Video Transmission Using Network Coding

Francisco de Asís López-Fuentes and Cesar Cabrera-Medina

Department of Information Technology, Autonomous Metropolitan University, México

Abstract—Network coding is a new technique in the field of information theory and coding theory. This emerging technique offers great benefits in the communication networks such as increased throughput and energy savings. In this paper, we evaluate network coding for video transmission scenarios. In these scenarios, the source nodes encode the video packets, while the intermediate nodes implement network coding before forwarding the encoded packets to the end nodes. Finally, the end nodes decode the received packets in order to recovery the original video. H.264/MPEG-4 AVC is used as the video compression standard in this work. We have used the network simulator (NS-2) for our simulations. Our results show that network coding improves the system throughput, reduces the packet loss and improves the video quality in the end nodes.

Keywords—communication networks, information theory, multicast, network coding video streaming.

1. Introduction

Recently, a large amount of traffic is generated on the Internet. Mainly, this traffic is generated by multimedia applications such as video streaming, video download or video-conferencing. Additionally, many Internet users use this infrastructure to share and distribute data, music, videos and photos. Many of these applications involve systems with multiple senders. Multiple sources help to alleviate the unpredictability congestion in the Internet and it has been proposed as an alternative to edge streaming to provide smooth video delivery [1], [2].

The great traffic generated by all these applications, has created problems on the communications networks, such as delays and a bad quality in the received files. Due this situation, new techniques need to be introduced to deal with these problems. For example, video multicast is a network service through which a video is streamed to a group of interested receivers. Video streaming typically requires high data rate, low-latency, or high throughput in order to offer video quality to the viewers. However, traditional transmission techniques, based on a method known as store-and-forward, are not capable of supporting such applications. Store-and-forward is a technique in which data packets received from an input link of an intermediate node are stored and a copy is forwarded to the next node via an output link [3]. The intermediate node verifies the integrity of the message before forwarding it. This traditional transmission technique introduces a delay at the input to each link along the packet's route, which can affect the video quality to

the viewers. Our proposed work aims to give a solution to some problem in this area.

Network coding was introduced by Ahlswede *et al.* [4] as a new technique for the diffusion of the information in the field of information theory. This new technique allows that the intermediates nodes encode the received packets on the intermediate nodes, for immediately forwarding the encoded packets to the end nodes. This fact has generated a considerable debate between the traditional technique of storing and forwarding against the network coding technique. Tuninety *et al.* [5] study both schemes and they found that network coding improves the throughput, scalability and robustness of a system. Network coding has impacted in the following areas: Mesh, VANETs, MANETs, ad-hoc, sensor networks, security issues and content distribution. Several benefits in the communication networks by using network coding have been reported in the literature.

This paper evaluates the transmission of video streaming for multicast schemes using network coding. Our network coding implementation realized in the intermediate nodes is inspired by the model introduced by Chou *et al.* in [6]. Our simulations use video sequences in the CIF (Common Intermediate Format) video format. CIF defines a video sequence with a resolution of 352×288. These video sequences are compressed using the standard H.264/MPEG-4AVC. The communication between all nodes is established via the User Datagram Protocol (UDP). We compare the performance of a multi-source system using network coding versus a traditional multi-source system in terms of packets loss, throughput and video quality. To this end, we evaluate video transmission sessions with network coding for two strategies. In the first strategy, video sequences with different size are transmitted by each source, while in the second strategy all video sequences transmitted by all sources have the same size. The obtained results show that network coding improves the system throughput, decreases the packet loss and improve the video quality in the end nodes.

The rest of this paper is organized as follows. Section 2 presents the main idea of network coding. Then, we give an overview about the related work in Section 3. In Section 4 we describe our model based on network coding. The performance of our proposed model is evaluated in Section 5. This paper concludes in Section 6.

