Video multicast using Network Coding

Adarsh K. Ramasubramonian[a], and John W. Woods[a]

[a]Dept. of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY USA

ABSTRACT

We investigate the problem of video multicast in lossy networks using network coding and multiple description codes. The rate allocation for multiple descriptions can be optimized at the source to generate a scalable video bitstream such that the expected PSNR of the video at the receivers is maximized. We show that using network coding with multiple description codes can significantly improve the quality of video obtained at the receiver, in comparison to routing (with or without replication). Simulations show that as the loss rate increases, the improvement in the performance increases and in certain cases for loss rate of 0.20, the improvement can be as high as 3 to 3.5 dB when compared to routing with replication. Moreover, network coding obviates the need to construct multiple multicast trees for transmission, which is necessary in routing with replication.

Keywords: Network coding, video multicast, multiple description codes

1. INTRODUCTION

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. 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. 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 bitstream, Unequal Error Protection (UEP) methods were developed and are widely used.[1] To ensure that the quality of received video is optimal, several algorithms were developed to optimize the rate allocation problem.[2,3]

Ahlswede et al.[4] showed that using network coding, that requires intermediate nodes to encode incoming data, a multicast capacity equal to the minimum of the min-cut capacities of all the receivers can be achieved. This multicast capacity cannot be achieved using only routing (with or without replication). 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.

Chenguang et al.[5] developed a layered network coding solution to the multicast problem for lossless links, where each source layer is split into a separate stream and network coding is done separately for each stream. Shao et al.[6] studied the effect of using network coding with descriptions in lossless networks; however, they do not provide any experimental results. Wang et al.[7] developed an unequal error protection solution for wireless video streaming on packet erasure channels using network coding. They consider transmitting a layered coded bitstream in the form of descriptions. However, they do not consider the optimization of the rate allocation for the descriptions. Sarshar and Wu[8] considered multiple description coded video to be transmitted using routing with replication to heterogeneous users of a lossless network. Our work is similar to this case, however we look at the implementation of multicast to users in a lossy network and use network coding. In this work, we seek to

For correspondence:
A.K.R.: E-mail: ramasa@rpi.edu
J.W.W: E-mail: woods@ecse.rpi.edu

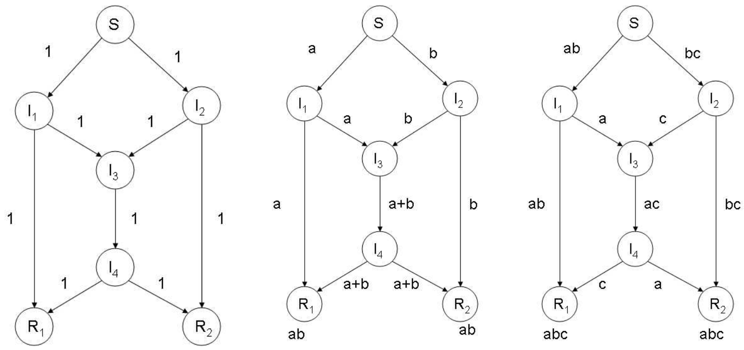


Figure 1. (a) Butterfly network. (b) shows how multicast capacity is achieved using network coding. (c) shows how the routing capacity is achieved.

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. Our method does not need multiple multicast trees that are necessary in routing with replication.

2. NETWORK CODING

Routing of packets has been the primary method by which data have been transmitted across packet networks, such as the Internet. When multiple users are interested in the same data from the same source, multicast routing is typically used. However, routing alone is not sufficient to achieve the multicast capacity of a network. By allowing intermediate nodes in a network to code incoming bytes/packets, Ahlswede et al.[4] showed that the multicast capacity of a network can be achieved. This is best illustrated by the example of the butterfly network shown in Fig. 1(a), where all links have unit capacity. From the min-cut max-flow theorem, the multicast capacity of the network to each receiver R_1 and R_2 is 2. Using routing (with replication), shown in Fig. 1(c), we can only achieve a capacity of 1.5[9,10] for both the users. Using network coding, both users can receive source data at the multicast capacity(Fig. 1(b)).

2.1 Advantages of network coding

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.[11] 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. In Fig. 1(b), I_3 combines the flows to receivers R_1 and R_2 into the same link in a way that is not possible using routing. Even in networks where there is no throughput advantage, network coding can simplify code construction.

