Optimizing the Joint Source/Network Coding for Video Streaming over Multi-hop Wireless Networks

Huali Cui[1], Depei Qian[1,2], Xingjun Zhang[1], Ilsun You[3] and Xiaoshe Dong[1]
[1] Department of Computer Science and Technology, Xi'an Jiaotong University
Xi'an, 710049, P.R.China
[e-mail: hualic@stu.xjtu.edu.cn, xjzhang@mail.xjtu.edu.cn, xsdong@mail.xjtu.edu.cn]
[2] Sino-German Joint Software Institute, School of Computer Science, BeiHang University
Beijing, 100191, P.R.China
[e-mail: depeiq@buaa.edu.cn]
[3] School of Information Science, Korean Bible University
Seoul, 139-791, South Korea
[e-mail: isyou@bible.ac.kr]
*Corresponding author: Xingjun Zhang

Received January 17, 2013; revised March 16, 2013; accepted April 2, 2013; published April 30, 2013

Abstract

Supporting video streaming over multi-hop wireless networks is particularly challenging due to the time-varying and error-prone characteristics of the wireless channel. In this paper, we propose a joint optimization scheme for video streaming over multi-hop wireless networks. Our coding scheme, called Joint Source/Network Coding (JSNC), combines source coding and network coding to maximize the video quality under the limited wireless resources and coding constraints. JSNC segments the streaming data into generations at the source node and exploits the intra-session coding on both the source and the intermediate nodes. The size of the generation and the level of redundancy influence the streaming performance significantly and need to be determined carefully. We formulate the problem as an optimization problem with the objective of minimizing the end-to-end distortion by jointly considering the generation size and the coding redundancy. The simulation results demonstrate that, with the appropriate generation size and coding redundancy, the JSNC scheme can achieve an optimal performance for video streaming over multi-hop wireless networks.

Keywords: Network coding, multi-hop wireless networks, video streaming

A preliminary version of this paper appeared in IEEE VTC 2012-Spring, May 6-9, Yokohama, Japan. This version involves more technical details, intensive optimization analysis and simulations. This work is partially supported by National Natural Science Foundation of China (No. 60673180), National Key Technologies R&D Program of China (No. 2011BAH04B03), Joint Chinese-Italian Research Infrastructure (No. 2009DFA12110) and National High Technology Research and Development Program of China (No. 2011AA01A204, No. 2012AA01A306).

http://dx.doi.org/10.3837/tiis.2013.04.011

1. Introduction

Real-time video streaming applications such as video telephony, video conferencing, and video gaming over wireless multi-hop networks have attracted broad attention recently. Due to the time-varying and error-prone characteristics of the wireless channel, supporting video streaming over multi-hop wireless networks faces greater technical challenges [1][2][3]. Although the bandwidth of the wireless networks has been increased significantly in recent years, it is still difficult to meet the bandwidth requirement of high-quality multimedia streaming applications. Furthermore, high packet loss rate encountered in the wireless networks makes the video delivery even more challenging, especially in the case of multi-hop wireless mesh networks.

On the other hand, the wireless multi-hop network is able to perform basic operations at intermediate network nodes, which eases the implementation of network coding. Network coding was initially proposed as a solution for efficient utilization of the network bandwidth [4]. Besides the benefit obtained in increasing throughput, network coding also shows better resilience to errors, which enables network coding to be used in improving the quality of video transmissions [5][6].

Theoretically, network coding can be implemented across the entire flow. However, this is not feasible in practice because the relay node with limited buffer space can only buffer the received packets for a short period of time. For this reason, the whole streaming flow should be split into segments and each segment can be encoded separately. Packets should be encoded with similar decoding time-stamps so that the decoding is not excessively delayed. In order to deal with the timing constraints, the concept of generations has been introduced in [7]. Generally, one video stream is divided into several generations, and only packets from the same generation are combined together (intra-session network coding) [8]. Coding per generation has multiple practical advantages for streaming applications [7]. First, it permits the deployment of distributed solutions and limits the overhead of coding and decoding. Secondly, as the stream is split into generations, the strict delay requirements in video streaming scenarios can be satisfied.

In the case of network coding with generations, the size of generation has significant impact on the performance of media communication over multi-hop wireless networks, and therefore, is a critical network coding parameter. The packets coded with random network coding will have equal significance. The more packets encoded within a generation, the better chance for the intermediate node to generate novel packets for outgoing links. The packet diversity will be enhanced by larger generation size. However, it is obvious that the decoding delay at the destination will increase with larger generation size. For video streaming applications, delay must be minimized to guarantee continuous playback at the receiver side. For this reason the generation size cannot be too large. So, it is very important to select the appropriate generation size by making tradeoff between decodability and playback delay.

Redundancy introduced by the intra-session network coding is another important factor affecting the streaming performance. Adding redundant packets can help to recover the lost packets in a timely fashion. But on the other hand, there are two drawbacks with this approach. First, from the perspective of network load, adding redundant packets will increase the raw packet loss rate in the network because of the additional loads introduced by the redundant packets. Secondly, higher redundancy implies increased distortion induced by the source encoder. In our study, for multi-hop wireless networks with one-hop network coding, both packet loss rate and burst length involved in the hop are taken into account to determine the amount of necessary redundancy.

In this paper, we propose a network coding optimization scheme for video streaming over multi-hop wireless networks. We consider the scenario where a compressed video sequence is sent from the server to the end users via a mesh network with lossy wireless channels. As illustrated in **Fig. 1**, the streaming server is connected to the mesh network with gateway router. The client is connected to multiple relay nodes which are able to perform network coding operations. We define the relay nodes with coding capability as the intermediate nodes. The nodes can overhear the transmissions in their neighborhood due to the shared nature of the wireless media, as indicated by the dash line in **Fig. 1**. We analyze how several metrics such as decoding delay, diversity of packets in the network and video distortion are affected by the size of generation and network coding redundancy.

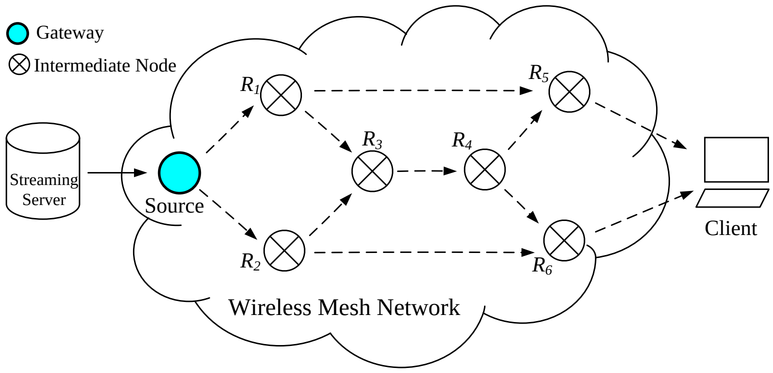


Fig. 1. Streaming over wireless mesh network

The objective of this paper is to reveal the effects of two different coding parameters, generation size and redundancy level, to the video quality. The expected end-to-end video distortion under the wireless network circumstance and the coding constraints is minimized by a Joint Source/Network Coding (JSNC) scheme which jointly optimizes those two coding parameters.

