Joint Source-Network Coding Optimization for Video Streaming over Wireless Multi-hop Networks

Huali Cui*, Depei Qian*†, Xingjun Zhang*, Cuiping Jing* and Yifei Sun*
*Department of Computer Science, Xi'an Jiaotong University, Xi'an, China
†Sino-German Joint Software Institute, School of Computer Science, BeiHang University, Beijing, China
Email: hualic@stu.xjtu.edu.cn

***Abstract*—For the reason of unreliable and shared media, supporting video streaming over wireless multi-hop networks faces greater technical challenges. In this paper, we investigate the optimization issue and propose a joint source-network coding scheme, which segments the streaming source into generations so as to maximize the video streaming quality. The factors influenced by the size of generation include the source rate, the efficiency of coding and the decoding delay. At the source node, the faster the source rate, the more packets generated. At the intermediate nodes, the number of packets transmitted into the network is decided by the network coding strategies. The experiment results indicate that with appropriate generation size, the joint source-network coding scheme can enhance the performance of video streaming over the wireless multi-hop networks.**

I. Introduction

Recently, more and more attention has been devoted to applications such as peer-to-peer (P2P) and wireless mesh networks, in which peer nodes can self-organize in order to exploit the network infrastructure more efficiently. The network diversity can be used to enhance the quality of video streaming transmission by increasing bandwidth throughput and reducing network delay and packet loss [1]. However, this also requires the development of appropriate streaming mechanisms, so that the network resources can be fully exploited. Network coding has emerged as a solution to efficiently utilize the network bandwidth [2]. Besides throughput benefits, network coding has the capability of error resilience and facilitates scheduling. When employing network coding for media communication over wireless multi-hop networks, a critical network coding parameter is the generation size which also affects the source coding efficiency.

Ideally, network coding should be implemented across the entire flow. However, this is not practical. Generally, one video streaming is divided into several generations, and only packets from the same generation are combined together. Generation coded separately has multiple advantages such as limiting the overhead of coding and decoding, reducing the state at the intermediate nodes. For practical reasons, many factors affect the generation size such as the complexity and performance of decoding, and header overhead for storing the coefficient vectors [3]. From the view point of throughput, the authors analyzed and compared the performance of generation sizes with 8, 12, 16, 32, 64, and 128 packets [4] [5]. In [6] authors analyze the performance of "encoding number", i.e., the number of packets that can be encoded by a coding node in each transmission, and provide an upper bound on the encoding number for the general coding approach. The work in [6] tries to answer the question of how many packets we can encode. But the work in [4], [5] and [6] do not tell us how many packets should be encoded to satisfy various applications requirement, such as maximizing the end-to-end quality of video transmission. H. Seferoglu *et al.* also investigated this issue. The strategy they proposed is determining the generation size according to the application requirements (e.g., media transmission) for UDP, or setting the generation size to equal to the TCP congestion window for TCP [7].

From the perspective of source coding, the number of packets in a generation is changing with the source coding rate. So it is necessary to joint consideration of video source coding, network coding, and network infrastructure when supporting video streaming. A fine-grain adaptive forward error correction (FGA-FEC) coding scheme for scalable video bit-stream is proposed in [8]. Both embedded source bit-stream and the error-control codes are adapted at block level in intermediate overlay nodes to satisfy heterogeneous users with both different video quality preferences and various network connections. The limitation of FEC is that it is employed on an end-to-end basis and the network diversity can not be fully utilized.

In this paper, we propose a joint source-network coding scheme, and investigate the optimization issues of how to segment one source into generations. We analyze the factors affected by the size of generation, such as the decoding delay, the diversity of packets in the network and the video distortion. The purpose of our study is to determine the number of source packets that we should put in each generation and the number of network coding packets to be generated in the network in order to maximize the end-to-end video quality.

The rest of the paper is organized as the follows. Section II presents some background knowledge on scalable video coding and multipath network coding. Section III presents the formulation and analyzes the the overhead. Experiments by packet-level simulation and the results are presented in Section IV. Section V concludes the paper.