In directed acyclic networks, it is sufficient to use linear codes to achieve the multicast capacity.[12] Sanders et al.[13] developed a polynomial time algorithm to construct codes for such networks. Another popular scheme is randomized network coding.[14]

2.2 Practical Network Coding

Network coding is also beneficial in robust transmission on lossy networks. Chou et al[15] presented a randomized network coding scheme Practical Network Coding (PNC), that can be implemented in real networks. In PNC, the data stream is divided into groups, called generations, and each generation is grouped into a set of packets (say N of them). At the source, the packets belonging to the same generation are linearly combined using random coefficients. In order to facilitate decoding at the receiver, these random coefficients are attached to the generated packets. At the intermediate nodes, packets are collected in a buffer and grouped according to the generation number. When sufficient number of packets of a generation arrive, the node linearly combines these packets and sends them to the next node. The encoding vectors that are attached to the packet are also linearly combined. At the receiver, if N linearly independent packets of a generation arrive, the entire generation can be decoded. The overhead involved in including the encoding vectors would be negligible if the packet sizes are large. To provide prioritized protection to the source, a UEP based scheme (such as Priority Encoded Transmission,[1] or PET) can also be used with network coding.[15] This ensures that even if all packets belonging to a generation do not arrive at the receiver, the source data can be recovered partially.

3. NETWORK CODING WITH MD-FEC

3.1 MD-FEC

Several algorithms were developed based on the PET scheme to obtain the rate allocation of various layers of the video bitstream.[2,3] Puri and Ramachandran developed MD-FEC[3] (Multiple description coding with forward error correction) a nearly optimal method to find the bit-rate allocation for transmission over a lossy link, such that the expected distortion of the received video is minimized. The embedded video stream is split into N descriptions and each layer is protected based on its importance. As more packets arrive, more layers can be decoded and the quality of the reconstructed video improves. The bandwidth of the link acts as the constraint to the optimization problem.

Assume that a set of N descriptions of a GOP is to be sent from a source to a receiver, connected by a single link with available link bandwidth B and packet-loss rate p. Let the number of packets arriving at the receiver have a probability distribution given by q_k for $k = 0, 1, 2, \ldots, N$. For a packet erasure channel, this distribution can be modeled as a binomial random variable with parameters (N, p). The distortion associated with the decoded video, $D(R)$, is a function of the source rate of decoded video. In our case, $D(R)$ would correspond to the distortion observed when the GOP is decoded at rate R. MD-FEC attempts to minimize the expected distortion at the receiver. The algorithm optimizes N source rate points $\{R_i\}_{i=1}^{N}$ based on following two constraints - (1) the quality of video should not decrease as more descriptions are received, and (2) the overall channel rate of descriptions should not exceed the available bandwidth. The problem formulation can be expressed as

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

subject to,

$$R_1 \leq R_2 \leq \ldots \leq R_N, \text{ and} \tag{2}$$

$$\sum_{k=1}^{N} \alpha_k R_k \leq B, \tag{3}$$

where, $\alpha_k = \frac{1}{k(k+1)}$, for $k = 1, 2, \ldots, N-1$, and $\alpha_N = 1$. The rate allocation obtained by the MD-FEC algorithm is nearly optimal.[3] The distortion metric used in the optimization is often replaced by PSNR. If PSNR is used, the objective would be to maximize the expected PSNR observed at the receiver. For our simulations, we maximize the expected PSNR to obtain the source rate breakpoints.

3.2 MD-FEC with random codes

Puri and Ramachandran used Reed Solomon (RS) codes to protect the descriptions from packet losses. However, any code that can provide erasure protection can be used. Random codes can also be used, provided we are able to convey the coding vectors to the receiver. The rate-allocation algorithm for random codes would be the same as when using RS codes. We fill the source bitstream in the data byte locations, and zeros in the parity byte locations. Once the data have been arranged in the packets, we perform a linear combination of the packets and attach the encoding vector (vector containing the random coefficients used for the linear combination) to prefix the packets. The packets are then sent to the receiver. Decoding at the receiver is performed using Gaussian

