Implementing Video Data Compression along with Network Coding in Video Multicast

Ankush Kapoor, Harsh Magotra, Garima Bharti, Shweta Pandit
Dept. of ECE, JUIT, Waknaghat
Solan, H.P. 173215, India
{ankush8819, harshmagotra09, geru.025, ershweta3}@gmail.com

Nitin Rakesh
Dept. of Computer Science and Engineering
Jaypee University of Information and Technology, Waknaghat,
Solan, H.P. 173215, India
nitin.rakesh@gmail.com

***Abstract*— This paper presents a technical review to real time Video Multicast. In this paper Video Multicast technique is being described and to improve the quality and efficiency using network coding and video data compression. Video data compression is concerned with reducing the amount of data required to reproduce a digital video. It is a key component in facilitating the widespread use of digital video, which is currently prevented by the mismatch between the huge storage and transmission bandwidth requirements of video and the limited capacity of existing computer systems and communications networks. In this paper we provide measure to remove the problem of video multicast in lossy networks using data compression, network coding and multiple description codes. The rate allocation for multiple descriptions can be optimized at the source to generate a scalable video bit stream such that the expected PSNR of the video at the receivers is maximized. Here we implement Huffman Coding as it is being lossless data compression. Although the range of video compression standards now available, such as H.261 or MPEG-2, exhibit acceptable performance when operating at high bit rates (eg >64 Kbps) and we can implement these.**

***Keywords*—coding, data compression, MD-FEC.**

I. Introduction

In the area of multimedia, multicast is the delivery of a message or useful data to a group of destination computers simultaneously a single transmission from the source creating copies automatically in other network elements, such as routers, only when the topology of the network requires it. The main challenges towards reliable video multicast over the Internet are limited available bandwidth for transmission and packet losses that occur in the links. Traditionally, data over the Internet have been sent via packets that are forwarded by nodes/routers inside the network (pure routing) or by multicast, where nodes replicate and forward the data [1]. Application-layer multicast networks were introduced to overcome difficulties in implementing network-layer multicast. Packets were sent in multicast trees and the end users were leaf nodes of these trees.

Compression is useful because it helps reduce the consumption of expensive resources, such as data storage device or bandwidth of our link. On the downside, compressed data must be decompressed to be used, and this extra processing may decrease the performance of some applications. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed (the option of decompressing the video in full before watching it may be inconvenient, and requires storage space for the decompressed video).Huffman coding is an algorithm for constructing efficient source codes for a Discrete Memoryless Source (DMS) with source symbols that are not equally probable [2]. The variable length encoding algorithm was suggested by Huffman in 1952 based on the source symbol probabilities P(x),i=1,2,…..L. Network coding is a technique where instead of simply relaying the packets of information they receive the nodes of a network will take several packets and combine them together for transmission. Purpose of using network coding is to attain maximum possible information flow in a network. The traditional approach (store-and- forward) to multicasting is to find an efficient multicast tree in the network, with the nodes acting as switches which can route and/or replicate information. Network coding is a new theory which changes the way we think of transmitting information in a network. Steps which we will follow in our approach are as in fig 1.

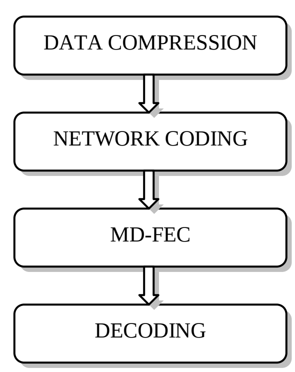


Figure 1. Algorithmic Flow in proposed model.

Concurrently, scalable video coding was developed and the various layers of video were transmitted in different multicast trees. Forward error correction (FEC) based techniques have been widely used to protect data from losses. In order to protect a prioritized video bit stream, Unequal Error Protection (UEP) methods were developed and are widely used [4]. To ensure that the quality of received video is optimal; several algorithms were developed to optimize the rate allocation problem [2-3]. There has been a lot of research going on to use network coding in video multicast. There has been active work both in peer-to-peer networks and in application-layer overlay multicast networks.

We consider overlay multicast networks with a directed acyclic graph topology. It has been shown that there exist networks where the multicast capacity of network coding can be arbitrarily large compared to the routing schemes. However we look at the implementation of multicast to users in a lossy network and use network coding. In this work, we seek to optimize the rate allocation for highly embedded scalable video coders using practical network coding on directed acyclic graphs to give a robust video multicast scheme [6]. Our method does not need multiple multicast trees that are necessary in routing with replication.

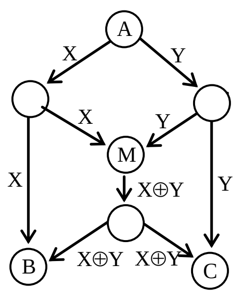


Figure 2. Butterfly network showing how multicast capacity is achieved using network coding.