II. Background

A. H.264/SVC Video

The Scalable Video Coding (SVC) [9] is an extension of the H.264/AVC standard, which allows efficient, standard-based

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

scalability of temporal, spatial, and quality resolution of a decoded video signal through adaptation of the bit stream.

Spatial scalability and temporal scalability describe cases in which subsets of the bit stream represent the source content with a reduced picture size (spatial resolution) or frame rate (temporal resolution), respectively. With quality scalability, the substream provides the same spatial-temporal resolution as the complete bit stream, but with a lower fidelity-where fidelity is often informally referred to as signal-to-noise ratio (SNR). Quality scalability is also commonly referred to as fidelity or SNR scalability.

SVC generates a base layer (BL) and one or more enhancement layers (ELs). The BL is a plain H.264/MPEG4-AVC bit-stream ensuring backward compatibility to existing receivers and guarantees basic display quality. Scalability within SVC is a functionality that allows the removal of parts of the bit-stream while achieving a reasonable coding efficiency of the decoded video at reduced temporal, SNR or spatial resolution. Each enhancement layer improves the video quality. But without the BL, video frames cannot be reconstructed sufficiently.

B. Multipath Network Coding

In contrast to the traditional store and forward transmission where, network nodes store and forward information without modification, the network coding solution allows network nodes to store, encode data before forwarding them to the outgoing links.

In network coding for a single communication session, i.e., unicast communication to one sink node or multicast of common information to multiple sink nodes, the data is typically divided into "generations". Coded transmissions deliver data within the same generation. This type of coding is called intra-session network coding [10], since we only code information symbols that will be decoded by the same set of sink nodes. Each node in the intra-session network coding, form its outputs as random linear combinations of its inputs. Each sink node can decode once it has received enough independent linear combinations of the source data. Intra-session coding can add redundancy to a flow (by adding linear combinations of packets) and thus improve reliability in the presence of loss.

Multipath network coding (MNC), first implemented in [4], is designed for long lasting unicast sessions in lossy wireless networks. In MNC, the source node first divides the original data file into generations. It then continuously gets packet streams from a generation using random linear code (RLC). Coded packets stream through multiple paths towards the destination. Intermediate forwarders can refresh the packet streams by re-encoding existing packets and transmitting the coded packets to downstream nodes. Once a sufficient number of packets accumulate at the destination, the original group of data blocks can be recovered.

III. Problem Formulation

The notation used in this section is given in Table I.

TABLE I
NOTATION

Symbol	Definition
s	video sessions in the network
N	number of generations in a video streaming
K	number of packets per original generation
R	the video encoding rate at source
R_{nc}	video transmission rate after network coding
π	policy for transmission a single generation
$C(\pi)$	expected cost of generations under π
$D(\pi)$	expected distortion over entire flow under π
D_0	distortion of entire flow if no generation can be decoded
G_{pkt}	the size of packets to be delivered
G_{hdr}	the size of packet headers used for network coding
ϵ	packet loss probability after network coding

A. System coding model

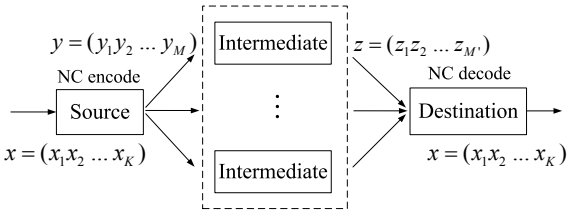

Fig. 1. The encoding and decoding process of streaming

At the source node, the raw video sequence is compressed into streaming s with SVC scheme. Packets in flow s are divided into generations with size K, where K may vary from one generation to another. The source node adds parity packets by creates a random linear combination of the K native packets within the generation. The number of parity packets depending on the loss rates of the links involved in this hop via intra-session network coding. Then the source node sends the coded M packets of the generations until the end and then proceeds to the next generation.