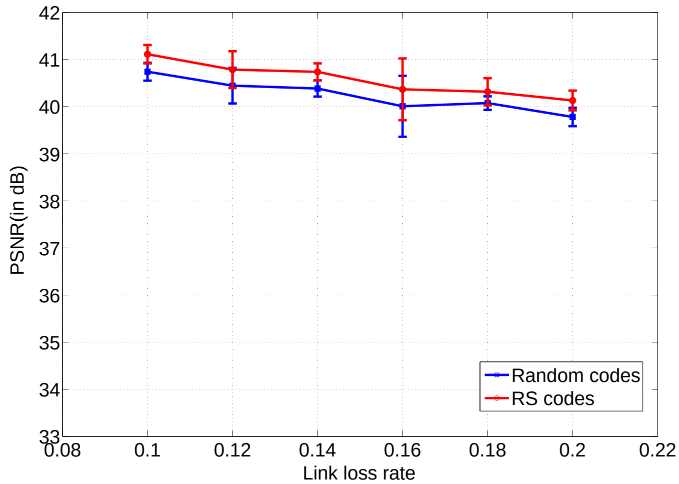


Figure 2. Comparison of average PSNR of the *Foreman* sequence obtained at the receiver using MD-FEC with random codes and RS codes. The source and receiver were connected by a 1000 Kbps link.

elimination, defined on the Galois Field used (in our case GF(2^8)). If all the descriptions do not arrive, only part of the source data can be recovered.

The motivation here is that since we use random coefficients to encode the packets in practical network coding, we could combine MD-FEC with practical network coding by using random codes for protection of packets, rather than RS codes. However, there are two main issues that need to be addressed when using random codes. The first issue is the small probability of non-decodability associated with random codes. As we are randomly choosing the coefficients for encoding the packets, there is a likelihood that we receive linearly dependent packets at the receiver. This probability has been shown to be very small, but does increase as the number of linear encoding sites in the network increases. This probability can be made smaller by increasing the size of the field of operation. We could do this by using a Galois Field GF(2^{16}), but then we would have to account for the increased cost of computation in the larger field. The second issue is the problem of including the encoding vectors in the packets transmitted. The encoding vectors should be included in the packets to facilitate decoding at the receiver. However, this results in a small overhead on the available bandwidth. For example, for packet size of around 1000 bytes and GOP size of 64 descriptions (or packets), the overhead for including the encoding vectors would be around 6 %. It is important to see the effect of these two factors on the PSNR of the received video. To this effect, we compare the performance of random codes versus RS codes on a single link for different link loss rates. We choose a source and a receiver connected by a 1000 Kbps link, and the loss rate is varied from 0.10 to 0.20, in steps of 0.02. We perform the simulations using RS codes and random codes by transmitting the *Foreman* sequence (CIF, 30 fps, 297 frames, 16 frames per GOP), and plot the results in Fig. 2. The results shown are obtained after averaging over 10 simulations. We observe that the average PSNR obtained using random codes is lower than that obtained using RS codes. The difference is less than 0.5 dB, and this difference is the result of both the overhead due to the encoding vectors, and the probability of non-decodability. Since the drop in performance is not significant, we will neglect the overhead of encoding vectors for other results in the paper. Effectively, we ignore the bandwidth for transmitting the encoding vectors.

To get a clear understanding of the relation between the encoding vector overhead and the packet size, we will have to look at both source and channel together. The total number of bytes that can be transmitted in one second in the link will depend on the available link bandwidth. This would translate to a certain number of bytes for each GOP. Given the GOP size (frames per GOP) and the frame rate, the maximum packet size will depend on the number of descriptions that the GOP will be coded into. We chose 64 descriptions for the above example, because that resulted in a packet size of more than 1000 bytes. If we choose too few descriptions, the packet size may exceed the maximum transmitted unit (MTU) for that network (1500 bytes for the Ethernet). Now, if we do want to have fewer descriptions in a GOP and still not exceed packet size, the GOP size will have to be reduced. This may not necessarily be good because as the GOP size decreases, the compression performance of the scalable coder could decrease. Based on the requirements of a problem, these parameters need to be chosen for the particular transmission. For the example above, for 64 descriptions, 16 frames per GOP, and 30 fps, the average packet size came out to be 1041 bytes. This results in an overhead of 6.1 %.