The rest of the paper is organized as the follows. Related work is discussed in Section 2. In Section 3, we present the JSNC scheme. Section 4 gives the formulation of distortion optimization and the corresponding algorithm, and analyzes the overhead of the JSNC scheme. Packet-level simulation and the results are presented in Section 5. Section 6 concludes the paper.

2. Related Work

The issues of applying network coding to video streaming over unreliable networks have been addressed in several research works [9][10][11]. In [9], network coding is used for video streaming over wireless mesh networks. When the transmitted flows are video streams, network codes should be selected so as to not only maximize the network throughput, but also improve the video quality. The authors propose a video-aware opportunistic network coding scheme which combines packets from different streams so as to increase throughput. At the same time, the decodability of the network code at the receiver, the packets importance in terms of video distortion, and the playback deadlines of video packets must all be taken into account in selection of coding scheme that contributes the most to the quality of video streams.

A similar work is presented in [10], a network coding system is used for WLAN-like or WiMAX-like networks. This scheme employs an optimized scheduling algorithm based on the Markov decision process to increase the bandwidth efficiency and maximize the multimedia transmission in both broadcast and unicast scenarios.

A streaming system over lossy overlay networks that benefits from joint Raptor codes and network coding is presented in [11]. The server applies a non-systematic Raptor coding on the streaming packets for error resiliency. In order to increase the packet diversity in the system, the overlay nodes selectively combine the coded packets according to the current available bandwidth. If the number of received packets exceeds the available outgoing bandwidth, some of the packets will be discarded. Otherwise, the relay node will create and forward new linear combinations of packets, thus the available resources can be fully exploited.

A practical network coding scheme which can be implemented in real networks is the random network coding scheme [12]. In this scheme, the data flow is divided into generations, and each generation contains a number of packets. Some factors influenced by the generation size have been identified, such as the complexity and performance of encoding and decoding, and the header overhead for storing the coefficient vectors [13].

In [14], the authors analyze the effect of "encoding number" in practical wireless network coding, i.e., how many packets can be encoded by the relay node in each transmission. They provide an upper bound of the encoding number in all possible coding approaches. They also propose a methodology for obtaining the average encoding number and the corresponding system throughput under a realistic wireless setting. The work in [14] tries to answer the question of how many packets can be encoded. However it does not tell us how many packets should be encoded to satisfy various applications requirements such as maximizing the end-to-end quality of video transmission. The authors of [15] also investigate this issue. The strategy they propose is to determine the generation size according to the application requirements (e.g., media transmission) to UDP, or to set the generation size to equal to the TCP congestion window. [16] proposes a theoreticl framework for analyzing the effect of the generation size to TCP traffic with intra-session network coding.

The impact of generation size on different coding schemes and different network scenarios is studied recently. The performance of segmented network coding (SNC) for bulk or stream-like data dissemination in Delay Tolerant Networks (DTNs) is analyzed. The behavior of epidemic routing using SNC for bulk data dissemination in DTNs is investigated. In the proposed feedbackless SNC-based protocol, the maximum sustained throughput is an increasing function of the segment (generation) size [17]. Three application-layer coding schemes for streaming over single-hop lossy links, i.e., random linear coding (RL), systematic random linear coding (RLS), and structured coding (MDS), are compared in [18]. The delivery packet count, net data throughput, and energy consumption are the evaluation metrics with a range of generation sizes.

In the mesh architecture, the relay nodes can perform much more than only packets forwarding. Coding can be implemented on the relay nodes and contribute to improving the performance of the streaming system. In order to enhance the reliability and throughput of video packet transmission, Reed-Solomon codes have been implemented in network-embedded Forward Error Correction coding (NEF). [19] introduces a NEF framework for overlay and P2P networks, which can outperform end-to-end FEC dramatically in terms of decodable probability and video quality. However, the NEF code suffers from its complexity and the increased latency. The reason is that both decoding and re-coding operations are needed on the intermediate nodes.

In this paper, we propose a joint source and network coding (JSNC) scheme for optimizing video streaming over multi-hop wireless networks. JSNC jointly uses source coding and network coding so that the source and the intermediate network nodes can jointly contribute to the effective delivery of multimedia streams. From the perspective of source coding, the number of packets in a generation, i.e., the size of the generation, is changing with the source coding rate. On the other hand, metrics such as decoding delay, packets diversity and video distortion are all influenced by the generation size. It is necessary to jointly consider the video source coding, network coding, and network infrastructure features when using network coding to support video streaming.

3. The Joint Source/Network Coding Scheme

3.1 Wireless Channel Model

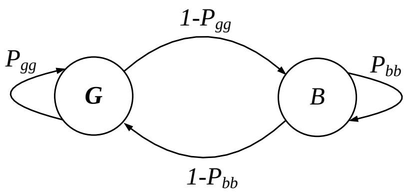


Fig. 2. Gilbert-Elliott model

The packet-loss patterns introduced by the wireless channel may be bursty. To capture the burst characteristics of link connectivity, we use the Gilbert-Elliott (GE) model [20] to characterize the wireless channel. GE model is a two-state Markov chain with two states denoted as *G* (Good) and *B* (Bad), as illustrated in **Fig. 2**. If a packet is transmitted from the source along a link in state *G*, it will be received correctly and timely, otherwise, over a channel in state *B*, packets are assumed to be lost before reaching the destination. P_{gg} denotes the probability of self-transition in state *G* and P_{bb} denotes the self-transition probability in the state *B*. Let P_G denotes the stationary probability that a channel is in state *G* and P_B the stationary probability of a channel in state *B*. Relationship among them is given as the follows:

$$P_G = \frac{1-P_{bb}}{2-P_{bb}-P_{gg}}, \quad P_B = 1-P_G \tag{1}$$

The amount of time that a channel stays in the *G* state is $L_G = 1/(1-P_{gg})$. After L_G, it changes to the *B* state and the average length of the burst error is $L_B = 1/(1-P_{bb})$.

3.2 Network Coding Model

The advantages of intra-session network coding for error resiliency can be illustrated by the simple two-link tandem network in **Fig. 3** [21]. The source *S* transmits packets to the destination *D* through the relay node *R*. The capacity of two links is one packet per time unit. ε_1 and ε_2 are the loss rate of the links $S \rightarrow R$ and $R \rightarrow D$, respectively, the transmission rate from source to destination is $(1-\varepsilon_1)\cdot(1-\varepsilon_2)$ packets per time slot. However, if the relay node can re-encode the packets, the communication rate increases and becomes equivalent to the minimum of the capacities: $\min\{(1-\varepsilon_1),(1-\varepsilon_2)\}$. Similarly, traditional channel coding with full decoding and re-encoding at node *R* also has the capability of enlarging the capacity. But network coding does not need the decoding operations at the intermediate nodes.