When an intermediate node hears a packet, it firstly checks whether the packet is an innovative packet. A packet is innovative if it is linearly independent of the packets that the node has previously received from this generation. The node ignores non-innovative packets, and stores the innovative packets it receives from the current generation. Then the node adds parity packets which also depending on the loss rates of the links involved in this hop. The same process is repeated at every intermediate node until the generation arrives at the destination.

At the receiver side, for each packet it receives, the destination node checks whether the packet is linearly independent of the previously received packets. Once the destination receives K innovative packets, it can decode the whole generation. At last the video sequence can be decoded from the recovered streaming.

Let $x = (x_1, x_2, \cdots, x_K)$ denote a original generation packets. The source node encodes generation x into $y = (y_1, y_2, \cdots, y_M)$. The network coding packets are transmitted

on the established paths and intermediate nodes. The intermediate node encode the received network coding packets within a generation and then send them to the next hop. Destination nodes receives network coding generation packets $z = (z_1, z_2, \cdots, z_{M'})$ and decode them with decoding matrices. Fig. 1 shows the encoding and decoding process for a streaming generation.

B. Video distortion model

Decoded video quality at the destination is mainly affected by two factors: encoder compression distortion D_{enc} at the source and distortion due to packet loss or late arrivals D_{loss}. Here we regard the late arrival packets the same as lost. So the total end-to-end distortion D_{dec} at the receiver can be expressed as: $D_{dec} = D_{enc} + D_{loss}$.

Let π_n be the transmission policy for generation $n \in \{1, \cdots, N\}$ and the transmission policies for all generations in the flow are $\pi = (\pi_1, \cdots, \pi_N)$. Any given policy will induce some degree of distortion and transmission cost for the entire flow. The transmission cost is the sum of the transmission costs of each generation in the flow. In turn, the expected transmission cost of each generation $n \in \{1, \cdots, N\}$ is the cost per byte in transmitting the generation, $\rho(\pi_n)$ times the generation size in bytes, B_n . That is, we have the cost for the expected transmission:

$$C(\pi) = \sum_n B_n \rho(\pi_n) \tag{1}$$

For a given generation, it is useful to the reconstructed video at the receiving peer only if the number of packets received on time is more than *K*. If a generation is decodable by the destination on time, then the reconstruction error is reduced by the quantity Δd_n ($0 < \Delta d_n < 1$, $\sum_n \Delta d_n = 1$), otherwise the reconstruction error is not reduced. Hence the total reduction in reconstruction error for the flow is $\sum_{m \le n} \Delta d_n \prod_{m \le n} (1 - \epsilon(\pi_m))$. The distortion can be obtained by subtracting this quantity from the reconstruction error for the flow if no data units are received. Consequently, the expected distortion of a reconstructed video sequence at a destination can be defined as:

$$D(\pi) = D_0 - \sum_{m \le n} \Delta D_l \prod_{m \le n} (1 - \epsilon(\pi_m)) \tag{2}$$

We can solve the optimization problem by finding the policy that minimizes the expected Lagrangian [11]:

$$\begin{aligned} J(\pi) &= D(\pi) + \lambda C(\pi) \\ &= D_0 + \sum_n \left[\Delta D_n \left(-\prod_{m \le n} (1 - \epsilon(\pi_m)) \right) + \lambda B_n \rho(\pi_m) \right] \end{aligned} \tag{3}$$

C. Remarks on overhead

1) Transmission overhead: Random linear network coding may require the use of packet headers to identify generations and encoding vectors for each packet; we augment the expression above to account for this packet overhead. To account for network coding overhead, we compute the actual transmission rate of network coding packets:

$$R_{nc} = R \frac{G_{pkt}}{G_{pkt} + G_{hdr}} \tag{4}$$

2) Delay overhead: Firstly, to recover original packets, it is necessary for the destination to receive the number of linear combinations at least as the number of original packets in the generation. The more packets coded, the longer the time for receiving the sufficient number of packets. Secondly, the encoding and decoding time also increasing with the increase of the generation size. So, the playback of the video streaming should be taken into consideration when setting the generation size.