In this paper we present an extended version of our other work [7]. Specifically, the following new material has been
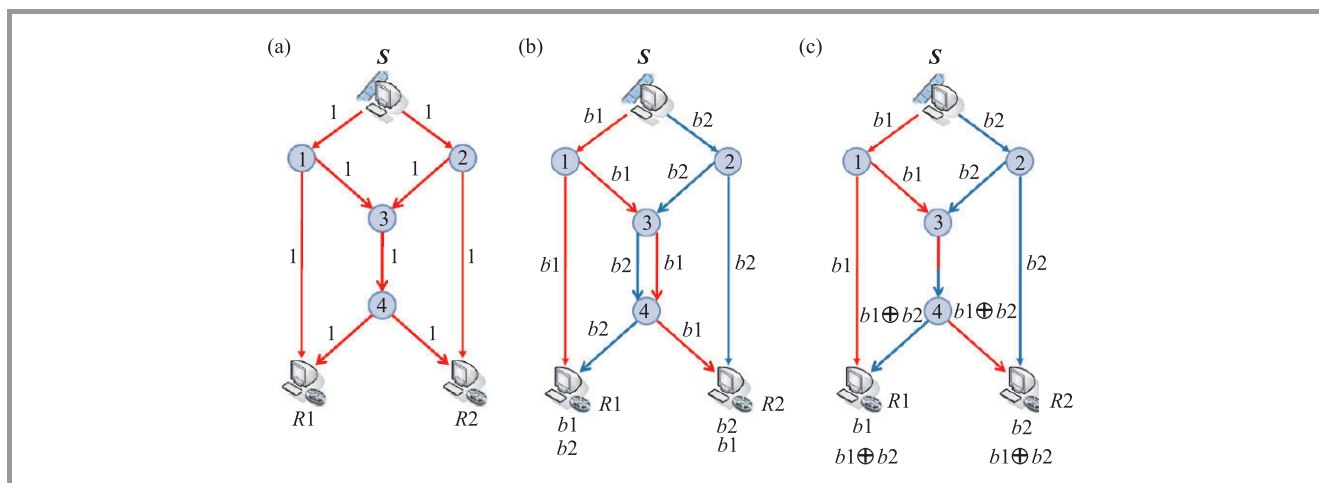
Fig. 1. Communications network: (a) capacity of the edges, (b) traditional approach and (c) approach with network coding.

added. First, an exposition about the features and benefits of network coding for video streaming are introduced in Section 3. Second, specific video frames pictures comparing the different reconstruction quality by using network coding are presented in Section 5. Percentages of packet loss for experiments 1 and 2 have been added too. Finally, a third experiment to evaluate the buffer usage during a video transmission with network coding is added in Section 6.

2. Network Coding Overview

Network coding is a new technique in the field of information theory, proposed by Ahlswede *et al.* in [4]. In this paper, a new problem on networks communications related to information flow is discussed. The authors reveal that it is in general not optimal to regard the information to be multicast as a fluid, rather, by employing coding at the intermediates nodes on the networks, to increase the flow without exceeding the channel capacity. For this study, the network is represented as a directed graph $G = (V, E)$, where the network nodes are represented by V, and the edges E represent the communication channels. The channel capacity is of one data unit per unit time. The source node is a node without any incoming edge.

Network coding is inspired by the Max-Flow Min-Cut theorem which states that [5]: "The maximum amount of flow through the source to the destination equals the minimum capacity required to remove from the network flows that cannot pass from source to destination".

In a single-source multicast session, source node $s \in S$ transmits information at rate R to all receivers $t \in T$, and the maximum multicast information rate in this scenario can be achieved only by allowing coding at intermediate nodes [4]. This optimal multicast rate can be given by finding the capacity through the Max-Flow Min-Cut theorem above described, which relates the maximum information flow through a network to the minimum cut capacity.

To understand better network coding, Fig. 1 presents a scenario for a butterfly network with a source node and two receiver nodes. Figure 1a shows the capacity of each edge. We can observe that the values of the maximum flow of S to any receiver, either $R1$ or $R2$ are equal to two. Therefore, in Fig. 1b, the source S can send two bits $b1$ and $b2$ to $R1$ and $R2$ simultaneously. In this scheme, each intermediate node only replicates and sends out the bit(s) received from upstream. On the other hand, Fig. 1c shows the same network configuration, only that now network coding is implemented. Here, operator $\oplus$ denotes the sum modulo 2. Thus, the receiver $R1$ can recover the two bits $b1$ and $b2$.

Only that $b2$ must be retrieved from $b1 \oplus b2$. Similarly, $R2$ can recover the two bits. In this example, network coding is applied in the node 3. Another important point to note is that the rate multicast increases, because for traditional transmission is 1 bit/time unit, while applying network coding the rate increases to 2 bits/time unit.

The most common benefits by using network coding in a communication network are [4]:

– bandwidth saving,

– improved system throughput,

– reduced delays.

