | 1522.05 | 33.23 |
| Foreman BL | $K_1^{(2)}$=7 | 162.68 | 29.45 |
| Foreman BL + EL | $K_2^{(2)}$=13 | 309.87 | 32.3 |
| Foreman BL + 2EL | $K_3^{(2)}$=24 | 569.93 | 34.52 |
| Foreman BL + 3EL | $K_4^{(2)}$=48 | 1150.6 | 38.41 |
| News BL | $K_1^{(3)}$=10 | 235.67 | 28.99 |
| News BL + EL | $K_2^{(3)}$=19 | 448.66 | 32.55 |
| News BL + 2EL | $K_3^{(3)}$=32 | 759.32 | 35.21 |
| News BL + 3EL | $K_4^{(3)}$=50 | 1118.19 | 38.05 |
| Coast BL | $K_1^{(4)}$=7 | 164.76 | 26.66 |
| Coast BL + EL | $K_2^{(4)}$=16 | 378.52 | 28.95 |
| Coast BL + 2EL | $K_3^{(4)}$=33 | 787.74 | 30.74 |
| Coast BL + 3EL | $K_4^{(4)}$=60 | 1435.02 | 33.55 |

be used as a simple approximation for the layer-by-layer decode-and-broadcast system design.

### Buffer-and-broadcast

Finally, the simplest *buffer-and-broadcast* solution follows the standard NC approach in which all the received encoded packets are buffered, and new encoded packets produced by applying RLC over the buffer content [6,7]. In the proposed UEP RLC case, the central point maintains $L$ separate buffers, each collecting encoded packets of different users produced over one of the $L$ windows. As soon as the upload phase starts filling the buffers, the broadcast phase starts producing encoded packets where each encoded packet results from applying RLC over one of the buffers selected independently by the appropriate window (i.e., buffer) selection distribution $\Gamma^{(BS)}(\xi)$. Although very efficient, this solution lacks efficient analysis and distortion-based optimization tools. In addition, the problem of broadcasting linearly dependent encoded packets may become significant as the upload user rates decrease and broadcast rate increases (i.e., the rate of encoded packets generation exceeds the rate of incoming source data).

## Conclusions

Real-time sharing of video content among multiple users over wireless networks is underlying a number of existing and upcoming mobile multimedia services. For robust, flexible and efficient implementation of such services, this article considered a combination of scalable video coding and UEP NC. We have presented analytical tools capable of producing the values of key system design parameters that result in the distortion-optimal system performance. The applications of the proposed tools are illustrated through several examples involving a simple single access point multi-user scenario.

## Endnotes

[a] For compactness, we denote $R_l(\boldsymbol{n})$ as $R_l$.

[b] Note that, due to the probabilistic encoding, the decoding performance is independent of the packet erasure process in the channel and depends only on the number $N$ of received packets.

[c] This model roughly captures the behaviour of adaptive modulation and coding (AMC) at the physical layer of cellular systems where, depending on the channel quality feedback available at the BS, different AMC modes could be approximated by different $(R, \epsilon)$ pairs. We assume slowly-varying channels where AMC mode changes are of the order of $T_{\mathrm{gof}}$.

[d] In state-of-the-art wireless cellular broadband systems such as LTE or WiMAX, channel quality indicators (CQI) are continuously fed back by user equipment to the BS.

[e] For presentation purpose, Figure 9 is obtained by brute-force calculation over a grid of points in $(T_{ul}, \Gamma_1^{(BS)})$ space. In general, (one of) the optimal solution(s) can be obtained by applying nonlinear programming methods such as sequential quadratic programming (e.g., using MATLAB).

**Author details**
[1] Department of Power, Electronics and Communication Engineering, University of Novi Sad, Trg D. Obradovića 6, Novi Sad, Serbia. [2] Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, G1 1XW, Glasgow, UK.