3.3 MD-FEC for single receiver, multiple links

MD-FEC described in Sec. 3.1 is designed for optimal rate allocation for a single link. One way to extend MD-FEC to larger networks is by performing a hop-by-hop MD-FEC, where the optimization is performed for each hop of the network. The packets arriving at the end of every link are decoded and then again encoded (called transcoding) based on the characteristics of the next link.

Transcoding at every intermediate node may be computationally very intensive. Instead, we could develop an end-to-end scheme. We can approximate the characteristics of the network by a single link between the source and each receiver, if we can find an effective bandwidth and loss rate that can adequately summarize the network between the source and each receiver. By studying the overall flow to each of the receivers, the effective bandwidth to the receiver can be found. To find the loss rate, we need to look carefully at the transmission scheme that we are working with. First we consider routing, and let us consider the network given in Fig. 3 as our example. S is the source, I is an intermediate node and R is the receiver. All packets are only routed. Let the bandwidths of the two links be identical and the loss rates be p_1 and p_2, The effective loss rate as seen by the receiver would be $1-(1-p_1)(1-p_2)$. This can be similarly extended to find the effective loss rate of more than two links, joined end-to-end. In case the flow of packets splits into two paths, the effective loss rate of the two paths together can be obtained by adding the weighted sum of the effective loss rates of each flow path, the weights being the flow in the respective paths. Thus, the effective loss rate between the source and any receiver can be found.

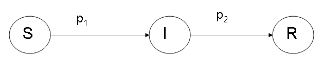


Figure 3. Network with one source and receiver, connected through an intermediate node.

Let us consider the same network in Fig. 3, but employ practical network coding at the nodes. Here again, the effective bandwidth is obtained by the flow to each receiver. Assume that source S sends N packets to node I. Some of the packets may be lost, and only N_1 ($N_1 \leq N$) packets may arrive at I. The packets sent by I to the next node are linear combinations of the received N_1 packets. If we restrict the number of packets sent from I to R to be N_1, the effective loss rate seen by R would be the same as seen during routing. However, we can send more linear combinations of packets from I, as long as the rate of our transmission does not exceed the bandwidth of the second link. This is made possible by the fact that we have included the encoding vectors in each packet. Therefore, as more linear combinations of packets are created at the intermediate node, the encoding vectors also are updated. The additional packets sent from I would act as additional protection for the data. Thus, losses in the two links do not add up as in the case of routing. The effective loss rate of two or more links joined end-to-end would be close to the maximum of the individual link loss rates. If the flow is split into two paths, the loss rate can be obtained by taking a weighted sum of the effective loss rate of each path (similar to routing).

3.4 Joint MD-FEC Network Coding solution

We have seen that MD-FEC can be used to optimize the bit rate allocation such that the expected distortion of received video over a lossy link is minimized (or the expected PSNR is maximized). Network coding says that multicast capacity of the network, equal to the capacity of the minimum cut, can be achieved. It would be advantageous if we can combine the two schemes into one, resulting in a robust joint source-channel-network coded video multicast.

Assuming network coding is to be used, the effective bandwidth and loss rate of each receiver can be found (Sec. 3.3). If all the receivers are symmetric with respect to the source (they have symmetric paths), then there would be no conflict of resources for the various receivers (using network coding). However, if different receivers have different effective loss rates, the distribution to be used at the source for the MD-FEC algorithm (q_k) would have multiple choices. In such cases, the effective loss rate could be set as that seen by the maximum number of receivers, or the maximum effective loss rate observed among all the users, or the distribution of number of descriptions can be chosen to be a weighted average of the distributions of all the users. In the first case, users with high loss rate would be affected if their request is not accommodated. In the second case, the users experiencing smaller effective loss rate would be affected because now the amount of parity bytes in the descriptions is more than what is required by them. The third choice may satisfy a significant number of users, however its effect on the overall performance of the system needs to be studied. The choice of the distribution can be made by the source based on the requirements of the multicast. In this work, we assume that the max-flow capacities of all the receivers are the same.