Video streaming systems are distinct from general data delivery systems, in which a client has to download the entire file before using it. In a video streaming session, the receiver can already consume the video while downloading. However, packets with large delay can be perceived by the video application as packet losses and the transmission of these useless packets waste network resources. Therefore, this kind of multimedia application requires high data rate, low-latency and low packet lost rate, which represent a significant challenge for the design of future network architectures. Thus, we believe that network coding can help to improve the performance of the video streaming systems.

3. Related Work

Several research works showing the benefits of network coding have been published in the literature. Mainly, network coding has impacted in the following areas: Mesh, VANET's, MANET's, ad-hoc, sensor networks, security issues, video streaming and content distribution. In the following, we briefly describe some of these applications.

Gkantsidis *et al.* [8] introduced a novel scheme for large scale content distribution using network coding. Based on these findings, Gkantsidis *et al.* report that the system increases the throughput between 20% and 30% compared with the coding applied only on the server. Also each block is unique and is more likely to be useful for other nodes, which facilitates the exchange of these. The authors argue that the distribution rate of 2–3 times improvement in comparison with systems without applying network coding. An application of network coding for wireless networks is introduced by Sundararajan *et al.* [9]. The authors argue that network coding has a positive impact on the TCP/IP (Transmission Control Protocol/Internet Protocol). To show this, a new layer is introduced between the transport layer (TCP) and network layer (IP). Here, the congestion control is not modified. The results show that the throughput has a slight increase when network coding is applied to TCP. Another important point is that traditional TCP fails when packet loss is increased, while the TCP with network coding shows robustness during packet loss situation.

SenseCode [10] is a work in sensor networks, which involves network coding techniques. In this work, the authors are interested in the following parameters: reliability and transmission energy. Reliability refers to the useful information that reaches the receiver, while the transmission energy refers to the amount of energy expended during the transmission of packets from each node. To realize the measurements of these parameters, SenseCode is evaluated and compared with the CTP protocol (Collection Tree Protocol) [11], on the platform TOSSIM [12]. The obtained results show that network coding provides a measure of balance between energy efficiency and reliability. Finally, a scheme where network coding is used to design a secure architecture for video transmission over wireless networks is proposed by Lima *et al.* [13]. The authors evaluate their proposed model using NS-2 [14], for a wireless scenario with losses. The results indicated that the encrypted packets using network coding techniques have a lower size that packets encrypted by the traditional method. Additionally, the probability of packet loss is lower, which allows a better video quality.

Recently, network coding has found a fertile application area in the video streaming systems. For example, Nguyen *et al.* [15] studied the application of network coding in wireless networks for video broadcasting. They propose an optimal scheme to generate erasure codes by using NC to retransmit lost packets. One of the most attractive features of network coding is its ability to achieve the op-

timal multicast rate [4]. In this paper, we extend these features to a multi-source scenario for the transmission of video.

4. Our Model

The main benefits of using network coding can be found in multicast communications where there are link capacities in network, the issue is only to compute the maximum multicast throughput possible for communication between a source node and a set of sink nodes [16]. In this section we propose a multi-source scenario based on a many-to-many distribution scheme in order to study the benefits offered by network coding during a video transmission. Our reference scheme is shown in Fig. 2. In this distri-
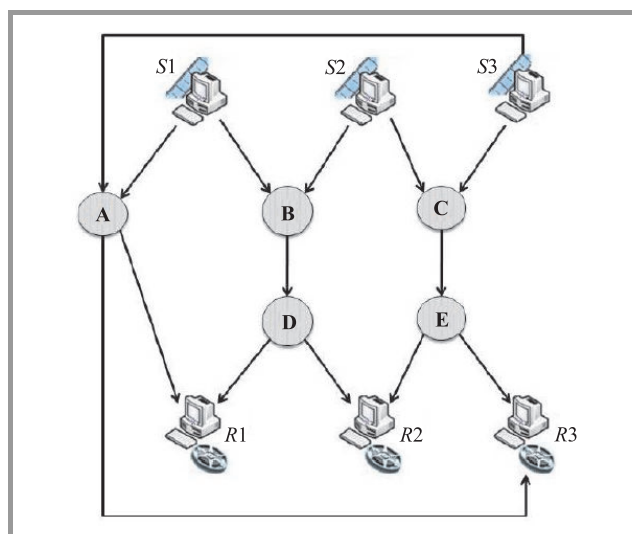
Fig. 2. Our many-to-many distribution scheme.

bution scheme there are three source nodes, three receiver nodes and six intermediate nodes. We use this distribution scenario to deploy our multicast schemes. Bottleneck problems are most likely to be present in this scenario. Thus, we can evaluate the systems robustness by applying network coding techniques. Two multicast schemes are deployed over the many-to-many distribution scheme:

- Without network coding. In this first strategy, the intermediates nodes act as in the traditional networks, that is to say, they take to store in its buffer packets from entering edges, for later forwarding these packets to the out edges. This buffer management policy in each node is called FIFO policy (First-In-First-Out).