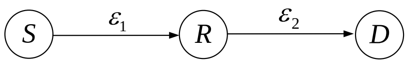


Fig. 3. Two-links tandem channel

Random linear network coding (RLNC) is proposed by T. Ho and M. Medard [12]. RLNC permits implementation of a distributed solution with independent coding decisions on each node. Each coding coefficient is randomly and independently assigned over the Galois Field (GF) without the full knowledge of network topology, and its computational complexity is significantly low. Such a distributed algorithm is particularly useful to multi-hop networks.

RLNC can achieve a reasonably high successful decoding probability with a relatively small field size. It has been shown that the successive network coding operations is with a 99.6% decodable probability when the computations are performed on GF (2^{16}). Generally, the field size of GF (2^8) is sufficient in practice [7]. It has been shown that this field size can guarantee high symbol diversity and low probability of building duplicate packets.

Consider a network coding flow transmitted at rate R_{nc} to the destination which does not violate the flow conservation constraint. Denote the flow rate on link (u, v) from node u to node v as R_u^v. At each intermediate node, the sending rate of fresh information flows must equal to the incoming rate of innovative packets. Consequently, the flow conservation constraints can be expressed as follows:

$$\sum_{v:(u,v)\in E} R_u^v - \sum_{v:(v,u)\in E} R_v^u = \begin{cases} R_{nc} & \text{if } u = \text{Src.} \\ -R_{nc} & \text{if } u = \text{Dst.} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where E is the set of directed links in the network.

3.3 The JSNC Scheme

Our Joint Source/Network Coding (JSNC) scheme works as the following. At the source node, the flow s is divided into generations with size K, where K may vary from one generation to another. Here for simplicity of computation, all the generations are with the same size. The K uncoded packets are called native packets. The source node adds redundant packets by creating a random linear combination of the K native packets within the generation. The number of redundant packets added depends on the loss rates of the links involved in the corresponding hop via intra-session network coding [8]. A coded packet can be expressed as a function of the native packets:

$$y_m = \sum_{i=1}^{K} f_{m,i} \cdot x_i \tag{3}$$

where $f_{m,i}$ is the coefficient randomly selected from the given finite field GF(2^8), $x = (x_1, x_2, \cdots, x_K)$ are the native packets from one generation. ($f_{m,1}, \ldots, f_{m,j}, \ldots, f_{m,K}$) is called the code vector of packet x_i, i=1,…,K.

Each data packet is tagged with the packet's global encoding vector, the generation number and other related information. The source node sends out the M coded packets $y=(y_1, y_2, \cdots, y_M)$ within a generation until the end of the generation and then proceeds to the next generation.

When an intermediate node hears a packet, it first checks whether the packet is an innovative packet. A packet is innovative if it is linearly independent of other packets that the node has previously received from the same generation. The node ignores non-innovative packets, and stores the innovative ones in the current generation. Then the intermediate node adds redundant packets depending on the loss rates of the links involved in the next hop. The same process is repeated at every intermediate node until the generation arrives at the destination. A re-encoding operation is similar to the encoding process. The re-encoded packet can also be expressed as a function of the native packets:

$$z_n = \sum_{m=1}^{K} f_{n,m} \cdot y_m = \sum_{m=1}^{K} f_{n,m} \left(\sum_{i=1}^{K} f_{m,i} \cdot x_i \right) = \sum_{i=1}^{K} \left(\sum_{m=1}^{K} f_{n,m} \cdot f_{m,i} \right) x_i \tag{4}$$

where $f_{n,m}$ are the coefficients randomly picked by the intermediate node, $(f_{n,1}, ..., f_{n,j}, ..., f_{n,K})$ is called the coding vector of packet y_m, m=1,…,M.

On the receiver side, for each packet received, the destination node checks whether the packet is linearly independent of the previously received packets. Once K innovative packets are received, the generation can be decoded. Otherwise, the whole generation will be discarded for the reason that we just consider an on-off decoding of generations. Eventually all generations in the video sequence with sufficient number of packets received will be decoded. The destination node receives coded generation packets $z=(z_1, z_2, \cdots, z_N)\,(N \geq K)$ and decodes them with matrix inversion:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix} = \begin{pmatrix} f_{11} & \cdots & f_{1K} \\ \vdots & \ddots & \vdots \\ f_{K1} & \cdots & f_{KK} \end{pmatrix}^{-1} \begin{pmatrix} z_1 \\ \vdots \\ z_K \end{pmatrix} \tag{5}$$

where x_i is the native packet after being decoded and z_i is the coded packet whose coding vector is $(f_{i,1}, ..., f_{i,j}, ..., f_{i,K})$.

Fig. 4 shows the framework of end-to-end streaming transmission with coded generations. The streaming data is encoded at the source node and transmitted to the destination node (client) via the intermediate nodes. The intermediate node receives the coded flow, re-encodes them and forwards them to the next hop. Finally, the coded data packets will be decoded at the destination node as soon as sufficient innovative packets are received.

The performance of JSNC is influenced by the generation size and the redundancy value as discussed in section 1. To maximizing video quality, both the length of generation and the degree of redundancy should be optimized.

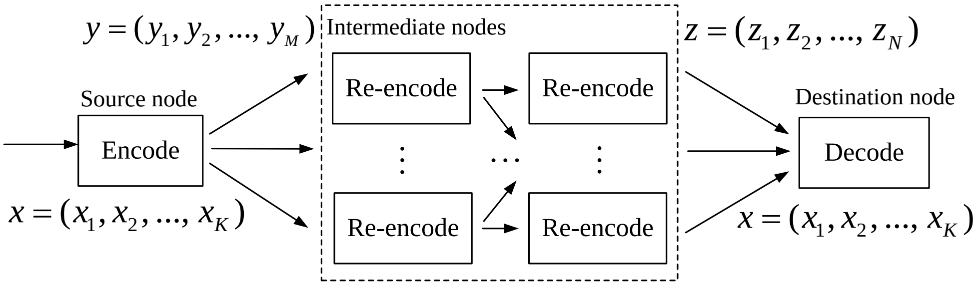


Fig. 4. The scheme of joint source/network coding for end-to-end streaming

4. JSNC Optimization

4.1 End-to-End Distortion Model

The problem we consider here is to maximize video quality at the client, that is to say, minimizing the total end-to-end distortion for the given network capacity and delay constraints. The end-to-end distortion can be generally calculated as the sum of the source distortion and the channel distortion [22]. The source distortion D_s is due to lossy encoding of the video sequence. Therefore D_s is mostly driven by the source rate R_s and the video content. The channel distortion D_σ is dependent on the packet loss (including losses due to delay) experienced in the network. Because we use the on-off decoding, the failure in decoding a generation will cause a relatively high distortion $(D_\sigma > D_s)$. Thus, D_σ can be considered as a constant and independent of D_s. The average decoding probability of each generation is $p_d (0 \le p_d \le 1)$. Here, we consider the expected end-to-end distortion, the expectation is taken with respect to the probability of decoding probability. The expected end-to-end distortion is expressed as:

$$E[D^l] = p_d \cdot E[D_s^l] + (1 - p_d) \cdot D_\sigma^l \tag{6}$$

where $E[D^l]$ is the expected end-to-end distortion for the l-th generation, $E[D_s^l]$ is the expected distortion when the generation is decoded correctly, D_σ^l is the distortion when the generation is decoded unsuccessfully.