IV. Performance Evaluation

We developed our own event-based network simulator with C/C++ to evaluate the proposed scheme. This gives us the flexibility to set network parameters and implement the network coding scheme. We set up a network as in Fig. 2 in which a source *S* streams a media sequence through intermediate nodes R_1, R_2, R_3 and R_4, to the destination *D*. The packet loss probability is marked on each link.

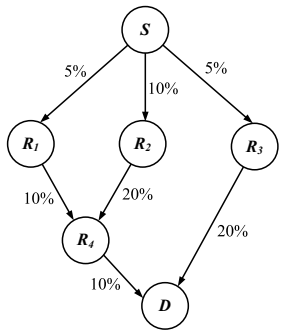

Fig. 2. Simulation network topology

For video encoding, we use the Joint Scalable Video Model (JSVM) to encode the QCIF *Foreman* sequence consisting of 300 frames distributed in I, P and B frames. *Foreman* is coded at the rate of 200Kbps, 400Kbps, 800Kbps, respectively, and the frame rate is 30 frames per second with a GoP (Group of Pictures) size of 9 frames. The video package size is 1024 bytes.

For random network coding, the base arithmetic field is a Galois Field GF (size of 2^8) containing 256 elements. This field size has been shown that this guarantees high symbol diversity and low probability of building duplicate packets.

We use a two-state Markov model (Gilbert model) to simulate the bursty packet loss behavior of the wireless network [12]. The two states of this model are denoted as *G* (Good) and *B* (Bad). In state *G*, packets are received correctly and timely, whereas, in state *B*, packets are assumed to be lost. Let P_{gg} denotes the probability of self-transition in the good state and P_{bb} denotes the self-transition probability in the bad state. So the stationary probability P_G that a link is in the *G*

state and the stationary probability P_B of a link in the *B* state can be given as follows:

$$P_G = \frac{1 - P_{bb}}{2 - P_{bb} - P_{gg}}, \quad P_B = 1 - P_G \tag{5}$$

The average length of burst errors $L_B = 1/(1 - P_{bb})$. In the experiment, we set the burst length increasing with the average packet loss rate.

Assume a generation is a GoP. Depending on the source coding rate, a generation may contain different number of packets. For example, at rate of 200Kbps, most of GoPs involve 15 packets, at rate of 400Kbps, 20 packets are more common, and at rate of 800Kbps, most generations contain 30 packets. We test the source rate at 200Kbps, 400Kbps and 800Kbps respectively. At each source rate, the generation size is set as 1 GoP, 2 GoPs and 3 GoPs. 20%, 30% and 40% network coding redundancy are added respectively to all generations of these three sizes. All results are obtained by averaging over ten simulations. Fig. 3 shows the comparison of average peak signal-to-noise ratio (PSNR) at different transmission schemes. When the source rate is set to 400Kbps and the generation size set as 2 GoPs, the PSNR curves with different network coding redundancy are shown in Fig. 4. When adding 30% network coding redundancy and the source rate is 800Kbps, the PSNR curves with different generation sizes are shown in Fig. 5. When set the generation to contain 2 GoPs and the network coding redundancy set as 40%, the PSNR curves with different source rates are shown in Fig. 6.

From Fig. 3 we can see that when more redundancy is added, the destination will receive more packets, so can get higher PSNR value. This point is straightforward. The PSNR is improved a lot when the redundancy increase from 20% to 30%. But when the network coding redundancy changes from 30% to 40%, the gain is not so obvious. This is because that 30% redundancy is large enough which enables the destination to receive almost all packets from the source, it is not necessary to further increase the redundancy. This can be seen obviously from Fig. 4.

With the generation size increasing, the PSNR is also enhanced. Fig. 3 shows this effect. We can see that the larger the generation size, the better the video performance. This can also be seen clearly from Fig. 5. But one disadvantage of increasing the generation size is that it needs a larger buffer to store the coded generation and the playback delay also increased. As the generation size becomes larger, it involves higher coding overhead.