- With network coding. In this scheme, network coding techniques are applied in the intermediate nodes of the network. Our proposed scheme is inspired by a work proposed in [6]. In our scheme (see Fig. 2), the intermediate nodes A, B and C perform a linear combination of packets, which are stored in its buffers. On the other hand, nodes D, and E only are responsible for storing and forwarding the received

packets. Once the receiver nodes receive the encoded packets encoded. They are responsible for decoding and reconstructing the original video.

4.1. Network Coding Scheme

Network coding has generated several proposals of encoding schemes. In the following, we describe the network coding technique used for a many-to-many distribution scenario. Our network coding scheme follows the concept introduced in [6], where a centralized knowledge of the networks is not required. This novel approach introduces a new package format, which consists of a global coding vector and the payload. The payload is divided into fields of size 2^8 or 2^{16}, that is to say, each symbol is 8 or 16 bits. This scheme assumes an acyclic graph (V, E), where each edge is of unit capacity and also has a source $s \in V$ and set of receivers $T \subseteq E$. Each edge $e \in E$ emanating from a node v carries a symbol $y(e)$ that is a linear combination of the symbols $y(e')$ on the edges e' that are incoming edges to node v.
Thus,

$$y(e) = \sum_{out(e')=v} m_e(e')y(e') \qquad (1)$$

and

$$m(e) = [m_e(e')]_{e':out(e')=v} \qquad (2)$$

represents the encoding function of the node v on the edges e. This obtained vector $m(e)$ is known as vector-linear coding.

Now, if v is the source node S, then the notation is maintained and the artificial edges $e'_1, \ldots, e'_h$ are introduced. These edges are considered as the input edges to node S, which are the symbols of the source $y(e'_i) = x$, where $i = 1, \ldots, h$. Thus, $y(e)$ on any edges $e \in E$ is a linear combination defined as

$$y(e) = \sum_{i=1,h} g(e)x_i \qquad (3)$$

of the symbols of the source, where the coefficients $g(e) = [g_1(e), \ldots, gh(e)]$ can be recursively determinate by

$$g(e) = \sum_{e':out(e')=v} m_e(e')g(e'). \qquad (4)$$

Thus, the vector $g(e)$ is called the global encoding vector along the edges e. Each node receives packets that are linear combinations of the source packages and stores them in an array.
In the scheme proposed by Chou $et\ al.$ [6], each node sends packets obtained as a random linear combination of packets stored in its buffer. So within each package is a global coding vector $g(e)$ of dimension h. Any receiver can then recover the source vectors $x_1, \ldots, x_h$ using Gaussian elimination on the vectors in its h received packets. Packets arrive at the receiving nodes on the incoming edges, and they may find related packages, that is to say, with the same source vector $x_1, \ldots, x_h$. This type of packages is known

as packets of the same generation, where h represents the size of the generation. All packages of the same generation are labeled with the same generation number. The policy used by Chou $et\ al.$ is "empty the current generation before it reaches the next generation". This policy indicates that we must attend the packets of a same generation that existed before to attend to other packets of a different generation.

5. Evaluation

Our simulation is implemented on the network simulator NS-2, which is a discrete event simulator targeted at networking research [14]. Several simulations of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks can be supported by this simulator. NS-2 is divided into two hierarchies, the compilation written in C++ and the interpreter in OTcl. Both hierarchies are closely linked. Our simulation scenario is shown in Fig. 3. Video sequences used in our simulations are represented in each source node. All receiver nodes expect to receiver all video sequences.