II. DATA COMPRESSION

Compression is useful because it helps reduce the consumption of expensive resources as requirements may outstrip the anticipated increase of storage space and bandwidth for data storage and data transmission. On the downside, compressed data must be decompressed to be used, and this extra processing may be detrimental to some applications. Video data compression is concerned with reducing the amount of data required to reproduce a digital video. It is a key component in facilitating the widespread use of digital video, which is currently, prevented by the mismatch between the huge storage and transmission bandwidth requirements of video and the limited capacity of existing computer systems and communications networks[9-10].

For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed (the option of decompressing the video in full before watching it may be inconvenient, and requires storage space for the decompressed video) [4]. The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced and the computational resources required to compress and uncompress the data[5][6].

Huffman code is a source code whose average word length approaches the fundamental limit set by the entropy of a discrete memoryless source. The Huffman code is optimum in the sense that no other uniquely decodable set of code-words has a smaller average code-word length for a given discrete memoryless source. The essence of the algorithm used to synthesize the code is to replace the prescribed set of source statistics of a discrete memoryless source with a simpler one. This reduction process is continued in step-by-step manner until we are left with a final set of source statistics (symbols) of only two, for which (0, 1) is an optimal code. Huffman was able to design the most efficient compression method of this type: no other mapping of individual source symbols to unique strings of bits will produce a smaller average output size when the actual symbol frequencies agree with those used to create the code. Let us implement data compression technique here so let us consider "Huffman" as data compression technique and take our input data to be stream of bits such as:

A B E A C A D A B E A….. where

P (A) =0.45

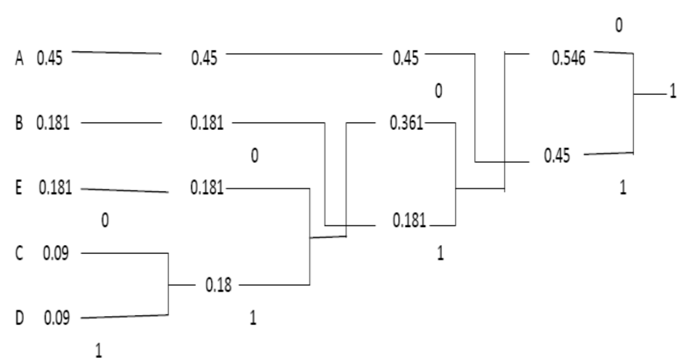
P (B) =0.181

P (E) =0.181

P (C) =0.09

P (D) =0.09

Huffman Coding works in the below given tree order:



So codeword are as:

A=1

B=01

E=000

C=0010

D=0011

Because input stream: A B E A C A D A B E A ....

Compressed stream: 1 01 000 1 0010 1 0011 1 01 000 1 ....

We are applying the approach that the remainder (no. of bits after compression /4) = 0, if not then we insert 0's at the end of compressed stream, to make each pair of 4 bits. Therefore compressed stream (after adding 0's):1010 0010 0101 0011 1010 0010.

And as from fig. 2 if we imply network coding concepts here we get data through XORing as shown below.

Network Coding by bit-by-bit XORing:

(1010) ⊕ (0010) = 1000
(0101) ⊕ (0011) = 0110
(1010) ⊕ (0010) = 1000….similarly other data streams were XOR and sent using network coding.

III. NETWORK CODING

Network coding is a technique where, instead of individually transferring each packet, we combine several packets and transfer them through our network [11]. This is used to increase throughput and to reduce delay in our transmission. Network coding is a subpart in Information and Coding theory. Routing of packets has been the primary method by which data have been transmitted across packet networks, such as the Internet. However, routing alone is not sufficient to achieve the multicast capacity of the network. Basically peer-to-peer networks use network coding to reduce the amount of routing information required by peers to achieve near optimal throughput [8]. In large networks this can be a significant advantage, since the routing overhead will increase in proportion to size of the network.

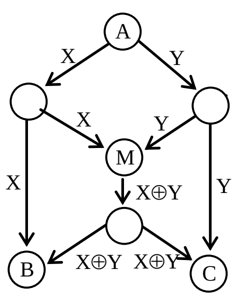


Figure 3. Butterfly network showing data travel in a network using network coding.

Some of the advantages of network coding are illustrated below. While it has been shown that there exist networks where the improvement in capacity using network coding is arbitrarily large, not all networks exhibit a gain in throughput using network coding - such as lossless tree networks. In lossless networks, capacity improvement occurs only when the flow to two or more receivers is limited by the capacity of a common link. Network coding combines the flows to the receivers in the same link such that the individual flow to each receiver on that link is not affected.