The source distortion is mostly driven by the generation size K. The decodable probability of each generation is related to the generation size K as well as the network coding redundancy r. So, the problem of selecting the proper K and r becomes equivalent to minimizing the end-to-end distortion under constraints. Therefore, the total end-to-end distortion minimization problem can be formulated as the follows:

$$\begin{aligned} &\arg\min_{K,r} \sum_{l=1}^{L} E[D^l(K,r)] \\ &\text{s.t.} \quad 1)\ T_{rcv}^l \le T_{pb}^l \\ &\qquad\quad 2)\ \frac{K^l + r^l}{K^l} R_s \le R_c \end{aligned} \tag{7}$$

where L is the number of generations, T_{rcv}^{l} is the time needed for receiving the packets in the l-th generation, T_{pb}^{l} is the playout deadline of the l-th generation. R_c is the max-flow from the source to the receiving node. The constraint 1 in (7) requires that the packet transmission delay must be lower than its playout time. And the constraint 2 ensures that for each link, the flow rates cannot exceed the link's capacity.

The rate-distortion (*R-D*) model in [23] for video coding describes the distortion caused by the encoder.

$$D_s\left(R_s\right) = D_0 + \frac{\theta}{R_s - R_0} \tag{8}$$

where D_0 is the distortion offset, θ is the *R-D* factor and R_0 is the rate offset, all depending on the coding scheme and the content of the video sequence. They can also be estimated from empirical rate-distortion curves by training and curve matching.

The relationship between the generation size K and the source distortion with average rate per frame can be shown as the follows:

$$\begin{aligned} D_s^l\left(\frac{K^l}{L \cdot S_{GOP}^l}\right) &= D_0^l + \frac{\theta^l}{\frac{K^l \cdot S_{pkt}}{L \cdot S_{GOP}^l} - R_0^l} \\ &= D_0^l + \frac{\theta^{l\prime}}{\frac{K^l}{L \cdot S_{GOP}^l} - R_0^{l\prime}} \end{aligned} \tag{9}$$

where S_{GOP}^{l} is the number of frames in the l-th generation, S_{pkt} is the constant packet size, and $\theta^{l\prime}$ is obtained by θ^l / S_{pkt} and $R_0^{l\prime}$ is R_0^l / S_{pkt}.

According to [24], the impact of the generation size to decoding probability is negligible. There is no significant difference between the case when the generation size $K = 5$ and the case when $K = 100$. Actually, if the field size q is greater than 3, one extra packet is sufficient for having linearly independent packets. [25] gives the upper bound of the average number of coded packets that have to be received for decoding, which is:

$$K(1+\omega) = \min\left\{ K\frac{q}{q-1},\ K+1+\frac{1-q^{-K+1}}{q-1} \right\} \tag{10}$$

where $\omega > 0$ and $K(1+\omega)$ is slightly larger than K.

So, the probability that the receiver can decode correctly over N coded packets is given by:

$$p_d = \begin{cases} 0 & \text{if } N < K(1+\omega) \\ \sum_{j=K(1+\omega)}^{N} \binom{N}{j} (1-\varepsilon)^j \varepsilon^{N-j} & \text{if } N \geq K(1+\omega) \end{cases} \tag{11}$$

where ε is the actual packet loss probability using the random linear network coding scheme. $N=(K+r)(1-\varepsilon)$ is the average number of packets which can be received in each generation.

For video streaming with strict delay constraints, the delay introduced by the generation size should be taken into consideration. The larger the generation size, the longer the time needed to receive sufficient number of packets for decoding. The time needed for receiving the packets in each generation can be expressed as the follows:

$$T_{rcv}^{l}=\frac{K^{l}(1+\omega)\cdot S_{pkt}}{R_{c}} \tag{12}$$

4.2 Solution Algorithm

Let $\lambda^{l}\geq 0$ and $\eta^{l}\geq 0$ be the Lagrange multipliers for the delay and capacity constraints, respectively. The constraint function (7) can be converted into an unconstrained Lagrangian problem as the follows:

$$\begin{aligned}F(K,r,\lambda,\eta)&=\sum_{l=1}^{L}\left\{E[D^{l}]+\lambda^{l}(T_{rcv}^{l}-T_{pb}^{l})+\eta^{l}(\frac{K^{l}+r^{l}}{K^{l}}R_{s}-R_{c})\right\}\\&=\sum_{l=1}^{L}\left\{E\left[D_{0}^{l}+\frac{\theta^{l\prime}}{\frac{K^{l}}{L\cdot S_{GOP}^{l}}-R_{0}^{l\prime}}\right]p_{d}-(1-p_{d})D_{\sigma}^{l}\right\}\\&\quad+\sum_{l=1}^{L}\lambda^{l}\left(\frac{K^{l}(1+\omega)\cdot S_{pkt}}{R_{c}}-T_{pb}^{l}\right)+\sum_{l=1}^{L}\eta^{l}\left(\frac{K^{l}+r^{l}}{K^{l}}R_{s}-R_{c}\right)\end{aligned} \tag{13}$$

Since the objective function is differentiable with respect to the variables K^{l}, r^{l}, λ^{l} and η^{l}, this optimization formulation can be solved by gradient descent algorithm [26].

$$\begin{aligned}K^{l}(t+1)&=\left[K^{l}(t)+\alpha(t)\frac{\partial F(K^{l},r^{l},\lambda^{l},\eta^{l})}{\partial K^{l}}\right]^{+}\\r^{l}(t+1)&=\left[r^{l}(t)+\beta(t)\frac{\partial F(K^{l},r^{l},\lambda^{l},\eta^{l})}{\partial r^{l}}\right]^{+}\\\lambda^{l}(t+1)&=\left[\lambda^{l}(t)+\gamma(t)\frac{\partial F(K^{l},r^{l},\lambda^{l},\eta^{l})}{\partial \lambda^{l}}\right]^{+}\\\eta^{l}(t+1)&=\left[\eta^{l}(t)+\delta(t)\frac{\partial F(K^{l},r^{l},\lambda^{l},\eta^{l})}{\partial \eta^{l}}\right]^{+}\end{aligned} \tag{14}$$

where t denotes the iteration index, $\alpha(t)$, $\beta(t)$, $\gamma(t)$ and $\delta(t)$ are positive step sizes associate with K^{l}, r^{l}, λ^{l} and η^{l} respectively, and "+" denotes the projection on the nonnegative real numbers.

4.3 Overhead Analysis

1) Transmission overhead:

For a constant source rate, the smaller the packet size used for transmission, the more packets will be within a generation. For example, at the source rate of 200Kbps, when the packet size is changed from 1024 bytes to 512 bytes, the average number of packets per generation will be increased from 15 to 25. The benefit of having more packets within a generation is straightforward. The diversity in the network can be maximized and the coding efficiency of the intermediate nodes will also be increased by having more chances to mix up the packets. However, the gain obtained from the smaller packet is at the cost of higher overhead in transmission. When the packet size is smaller, the proportion of header information in the total transmission flow becomes larger.