Fig. 3. Network coding simulation scenario with three sources and three receivers

We use the following simulation parameters. The edges delay is 10 ms, the buffers size in each node is 819 kbit, and UDP is the transport protocol. A dense mode version is used in the routing protocol. This protocol works as follows. Initially, the protocol assumes that all nodes in the network are receivers of multicast traffic and therefore packets are transmitted across the network. Each node that receives such traffic, but not requires it, sends a prune message via the interface where it is received. This process is known as tree pruning and runs every few minutes.
To simulate the video traffic in our experiments, we have used the video traffic generated from the EvalVid project (a video quality evaluation tool). EvalVid is a research project carried out by the TKN group at the Faculty of Electrical Engineering and Computer Science at the Technische Universität Berlin [17]. The video traffic generation in the NS-2 simulator follows the next steps:

1. We use uncompressed video sequences in CIF format (352 × 288), which are available from [18]. This video sequences are sampled in a videoconferencing format (format: 4:2:0).

2. After this, we compress the video files according to the H.264/MPEG-4 AVC standard. To this end, the videos sequences were encoded to 30 frames per second with a variable bit rate using the JM 17.1 software [19].

3. Then, the video frames are packaged using the Real-Time Protocol (RTP). The MP4Box software is used to carry through this task [20]. The size of each packet is 1024 B.

4. Next, a trace file is generated in order to be used in the NS-2 network simulator. This file is created using the mp4trace software, which is part of the EvalVid project [17].

5. Finally, a traffic source is generated in the NS-2 network simulator. This goal is reached by establishing an UDP shipping agent and a receiving agent in the simulator [17].

5.1. Throughput Evaluation

Our average throughput evaluation considers two different experiments related to the size of the video files. In the first experiment, we use the Foreman (300 frames), Akiyo (300 frames) and Football (90 frames) video sequences, which are allocated in the source 1, source 2 and source 3, respectively. We can see that a video sequence (Football) has a different size. In our second experiment, we use the following video sequences, Container (300 frames), Akiyo (300 frames) and Foreman (300 frames), which are allocated in the source 1, source 2 and source 3, respectively. Unlike the first experiment, now all video sequences have the same size. Figure 4 shows the system average through-
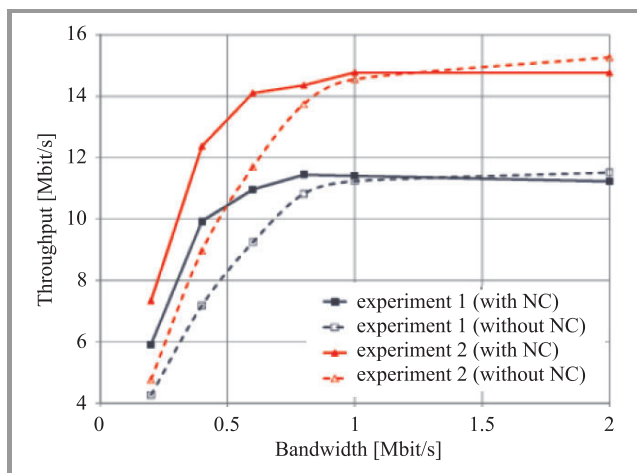
Fig. 4. Average system throughput for both experiments.

put for both experiments. We can see that in experiment 1, the average throughput is higher if network coding is applied while the bandwidth is below 1 Mbit/s. After this

threshold, the bandwidth is increased above 1 Mbit/s and the average throughput is stabilized. With respect to the experiment 2, we can see that the results are similar, but the average throughput is increased for both strategies, either using or not using network coding. However, the average throughput is still higher when the network coding strategy is applied.

5.2. Video Quality Evaluation

Our second parameter to be evaluated is the video quality. We use the peak-signal-to-noise-ratio (PSNR) as the quality metric. For this experiment, the Foreman, Akiyo and Football video sequences are encoded at different bit rate using the JM software [19]. The bit rates used for encoding these video sequences are 16, 64, 200, 500 and 1000 kbit/s. We measure the PSNR in the three receiving nodes. However, in this paper, the PSNRs obtained in the receiving node 1 are shown only in Fig. 5. We can see how the Akiyo and Foreman video sequences reach a higher PSNR when network coding is applied than traditional video transmission without network coding.
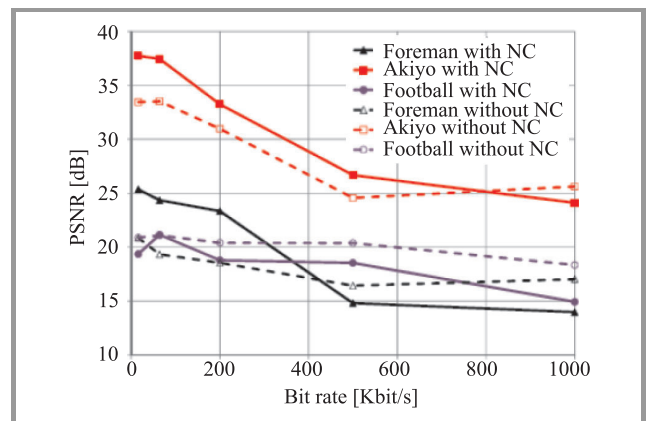

Fig. 5. Average PSNR for experiment 1.