The protection for the packets would be provided by random codes - hence the parity byte locations at the source would be filled with zeros. Random codes, thus, serve two purposes - first, to protect the packets from losses, and second, to enable network coding at the nodes. The bitstream corresponding to each GOP is grouped into one generation. Packets sent by the source are linearly combined with random coefficients. The intermediate nodes also combine packets belonging to a generation, and the receivers decode the source data from the packets they receive. The main issue that needs to be addressed is the effect of linear combination at intermediate nodes on the MD-FEC coded packets. Packets sent out of the source are obtained by linear combination of the source data packets. When packets of a generation are coded at the intermediate nodes, the resultant packet is another instance of linear combination of the source data packets. The receiver will, therefore, not be able to distinguish whether the packets were coded at the source or re-encoded at an intermediate node. Thus, network coding does not affect the MD-FEC coding of packets.

4. RESULTS

We used the highly scalable and efficient MC-EZBC coder[16] to encode the Foreman sequence (CIF, 30 fps, 297 frames, 16 frames/GOP). We simulated the video multicast on the butterfly network of Fig. 1 and modeled each link of the network as a packet erasure channel with a bandwidth of 500 Kbps. We used 64 descriptions to code every GOP; each description of a GOP is sent in one packet, in which the GOP number is included. We carried out MD-FEC optimization to maximize the expected PSNR at the receivers. As the two receivers of the network are symmetric, both have the same effective bandwidth and loss rate. Therefore, we do not have to make a decision about which effective loss rate to use at the server for the MD-FEC optimization. All values were obtained by averaging across 10 simulations. The error concealment strategy used at the receiver was a combination of frame freeze and half-second dissolve. We performed the multicast using three transmission schemes:

1. Only routing - in this method, intermediate nodes are allowed to only forward the data. Nodes cannot replicate the packets. The resultant would be two tree networks, one tree for each of the receivers. The bandwidth supported by each tree is 500 Kbps.

2. Routing with replication - in this method, intermediate nodes are allowed to forward packets, as well as replicate them. For our network, each receiver has two paths by which packets arrive - one path with a bandwidth of 500 Kbps (same as the case of only routing), and the other path with a bandwidth of 250 Kbps.

3. Network coding - in this method, intermediate nodes are allowed to do practical network coding. Packets are transmitted through all the links. Note that this is not the solution obtained by using Fig. 1(b), but rather by performing random network coding at all the intermediate nodes. This helps us in reducing the effect of packet losses in the network.

For each of the three schemes, we perform the MD-FEC optimization for various design loss rates. We assume that the distribution with the design loss rates is binomial, and this is used for the MD-FEC optimization. We show that the design loss rate that maximizes the average PSNR is very close to the effective loss rate (Sec. 3.3).

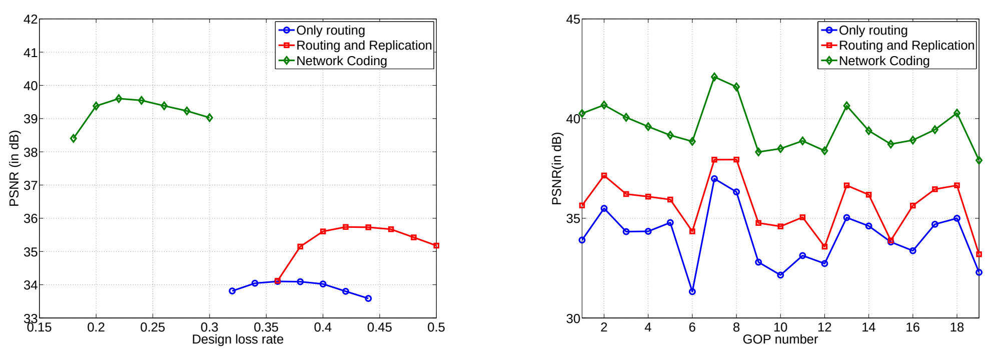
4.1 Implementation in NS-2

We performed all simulations on the network simulator NS-2.[17] In order to run practical network coding, we wrote applications that can be used with NS-2 to simulate an application layer video multicast. The source application codes the input data and sends them along the outgoing links. The applications running on intermediate nodes combine incoming packets belonging to one generation and send the linearly combined packets to the next downstream node. The receiver application decodes packets belonging to a generation to obtain the source video bitstream.