In practical random linear network coding, both the generation ID and encoding vector are included in the packet header. To account for network coding overhead, the actual transmission rate after network coding R_{nc} can be shown as the follows:

$$R_{nc} = R_c \frac{S_{pkt}}{S_{pkt} + S_{hdr}} \tag{15}$$

where S_{hdr} is the overhead for packet headers used in RLNC.

2) Computational complexity for coding:

The benefit of using network coding comes at the cost of computational overhead for encoding and decoding. For RLNC, the computational complexity is closely related to the generation size *K*. The encoding operation is to multiply the original blocks by the randomly selected coefficients in the GF(2^8). So the encoding complexity is $O(K^2)$. Decoding the RLNC coded generation involves matrix inversion using Gaussian elimination, which has a complexity of $O(K^3)$. We can see that with the increase of the generation size, the complexity of encoding and decoding will increase. The more time is needed for encoding and decoding operations.

3) Node selectivity:

In fact, it is not necessary to perform network coding on every node, especially in the case of large scale heterogeneous network. The delay as well as the computational overhead in the system will grow with the increase of nodes participating in network coding. Some relay nodes may have limited computational capability and can not perform coding operation efficiently. So, it is more efficient to select a subset of important relay nodes to be the intermediate nodes for performing network coding in order to control the overhead and complexity of network coding, and to exploit efficiently the diversity in the network.

5. Performance Evaluation

5.1 Simulation Setup

We analyze the performance of the proposed JSNC optimization scheme in various network scenarios by simulation. We developed an event-driven network simulator with C/C++ to evaluate the proposed scheme. This approach gives us the flexibility to vary the network parameters and to implement the network coding scheme.

The network topology used in our simulation is the same as shown in **Fig. 1**. The source node streams a compressed media data through intermediate nodes R_1, R_2, R_3, R_4, R_5 and R_6, to the client node. The test video sequence *Foreman* is in QCIF resolution (176×144) at 30 fps (frames per second). We use JSVM Reference Software (version 9.19.7 [27]) to encode the sequence into an H.264/AVC stream with a GOP (Group of Pictures) size of 9 frames. The burst lossy patterns of the wireless links are generated by the GE model.

The packet loss rate on each link is in the range of 5% to 20%. The average burst length in GE model is equal to 9 packets. **Table 1** shows the average packet loss rate and the corresponding GE model parameters.

Table 1. Average packet loss rate and the corresponding GE model parameters

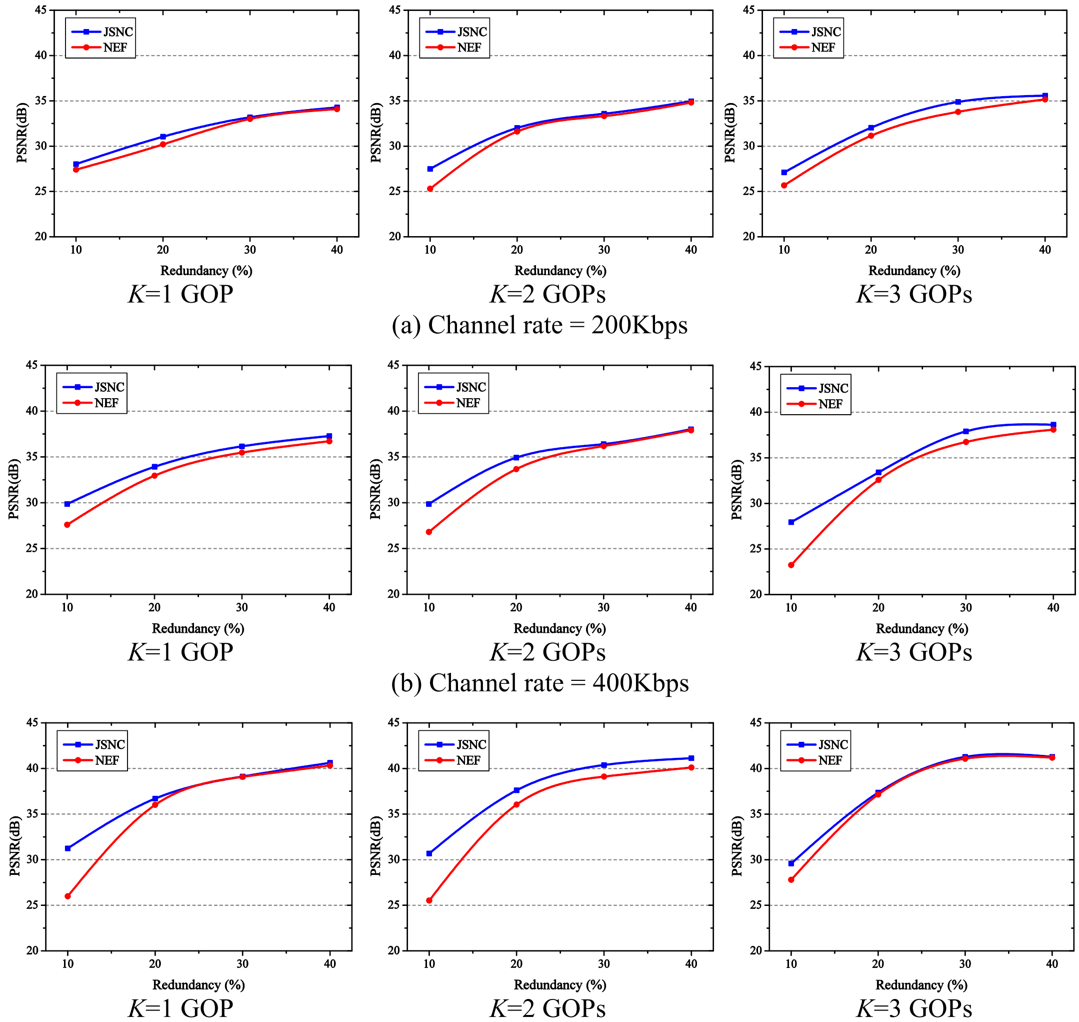
Average packet loss rate	**GE model parameters**	
	P_{gg}	P_{bb}
5%	0.9942	0.8889
10%	0.9877	0.8889
15%	0.9804	0.8889
20%	0.9722	0.8889

Assume the unit of generation size K is GOP and the packet size is 1024 bytes. Depending on the source coding rate, each GOP may contain different number of packets. For example, at the rate of 200Kbps, most GOPs involve 15 packets, at the rate of 400Kbps, 20 packets are more common, and at the rate of 800Kbps, usually 30 packets. We test the source coding rate of 200Kbps, 400Kbps, and 800Kbps, respectively. At each source rate, the generation size K is set to 1 GOP, 2 GOPs and 3 GOPs. 10%, 20%, 30% and 40% network coding redundancy are introduced to each of the three generation sizes, respectively. The performances of all three generation sizes with various packet sizes (i.e., 256 bytes, 512 bytes and 1024 bytes) are also evaluated. The video quality is measured as the Mean-Squared-Error (MSE) over all frames of the video sequence. We use the peak signal-to-noise ratio PSNR ($\mathrm{PSNR} = 10\log_{10}(255^2/\mathrm{MSE})$) to illustrate simulation results. All results are obtained by averaging over ten simulation runs.