IV. MD-FEC IN VIDEO MULTICAST

Puri and Ramachandran [3] have given and implemented an idea about MDFEC which we are using in our concept as explained further. In the current deployment of the Internet, network routers are typically oblivious to the structure or content of the packets and treat all packets equally. While this simple network design has enabled the Internet to scale to its present size, it adversely impacts multimedia rows. Furthermore, emerging multimedia compression standards for image/video coding are evolving towards a multi-resolution (MR) or layered representation of the coded bit streams. Multiple descriptions (MD) source coding has recently emerged as an attractive framework for robust transmission over unreliable channels.
The problem formulation can be expressed as [4]:

$$\min_{\{R_i\}_{i=1}^{N}} \sum_{k=0}^{N} q_k D(R_k),$$

$$R_1 \le R_2 \le \cdots \le R_N, \text{ and}$$

$$\sum_{k=1}^{N} \alpha_k R_k \le B,$$

$$\text{where}, \alpha_k = \frac{1}{k(k+1)}, \text{for } k = 1, 2, \ldots, N-1, \text{and } \alpha_N = 1.$$

The essential idea is to generate multiple independent descriptions (N in number) of the source such that each description independently describes the source with a certain desired fidelity; when more than one description is available, they can be synergistically combined to enhance the quality. The proposed strategy uses forward error correction (FEC) channel codes and incorporates the priority encoding transmission (PET) philosophy in formulating a systematic, fast and end-to-end rate distortion optimized algorithm [7]. In this section, we briefly describe the mechanics of the packetization strategy that converts the prioritized multi-resolution bit stream into an *N*-packet un-prioritized MD packet stream using efficient erasure channel codes. Each description in the MD stream occupies an entire network packet, thus the terms description and packet are used interchangeably.
The rate allocation as obtained by MDFEC is nearly optimal. The distortion metric used in optimization is often replaced by PSNR. If PSNR is used, the objective would be to maximize the expected PSNR at the destination [11].

V. FINAL DATA AT RECIEVER NODES

Basically data received at our final nodes is the network coded compressed data and in fig. 3 at nodes B and C we get data in the form as shown below:

At node B → x and x⊕ [x⊕y] = y
At node C → y and y ⊕ [x⊕y] = x

So at both nodes we receive x and y which we have initially transmitted. Also in fig 3, as x and y denote to individual streams which contains four bits of data in compressed form which can be decoded in the following manner.

PROBLEM: If before network coding at transmitter, remainder=1, that means we have to add three zero's at the end. Since we have code 000 for E and 1 for A, there may be confusion at receiver, that whether decoder decodes 1000 as A or as A (=1) and E (=000).

SOLUTION: In order to decode the input data properly at the receiver end, we use two bits to represent the number of zeros padded in the input stream (see fig. 4 for details).

Preamble (2 bits)	Data (compressed)

PREAMBLE	NUMBER OF ZERO'S ADDEDAT THE END
00	0
01	1
10	2
11	3

Figure 4. Final compressed data stream through the transmission channel.

VI. CONCLUSIONS

In this paper we will transmit our video in multicast fashion. To avoid losses and to increase our efficiency of the system, we take data compression technique and network coding into account.

We have improved our system performance in the following aspects:

i. To reduce the weight of the link
ii. To increase throughput
iii. To increase capacity
iv. To avoid losses during transmission
v. To avoid delay
vi. To reduce receiver complexity
vii. Security of data is enhanced

So, here we have implemented data compression technique in combination with network coding in order to have high throughput and more efficient transmission system.

References

[1] Rohit Puri, K. Ramachandran, K.W. Lee, V. Bharghavan in Signal Processing: *Image Communication* 16 (2001) 745}762.

[2] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission" in IEEE Symposium on Foundations of Computer Science, pp. 604-612, 1994.

[3] A. Mohr, E. Riskin, and R. Ladner, "Unequal loss protection: graceful degradation of image quality over-packet erasure channels through forward error correction," IEEE Journal on Selected Areas in Communications 18, pp. 819-828, June 2000.

[4] R. Puri and K. Ramachandran, "Multiple description source coding through forward error correction," in Proc. of the 33rd Asilomar Conference on Signals, Systems and Computers, Pacific Groove, CA, pp. 342- 346, October 1999.

[5] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," IEEE Transactions on Information Theory 46, pp. 1204-1215, July 2000.

[6] X. Chenguang, X. Yinlong, Z. Cheng, W. Ruizhe, and W. Qingshan, "On network coding based multirate video streaming in directed networks," in IEEE International Performance, Computing and Communications Conference (IPCCC '07), pp. 332-339, 2007.

[7] M. Shao, X. Wu, and N. Sarshar, "Rainbow network flow with network coding," in Fourth Workshop on Network Coding, Theory, and Applications (NetCod), pp. 1-6, 2008.

[8] H. Wang, S. Xiao, and C. Kuo, "Robust and flexible wireless video multicast with network coding," in IEEE Global Telecommunications Conference (GLOBECOM '07), pp. 2129-2133, 2007.

[9] N. Sarshar and X. Wu, "Rate-distortion optimized multimedia communication in networks," in Visual Communications and Image Processing, January 2008.

[10] J. Cannons, T. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity," IEEE Transactions on Information Theory 52, pp. 777-788, March 2006.

[11] K. Bhattad, N. Ratnakar, R. Koetter, and K. R. Narayanan, "Minimal network coding for multicast," in International Symposium on Information Theory, pp. 1730-1734, 2005.