Fig. 3 shows that the source rate changes from 200Kbps to 400Kbps and 800Kbps, the number of packets in the GoP increases accordingly. This also can be seen from Fig. 6. For different source rate, both generation size and redundancy have nearly the same effect. This verifies the first two conclusions.

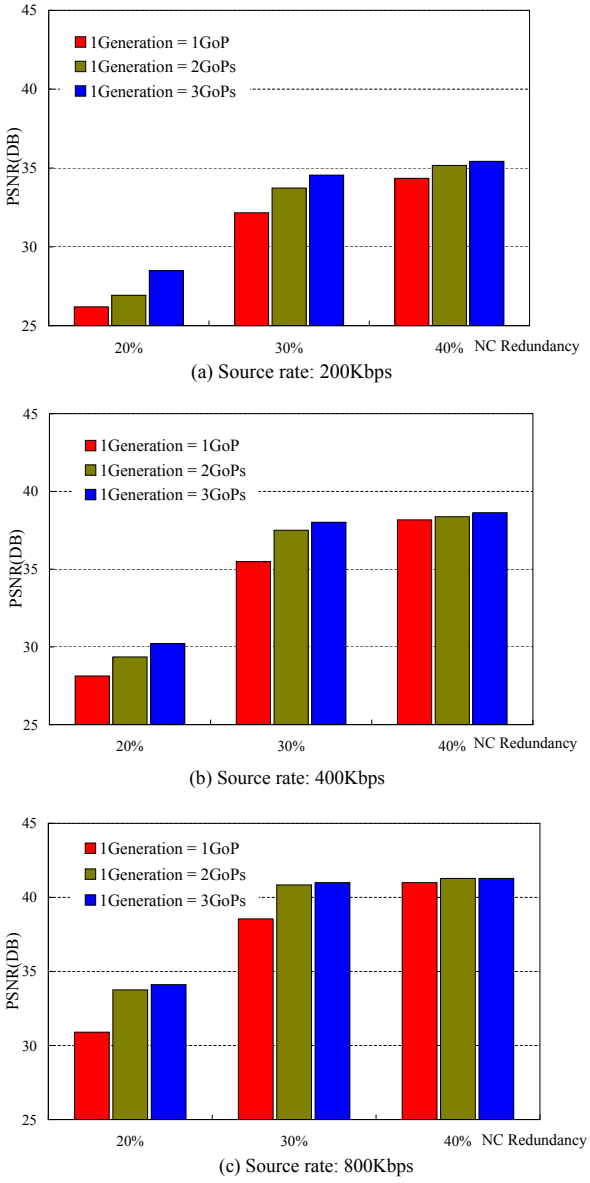

Fig. 3. Average PSNR at different transmission scheme

V. Conclusion

In this paper, we studied the problem of joint source-network coding optimization for video transmission over multi-hop wireless networks. The coding system and video distortion model are proposed and the overhead of transmission and delay are analyzed. The effects of different source rate, different generation size and different network coding redundancy are investigated from the perspective of video quality. We evaluate the performance of different strategies by extensive experiments. The simulation results show that with appropriate generation size, the joint source-network coding scheme can get better video quality performance.

Acknowledgment

This work is supported by National Key Technologies R&D Program (Grant No. 2011BAH04B03) and Joint Chinese-Italian Research Infrastructure for Cultural Heritage (Grant No. 2009DFA12110).