5.2 Simulation Results

We first compare the performance of JSNC with NEF strategy that performs RS codes at the network nodes. The two schemes are tested using different source rates and generation sizes, and different levels of redundancy. Here, for NEF, the generation size K is defined as the number of video packets in the FEC block. The source node generates the FEC block with RS(N, K) code. If the received packets within a coded generation are less than K, the intermediate node just forward the received packets as usual because the node can not decode the block and reconstruct the original data. If the intermediate node received K or more packets, the node uses the RS code to reconstruct the original data, reproducing the redundant data. Then the re-encoded packets are sent to the next hop. However, the RS codes suffer from a large encoding and decoding computational complexity. The encoding complexity is $O(K^2)$ and the complexity decoding process is $O(K^3)$.

The results shown in **Fig. 5** indicate that the performance of the JSNC scheme is better than that of the NEF scheme. The reason is that, the time cost of decoding and re-encoding operations in NEF is much higher than that of JSNC scheme which just performs re-encoding at the intermediate nodes. So, more generations in the NEF scheme will not have enough packets received because of packet expiration.



K=1 GOP *K*=2 GOPs *K*=3 GOPs

(a) Channel rate = 200Kbps

K=1 GOP *K*=2 GOPs *K*=3 GOPs

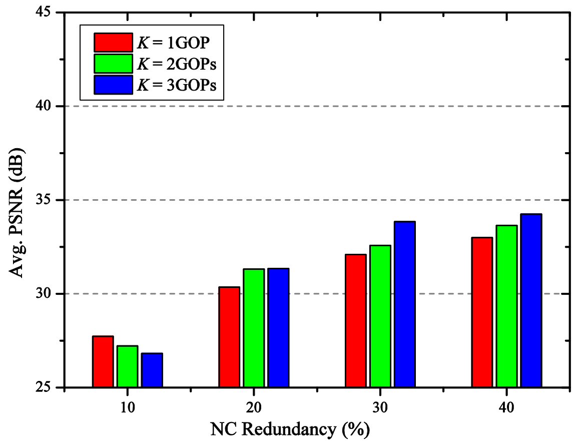
(b) Channel rate = 400Kbps

K=1 GOP *K*=2 GOPs *K*=3 GOPs

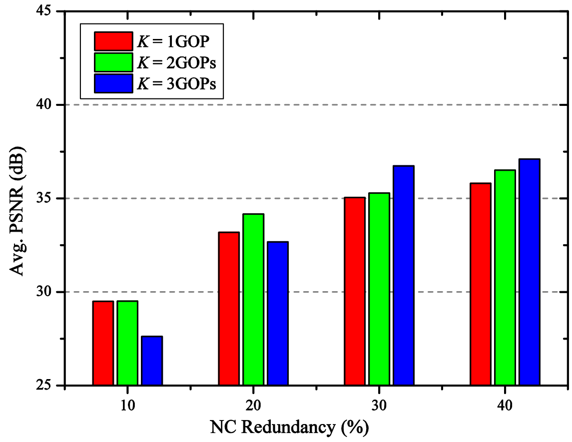
(c) Channel rate = 800Kbps

Fig. 5. Comparison of JSNC and NEF schemes

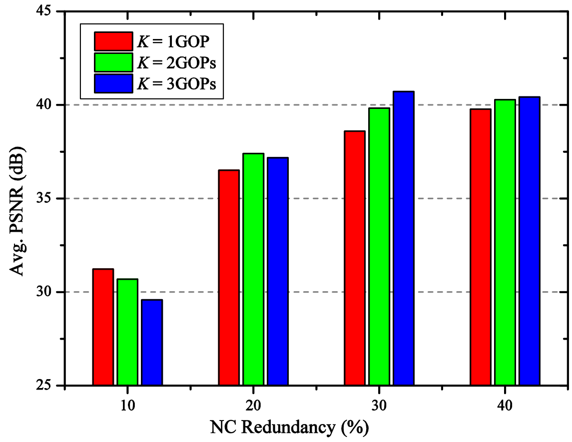
From **Fig. 6**, we can learn the influence of the generation size *K* to the PSNR in different simulation runs. We can see that when the network coding redundancy is low, the larger the generation size, the lower the streaming performance. The main reason for this is bursty loss characteristic of the channel and the on-off decoding of the generations. This means that adding 10% network coding redundancy may not be sufficient to protect the whole generation. When the generation size is 3 GOPs, bursty loss may lead to a situation that all three GOPs cannot be decoded. But if the generation size is 1 GOP, the distortion caused by the bursty loss may be limited to only one or two GOPs. With a higher network coding redundancy level, the video streaming performance becomes better when increasing the generation size. We can also see clearly in **Fig. 7** that the PSNR increases with the increase of the size *K* when adding sufficient (30%) redundancy. But one disadvantage of increasing the generation size is that it requires a larger buffer to store the coded generation and the playback delay will be increased. Also, as the generation size becomes larger, it involves a higher coding overhead.



(a) Channel rate: 200Kbps



(b) Channel rate: 400Kbps



(c) Channel rate: 800Kbps

Fig. 6. Average PSNR of JSNC with different generation sizes and NC redundancy levels

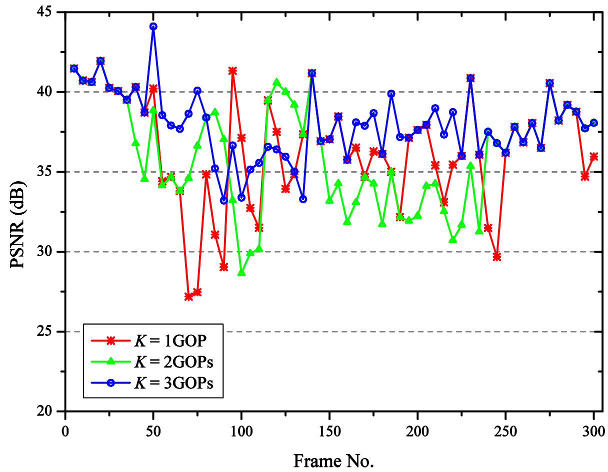


Fig. 7. Frame PSNR with different K (NC redundancy=30%, channel rate=400Kbps)

Fig. 6 also shows the influence of different redundancy levels to the average PSNR. It reveals that the average PSNR is affected not only by different generation sizes and different channel rates, but also by network coding redundancy. When the network coding redundancy is increased from 10% to 30%, the destination experiences an increasing PSNR. But when the network coding redundancy exceeds 30%, the PSNR gain is not so obvious and can even decrease, see the case when channel rate is 800Kbps and the generation size K is 3 GOPs. The reasons for this phenomenon are analyzed as the follows. First, the network channel quality will be aggravated when more redundant information transmitted over the network. The higher channel distortion may overwhelm the benefit obtained by adding redundancy for protection. Secondly, adding more redundancy means lower actual source rate. Therefore, unnecessary high network coding redundancy will augment the distortion caused by source encoding. From our simulation we learn that 30% redundancy might be close to the optimum. This fact can be seen more clearly from **Fig. 8**.