On the contrary, the results obtained for the Football sequence are opposite. This is due the packet loss in the network. The Football video sequence is of smaller size than Akiyo and Foreman video sequences; therefore its PSNR is most affected by the packet loss. Thus, the video quality

Table 1
Percentages of packet loss for experiment 1

Sequence	Foreman	Akiyo	Football
Receiver R1			
With NC	0.02	0	37.4
Without NC	17.78	34.2	41.85
Receiver R2			
With NC	9.82	4	32.63
Without NC	17.76	34.14	51.3
Receiver R3			
With NC	0	0.2	46.57
Without NC	18.89	15	41.84

for the Football sequence in the receiving node could be not acceptable. From these findings, we can argue that video quality can be improved by using network coding during a packet loss event in the network.

The percentages of packet loss in each receiver node are summarized in Table 1. Receiver nodes $R1$ and $R2$ have a lower packet loss using network coding than if it is not used. However, the football video sequence has a high percentage of packet loss in all receiver nodes. This is because the football video sequence has a small size (90 frames) in comparison with the Akiyo and Foreman video sequences, which have 300 frames. Figure 6 compares a video frame
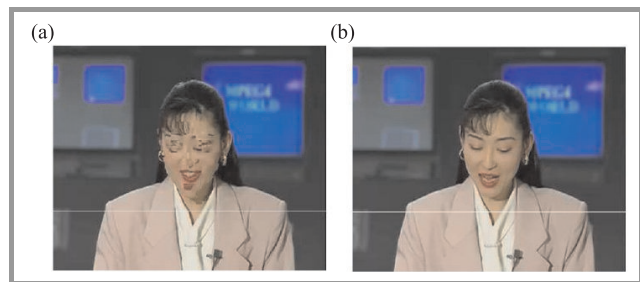
Fig. 6. A frame of the Akiyo video sequence during a transmission with packet loss: (a) without network coding; (b) using network coding.

of the Akiyo video sequence during a transmission with packet loss. Video frame in Fig. 6a is transmitted via a traditional network, while video frame in Fig. 6b is transmitted using network coding. We can observe that video quality is improved in the picture where network coding is used.

Figure 7 depicts the PSNR obtained from experiment 2. In this experiment, all video sequences have the same size. The resulting PSNRs are similar to PSNRs obtained from experiment 1. We can see how once again the PSNR is higher when network coding is applied. Contrary, if network coding is not used, the number of lost packet is increased and the video quality (PSNR) is proportionally reduced. Table 2 summarizes the percentages of packet loss in each receiver node for the experiment 2. All video sequences present a high percentage of packet loss in all
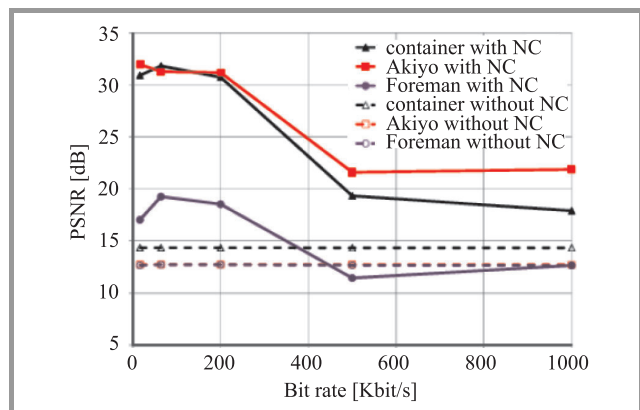

Fig. 7. Average PSNR for experiment 2.

Table 2
Percentages of packet loss for experiment 2

Sequence	Foreman	Akiyo	Football
Receiver $R1$			
With NC	0.0	0	5.10
Without NC	38.0	36.67	35.43
Receiver $R2$			
With NC	0.0	0.0	14.51
Without NC	38	36.67	46.02
Receiver $R3$			
With NC	14.27	0.0	0.0
Without NC	47.47	35.10	35.43

receiver nodes when network coding is not used. In contrast, we can observe that some video sequences are received free of packet loss by the receiver nodes when the network coding technique is used during the video transmission.