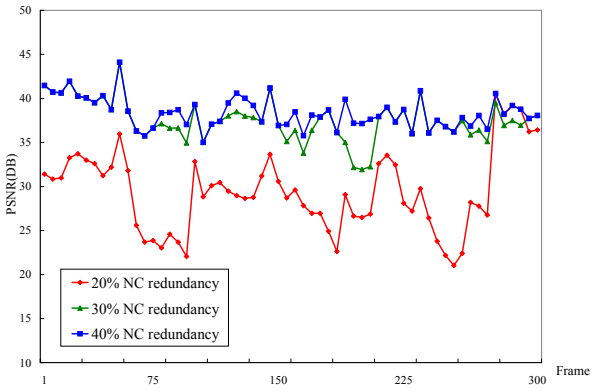

Fig. 4. Frame PSNR (source rate = 400Kbps, generation size = 2 GoPs)

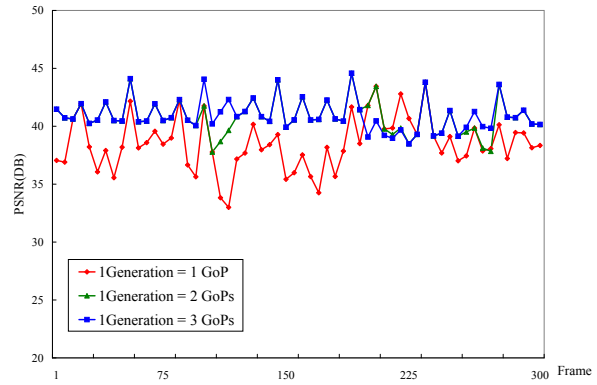

Fig. 5. Frame PSNR (NC redundancy = 30%, source rate = 800Kbps)

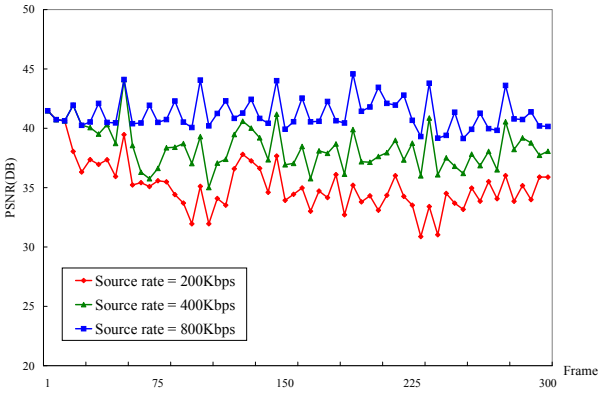

Fig. 6. Frame PSNR (generation size = 2 GoPs, NC redundancy = 40%)

REFERENCES

[1] P. Frossard, J. C. Martin, and M. R. Civanlar, "Media streaming with network diversity," *Proceedings of the IEEE*, vol. 96, pp. 39–53, Jan. 2008.

[2] R. Ahlswede, N. Cai, R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

[3] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, and S. Gheorghiu, "Multipath code casting for wireless mesh networks," in *Proc.of ACM CoNEXT*, New York, NY, Dec. 2007.

[4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc.of ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.

[5] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proc.of ACM SIGCOMM*, Seattle, Washington, Aug. 2008.

[6] J. Le, J. C. Lui, and D. Chiu, "On the performance bounds of practical wireless network coding," *IEEE Trans. on Mobile Computing*, vol. 9, pp. 1134–1146, Aug. 2010.

[7] H. Seferoglu and A. Markopoulou, "I^{2}NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. of IEEE Infocom*, Shanghai, China, Jul. 2011.

[8] Y. Shan, I. V. Bajic, J. W. Woods, and S. Kalyanaraman, "Scalable video streaming with fine-grain adaptive forward error correction," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 19, pp. 1302–1314, Sept. 2009.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 17, pp. 1103–1120, Sept. 2007.

[10] T. Ho, M. Mdard, R. Koetter, D. Karger, M. Effros, and J. S. et al., "A random linear network coding approach to multicast," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 4413–4430, Oct. 2006.

[11] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. on Multimedia*, vol. 8, pp. 390–404, Apr. 2006.

[12] J. Kim, R. M. Mersereau, Y. Altunbasak, and S. Member, "Distributed video streaming using multiple description coding and unequal error protection," *IEEE Trans. on Image Processing*, vol. 14, pp. 849–861, Jul. 2005.