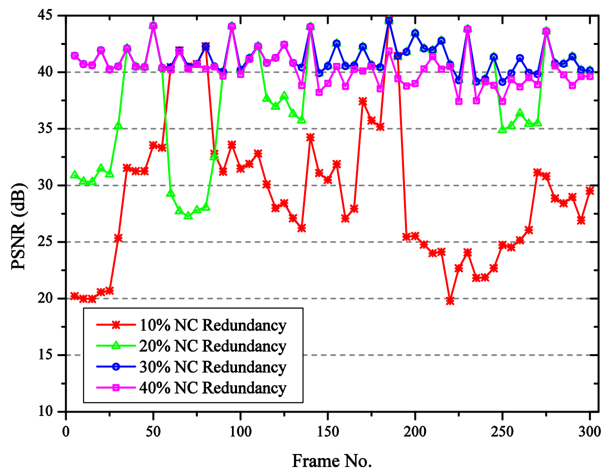
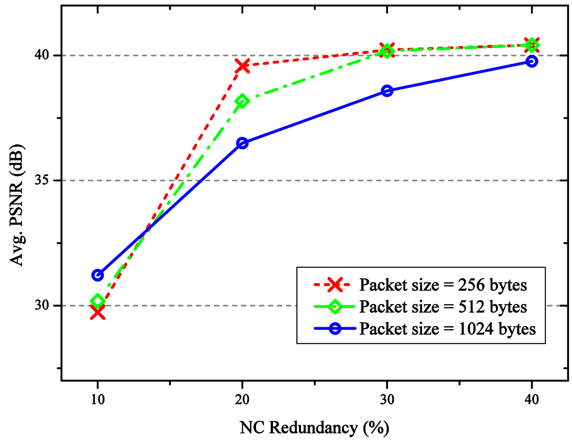
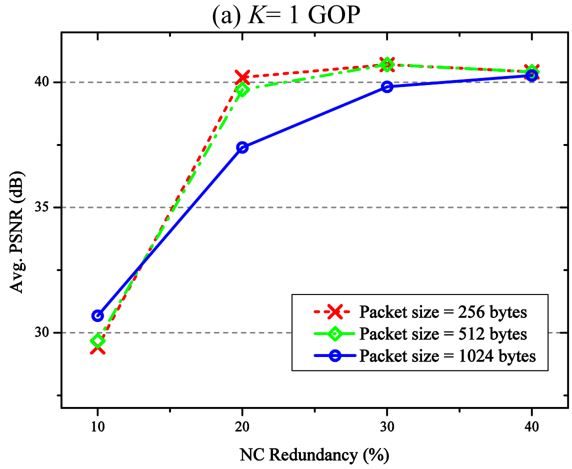


Fig. 8. Frame PSNR (channel rate=400Kbps, K=2 GOPs)

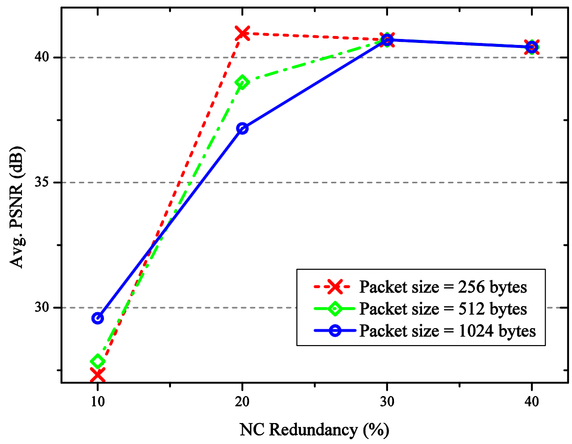
When decreasing the packet size, the number of packets in a generation with certain size will increase. **Fig. 9** shows the average PSNR of JSNC with different packet size settings when the channel rate is fixed at 800Kbps. The performance of the scheme with smaller packet size, such as 256 bytes, will be lower than that of other two settings when network coding redundancy level is low (10%). The reason is that the bursty loss will occur more frequently within the generation containing more packets. This results in less decodable generations because more generations cannot get sufficient number of packets for decoding. But on the other hand, when the network coding redundancy increases, a higher PSNR will be obtained in the case of using smaller packet size.



(a) *K*= 1 GOP



(b) *K* = 2 GOPs



(c) *K* = 3 GOPs

Fig. 9. Average PSNR of JSNC with different packet sizes at the channel rate 800Kbps

6. Conclusion

In this paper, we propose a scheme for optimizing the important parameters of the joint source/network coding (JSNC) for video streaming transmission over multi-hop wireless networks. The parameters considered in the optimization process include the source rate, the network coding generation size and the coding redundancy. In order to define the goal of optimizing the joint coding scheme, the end-to-end video distortion model is proposed and the computational and transmission overheads are analyzed. The relationship between different parameters as well as their effects to the video performance is investigated by analysis and simulation. Extensive simulations have been conducted to evaluate the performance of the JSNC scheme in different parameter settings. The simulation results show that with appropriate generation size and redundancy level, the JSNC scheme can achieve a higher video streaming quality.

As our future work, we will investigate the joint optimization scheme for the network coding generation with different priorities. The generation size and redundancy will be adapted to the dynamic network conditions. In addition, the different network topologies used for simulations will be taken into consideration.

References

[1] D. Li and J. Pan, "Performance evaluation of video streaming over multi-hop wireless local area networks," *IEEE Trans. Wireless Communications*, vol. 9, no. 1, pp. 338-347, Jan. 2010. Article (CrossRef Link)

[2] X. Cheng, P. Mohapatra, S.-J. Lee and S. Banerjee, "Performance evaluation of video streaming in multihop wireless mesh networks," in *Proc. of ACM NOSSDAV*, pp.57-62, May 28-30, 2008. Article (CrossRef Link)

[3] X. Qiu, H. Liu, D. Ghosal, B. Mukherjee, J. Benko, W. Li and R. Bajaj, "Enhancing the performance of video streaming in wireless mesh networks," *Wireless Pers. Commun.*, vol. 56, no.3, pp. 535-557, Apr. 2011. Article (CrossRef Link)

[4] R. Ahlswede, N. Cai, R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204-1216, July, 2000. Article (CrossRef Link)

[5] E. Magli, M. Wang, P. Frossard and A. Markopoulou, "Network coding meets multimedia: a review," *IEEE Trans. Multimedia*, vol. PP, no. 99, pp. 1-18, Jan. 2013. Article (CrossRef Link)

[6] K. Nguyen, T. Nguyen and S. Cheung, "Video streaming with network coding," *Springer J. of Signal Process. Systems*, vol. 59, no. 3, pp. 319–333, 2010. Article (CrossRef Link)

[7] P. A. Chou, Y. Wu and K. Jain, "Practical network coding," in *Proc. of 43rd Allerton Conf. on Communication, Control and Computing*, pp. 1-10, Oct. 1-3, 2003.