**References**
1. H Shiang, M van der Schaar, Multi-user video streaming over multi-hop wireless networks: a distributed, cross-layer approach based on priority queuing. IEEE J. Sel. Areas Commun. **25**(4), 770–785 (2007)
2. X Zhu, P Agrawal, J Pal Singh, T Alpcan, B Girod, Rate allocation for multi-user video streaming over heterogenous access networks. in *ACM MULTIMEDIA '07,* (Augsburg, Germany, 2007), pp. 37–46
3. R Ahlswede, N Cai, S yen Robert Li, RW Yeung, Network information flow. IEEE Trans. Inf. Theory. **46**(4), 1204–1216 (2000)
4. S yen Robert Li, RW Yeung, N Cai, Linear network coding. IEEE Trans. Inf. Theory. **49**(2), 371–381 (2003)
5. T Ho, M Medard, R Koetter, DR Kargerm, M Effros, J Shi, B Leong, A random linear network coding approach to multicast. IEEE Trans. Inf. Theory. **52**(10), 4413–4430 (2006)
6. PA Chou, Y Wu, K Jain, Practical network coding. in *Allerton 2003 Conference,* (2003)
7. DS Lun, M Medard, R Koetter, M Effros, On coding for reliable communication over packet networks. Phys. Commun. **1**, 3–20 (2008)
8. C Gkantsidis, P Rodriguez, Network coding for large scale content distribution. in *IEEE INFOCOM 2005,* (Miami, FL, USA, 2005), pp. 2235–2245
9. P Chou, Y Wu, Network coding for the internet and wireless networks. IEEE Signal Process. Mag. **24**(5), 77–85 (2007)
10. E Magli, P Frossard, An overview of network coding for multimedia streaming. in *IEEE ICME 2009,* (New York, NY, USA, 2009), pp. 1488–1491
11. J Zhao, F Yang, Q Zhang, Z Zhang, F Zhang, LION: layered overlay multicast with network coding. IEEE Trans. Multimed. **8**(5), 1021–1032 (2006)
12. M Wang, B Li, R$^2$: random push with random network coding in live peer-to-peer streaming. IEEE J. Sel Areas Commun. **25**(9), 1655–1666 (2007)
13. H Seferoglu, A Markopoulou, Video-aware opportunistic network coding over wireless networks. IEEE J. Sel. Areas Commun. **27**(5), 713–728 (2009)
14. N Thomos, P Frossard, Network coding of rateless video in streaming overlays. IEEE Trans. Circ. Syst. Video Techn. **20**(12), 1834–1847 (2010)
15. H Wang, R Chang, CCJ Kuo, Wireless multi-party video conferencing with network coding. in *IEEE ICME 2009,* (New York, NY, USA, 2009), pp. 1492–1495
16. M Ponec, S Sengupta, M Chen, J Li, PA Chou, Multi-rate peer-to-peer video conferencing: a distributed approach using scalable coding. in *IEEE ICME 2009,* (New York, NY, USA, 2009), pp. 1406–1413
17. H Zhang, J Zhou, Z Chen, J Li, Minimizing delay for video conference with network coding. in *ACM SIGCOMM 2009,* (Barcelona, Spain, 2009)
18. D Vukobratović, V Stanković, Unequal error protection random linear coding strategies for erasure channels. IEEE Trans. Commun. **60**(5), 1243–1252 (2012)

19. Y Wu, P Chou, SY Kung, Information exchange in wireless networks with network coding and physical-layer broadcast. in *Proc. CISS 2005*, (Baltimore, MD, USA, 2005)
20. S Katti, H Rahul, W Hu, D Katabi, M Medard, J Crowcroft, XORs in the air: practical wireless network coding. in *ACM SIGCOMM 2006*, (Pisa, Italy, 2006), pp. 243–254
21. U Horn, K Stuhlmuller, M Link, B Girod, Robust internet video transmission based on scalable coding and unequal error protection. Signal Process. Image Commun. **15**, 77–94 (1999)
22. V Stankovic, R Hamzaoui, Live video streaming over packet networks and wireless channels. in *IEEE Packet Video 2003*, (Nantes, France, 2003)
23. E Maani, AK Katsaggelos, Unequal error protection for robust streaming of scalable video over packet lossy networks. IEEE Trans. Circ. Syst. Video Tech. **20**(3), 407–416 (2010)
24. H Shojania, B Li, Random network coding on the iPhone: fact or fiction? in *ACM NOSSDAV 2009, USA*, (Williamsburg, VA, USA, 2009), pp. 37–42
25. P Vingelmann, F Fitzek, M Pedersen, J Heide, H Charaf, Synchronized multimedia streaming on the iPhone platform with network coding. in *IEEE CCNC 2011, USA*, (Las Vegas, NV, USA, 2011), pp. 875–879
26. L Wilhelmsson, LB Milstein, On the effect of imperfect interleaving for the Gilbert–Elliott channel. IEEE Trans. Commun. **47**(5), 681–688 (1999)
27. D Vukobratovic, V Stankovic, D Sejdinovic, L Stankovic, Z Xiong, Scalable video multicast using expanding window fountain codes. IEEE Trans. Multimed. **11**(6), 1094–1104 (2009)