5.3. Evaluation of the Buffer Occupancy

Finally, a third experiment is implemented in order to evaluate the buffer occupancy during a video transmission with network coding. We realize this evaluation in the buffers of the intermediate nodes because network coding is realized in these nodes. Specifically, we measure the buffer occupancy at the node 3 from our simulation scenario (see Fig. 3). Figure 8 shows a comparison of buffer usage using
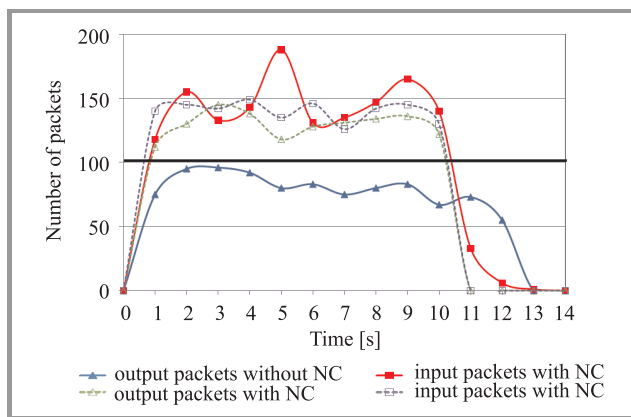

Fig. 8. Buffer occupancy during a transmission rate of 64 kbit/s.

network coding (NC) and a traditional transmission (without network coding) during a video transmission with a bit rate of 64 kbit/s. Threshold indicates the buffer size, which can store 100 packets of 1KB. We can see that during a traditional transmission, the number of input packets arriving to the buffer is higher than the number of output packet from it. This fact can produce packet loss. Contrary, in a video transmission with network coding, packets arriving to the buffer are combined and sent immediately to the next node. Thus, two or more packets can be attended at same time. Our experiments indicate that the combining time at the buffers are small and the packet loss is re-
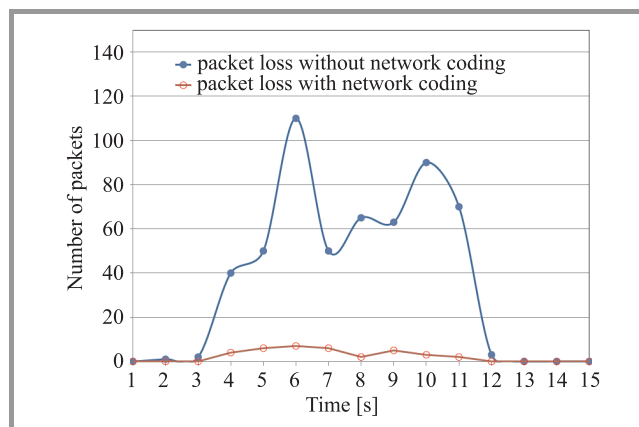
Fig. 9. Packet loss during a video transmission.

duced. Figure 9 compares the packet loss for these cases. Results shows that network coding can notably reduce the packet loss during a video transmission in communication networks with limited bandwidth. However, this comparison of results can be different when the communication networks have abundant bandwidth.

6. Conclusions

In this paper, we presented and evaluated a multi-source scheme using network coding for video delivery to a set of receiving nodes. Our network coding scheme does not require a centralized knowledge of the network and introduces a new package format, which contains a global coding vector and the payload. The video packets are streaming via UDP connections. We evaluated our multi-source scheme using the network simulator NS-2. Different video sequences were compressed under the H.264/MPEG-4 AVC standard and evaluated in this simulator. Mainly, we interest to evaluate the network performance and video quality during a video transmission using network coding. Additionally, buffer performance using network coding is evaluated. The results show that our scheme with network coding achieves a promising performance in terms of throughput, delay, packet loss, buffer usage and video quality compared with traditional networks.

We plan to extend our effort toward the peer-to-peer (P2P) networks. P2P networks are a promising infrastructure for video delivery. We believe that network coding can introduce several benefits in multi-source video multicast systems based on peer-to-peer networks. Also, scalable video techniques combined with network coding can be investigated.

References

[1] T. Nguyen and A. Zakhor, "Multiple sender distributed video streaming", *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 315–326, 2004.