[8] R. Gowaikar, A. F. Dana, B. Hassibi and M. Effros, "A practical scheme for wireless network operation," *IEEE Trans. Communications*, vol. 55, no. 3, pp. 463-476, Mar., 2007. Article (CrossRef Link)

[9] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 713-728, Jun. 2009. Article (CrossRef Link)

[10] D. Nguyen, T. Nguyen and X. Yang, "Joint network coding and scheduling for media streaming over multiuser wireless networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1086-1098, Mar. 2011. Article (CrossRef Link)

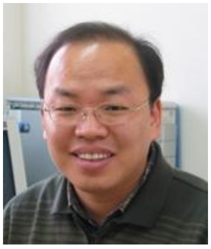
[11] N. Thomos and P. Frossard, "Network coding of rateless video in streaming overlays," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 20, no. 12, pp. 1834-1847, Dec. 2010. Article (CrossRef Link)

[12] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Information. Theory*, vol. 52, no. 10, pp. 4413-4430, Oct. 2006. Article (CrossRef Link)
[13] C. Gkantsidis, W. J. Hu, P. Key, B. Radunovic, P. Rodriguez and S. Gheorghiu, "Multipath code casting for wireless mesh networks," in *Proc. of ACM CoNEXT*, pp. 1-12, Dec. 10-13, 2007. Article (CrossRef Link)
[14] J. L. Le, J. C. Lui and D. M. Chiu, "On the performance bounds of practical wireless network coding," *IEEE Trans. Mobile Computing*, vol. 9, no. 8, pp. 1134-1146, Aug. 2010. Article (CrossRef Link)
[15] H. Seferoglu and A. Markopoulou, "I^2NC: intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. of IEEE INFOCOM*, pp. 1035-1043, Apr. 10-15, 2011. Article (CrossRef Link)
[16] G. Bhat and J. McNair, "Analyzing effect of generation size in intra-session network coding for multiple flows of TCP traffic," in *Proc. of IEEE MILCOM*, pp. 729-734, Nov. 7-10, 2011. Article (CrossRef Link)
[17] Z. Li, D. Zeng, S. Guo, S. Lu, D. Chen and W. Zhuang, "On the throughput of feedbackless segmented network coding in delay tolerant networks," *IEEE Wireless Communications Letters*, vol. 1, no. 2, pp. 93-96, Apr. 2012. Article (CrossRef Link)
[18] Y. Li, P. Vingelmann, M. Pedersen and E. Soljanin, "Round-robin streaming with generations," in *Proc. of NetCod*, pp. 143-148, Jun. 29-30, 2012. Article (CrossRef Link)
[19] M. Wu, S. Karande and H. Radha, "Network-embedded FEC for optimum throughput of multicast packet video," *Signal Processing: Image Communication*, vol. 20, no. 8, pp. 728-742, 2005. Article (CrossRef Link)
[20] J. Kim, R. M. Mersereau and Y. Altunbasak, "Distributed video streaming using multiple description coding and unequal error protection," *IEEE Trans. Image Processing*, vol. 14, no. 7, pp. 849-861, Jul. 2005. Article (CrossRef Link)
[21] T. Ho and S. Lun, *Network Coding an Introduction*. Cambridge University Press, UK, 2008. Article (CrossRef Link)
[22] L. Zhou, X. Wang, W. Tu, G. Mutean and B. Geller, "Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks," *IEEE J. on Sel. Areas in Commun.*, vol. 28, no. 3, pp. 409-419, Apr. 2010. Article (CrossRef Link)
[23] K. Stuhlmuller, N. Farber, M. Link and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012-1032, Jun. 2000. Article (CrossRef Link)
[24] O. Trullols-Cruces, J. M. Barcelo-Ordinas and M. Fiore, "Exact decoding probability under random linear network coding," *IEEE Communications Letters*, vol. 15, no. 1, pp. 67-69, Jan. 2011. Article (CrossRef Link)
[25] D. E. Lucani, M. Medard and M. Stojanovic, "Random linear network coding for time-division duplexing: Field size considerations," in *Proc. of IEEE GLOBECOM*, pp. 1-6, Nov. 30-Dec. 4, 2009. Article (CrossRef Link)
[26] S. Shakkottai and R. Srikant, *Network Optimization and Control.* Foundations and Trends in Networking, vol. 2, no. 3. Now Publishers, Netherlands, 2007. Article (CrossRef Link)
[27] J. Reichel, H. Schwarz, M. Wien and J. Vieron, *Joint Scalable Video Model, version 9.19.7*, Joint Video Team (JVT) of ISO-IEC MPEG & ITU-T VCEG, Jan. 2010.

Huali Cui received the M.S. degree in Computer Architecture from Xi'an Jiaotong University, China, in 2005. He is currently pursuing the Ph.D. degree from the Department of Computer Science & Technology, Xi'an Jiaotong University. His research interests include network measurement, network coding and wireless multimedia communications.

Depei Qian received the B.S. degree in Computer Science from Xi'an Jiaotong University in 1977, and the M.S. degree from Texas State University in 1984. Since 1992, he became a full professor in the Department of Computer Science & Technology, Xi'an Jiaotong University. His current research interests include computer architecture, high performance computing, mobile computing, sensor networks and distributed computing system.

Xingjun Zhang received his Ph.D degree in Computer Architecture from Xi'an Jiaotong University. From 1999 to 2005, he was Lecturer, Associate Professor in the Department of Computer Science & Technology of Xi'an Jiaotong University. From Feb. 2006 to Jan. 2009, he was a Research Fellow in the Department of Electronic Engineering of Aston University, United Kingdom. He is currently an associate professor in the Department of Computer Science & Technology of Xi'an Jiaotong University. His interests include high performance computer architecture, the new technologies of the computer networks and high performance computing.



Ilsun You received his M.S. and Ph.D. degrees in Computer Science from Dankook University, Seoul, Korea in 1997 and 2002, respectively. Since March 2005, he has been an Assistant Professor in the School of Information Science at the Korean Bible University, South Korea. His main research interests include network security and authentication. He is a member of the IEICE, KIISC and KSII.

Xiaoshe Dong received the B.Eng. degree in Computer Hardware from Xi'an Jiaotong University, Xi'an, China, in 1985, and the M.S. and Ph.D. degrees in Computer Architecture, both from Keio University, Tokyo, Japan, in 1996 and 1999 respectively. He was Lecturer during the period 1987-1994 and Associate Professor during 1999–2003 in the Department of Computer Science & Engineering of Xi'an Jiaotong University, where he has been the Full Professor with the Department of Computer Science & Engineering from 2003. His interests include high performance computer architecture and Grid computing.