[2] F. A. López-Fuentes and E. Steinbach, "Multi-source video multicast in peer-to-peer networks", in *Proc. IEEE Int. Symp. Parallel Distrib. Process. IPDPS'08*, Miami, FL, USA, 2008, pp. 1–8.

[3] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory", *Foundation Trends in Communications And Information Theory*", vol. 2, no. 4 and 5, pp. 241–381, 2005.

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

[5] B. Bollobas, *Graph Theory, and Introductory Course*. New York: Springer, 1979.

[6] P. Chou, Y. Wu, and K. Jain, "Practical network coding", in *Proc. 51st Allerton Conf. Communi., Control and Comput.*, Monticello, IL, USA, 2003.

[7] C. Cabrera-Medina, F. A. López-Fuentes, "Video transmission in multicast networks using network coding", in *Proc. Int. Conf. P2P, Parallel, Grid Cloud and Internet Comput. 3PGCIC-2011*, Barcelona, Spain, 2011, pp. 350–354.

[8] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution", in *Proc. IEEE INFOCOM 2005*, Miami, FL, USA, pp. 2235–2245, March 2005,.

[9] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher and J. Barros, "Network coding meets TCP", in *Proc. IEEE INFOCOMM 2009*, Rio de Janeiro, Brazil, 2009, pp. 280–288.

[10] L. Keller, E. Atsan, K. Argyraki, and C. Fragouli, "SenseCode: Network Coding for Reliable Sensor Networks", Tech. Rep. 2009, Ecole Polytechnique Federale Lausanne (EPFL). Last updated July, 2010.

[11] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss and P. Levis, "Collection tree protocol", in *Proc. 7th ACM Conf. Embedded Netw. Sensor Sys. SenSys*, Berkeley, CA, USA, 2009.

[12] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications", in *Proc. First ACM Conf. Embedded Netw. Sensor Sys. SenSys*, Los Angeles, CA, USA, 2003.

[13] L. Lima, S. Gheorghiu, J. Barros, M. Medard, and A. Toledo-López, "Secure network coding for multi-resolution wireless video streaming", *IEEE J. Selec. Areas Commun.*, vol. 28, no. 3, pp. 377–388, 2010.

[14] S. McCanne, S. Floyd, and K. Fall, "The network simulator, ns-2" [Online]. Available: http://www.isi.edu/nsnam/ns

[15] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *Proc. Packet Video*, Lausanne, Switzerland., 2007, pp. 326–33.

[16] T. Noguchi, T. Matsuda, and M. Yamamoto, "Performance evaluation of new multicast architecture with network coding", *IEICE Trans. Commun.*, vol. E86-B, no. 6, pp. 1788–1795, 2003.

[17] EvalVid, "A video quality evaluation tool" [Online]. Available: http://www.tkn.tuberlin.de/research/evalvid/

[18] "Video sequences" [Online]. Available: http://www.tkn.tu-berlin.de/research/evalvid/cif.html

[19] "JM v.17.1 software", Fraunhofer Heinrich-Hertz-Institut [Online]. Available: http://ip.hhi.de/index.htm

[20] "MP4Box software" [Online]. Available: http://www.tkn.tu-berlin.de/research/evalvid/

Francisco de Asís López-Fuentes received the B.Sc. degree in Electrical Engineering from the Oaxaca Institute of Technology, México. He received the M.Sc. degree in Computer Science Minoring in Networking from the Monterrey Institute of Technology (ITESM), and the doctoral degree in Electrical Engineering from Technical University of Munich, Germany. From 1988 to 1994, he was a design engineer in Siemens, Mexico. From 1996 to 2003, he was an Associate Professor at Technological University of Mixteca, México.

Dr. López-Fuentes is currently the Associate Professor of the Department of Information Technology at Autonomous Metropolitan University – Cuajimalpa Campus, México City, México. He has also served as a reviewer for several journals and conferences. His current research interests are in the area of networked multimedia systems, peer-to-peer networks, distributed systems as well as network security.
E-mail: flopez@correo.cua.uam.mx
Department of Information Technology
Autonomous Metropolitan University – Cuajimalpa
Constituyentes Av. 1047
11950 México, D. F., México

Cesar Medina-Cabrera is an information technology professional currently employed in the electricity industry. He spends most of his work time engaged in communication networks and video streaming applications. He holds the bachelor degree in Electronic Engineering from Autonomous Metropolitan University in México. At present he is finishing his master's degree in sciences and information technology at the same university. His research interests include the computer networks and communication systems.