4.2 PSNR comparison of network coding and routing

We vary the link loss rate to study the performance of the three methods. We choose loss rates of 0.20, 0.10, and 0.01 for the links of the butterfly network. We perform the simulations using all three methods.

Fig. 4(a) shows the performance for a link loss rate of 0.20. We varied the design loss rate in steps of 0.02 and found out the "best" loss rate, i.e., the one that maximizes the average PSNR per run obtained at the receivers (averaged over the two receivers). For all the schemes, when the loss rate is less than the "best" loss rate, the average PSNR falls sharply. At lower design loss rates, channel losses are more than what the packets have been protected against, and hence the channel distortion prevails. At higher design loss rates, the channel induced distortion is almost negligible and the source coding distortion becomes more prominent, indicated by the gradual decrease in the average PSNR. This is due to the fact that protection given to the packets is much more than required. For the given network and at the "best" design loss rate for each scheme, we find that network coding improves the average PSNR of the receiver by about 5 dB in comparison with only routing. Network coding also performs better than routing with replication - we observe an improvement of around 3.5 to 4 dB. We would like to remind the reader that for network coding, we have neglected the overhead involved in including the code generator vectors. If we include the overhead, the average PSNR obtained for network coding would be less by around 0.5 dB (as shown in Fig. 2). Thus the improvement of the network coding scheme over routing with replication would be 3 to 3.5 dB. Fig. 4(b) shows the variation of average PSNR with GOP, for the three schemes at their "best" loss rate. The improvement in the PSNR for network coding is due to two reasons - due to the higher effective bandwidth seen by each receiver from the source, and second, due to the additional descriptions that are generated by the intermediate nodes that reduce the effect of lost packets. Both the above factors result is a higher capacity to the receiver, and thus account for the improvement in the observed PSNR. Typical sequences obtained for the "best" loss rates for the three schemes are available in our website.[18]



(a) (b)

Figure 4. Results for the *Foreman* sequence for link loss rate of 0.20 in the butterfly network. (a) shows variation of avg. PSNR with modeled loss rate, and (b) shows variation of avg. PSNR with GOP for the "best" loss rate for the three schemes.

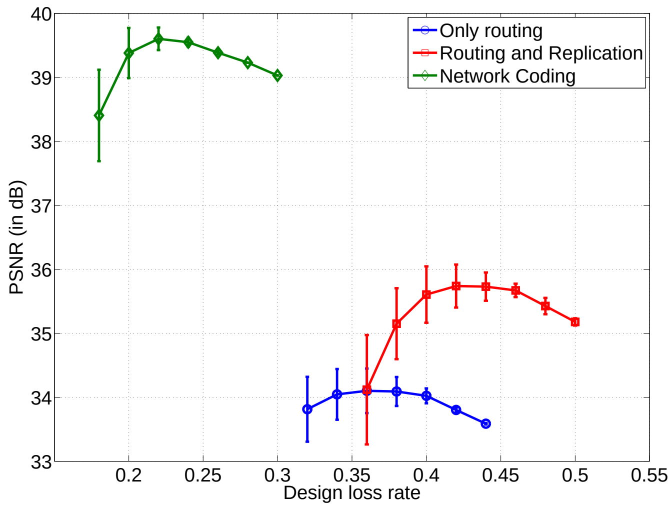
We find that the PSNR values for the routing-only scheme and routing (with replication) peak at modeled loss rates of around 0.36 to 0.38, and 0.42 to 0.46, respectively. For routing-only scheme, effective link between the source and each of the receivers can be modeled using an erasure link with loss rate $1-(1-0.2)^2$, which comes out to be 0.36. The simulation results give the "best" loss rate close to this value. For the case of routing (with replication), the flow of packets to each receiver is through 2 paths, one with an effective loss rate of 0.36 (two links) and the other with an effective loss rate of $1-(1-0.2)^4 = 0.59$ (four links). However, the flow in the first path is 500 Kbps, and in the second path, it is 250 Kbps. After scaling the loss rates with the flow, we obtain the effective loss rate to be 0.44. Here again, the simulations give the optimal loss rate for the design close to this value. For the case of network coding, we expect the effective loss rate to be around 0.2. Using simulations, we obtain the peak value at a modeled loss rate of around 0.22. The difference could be due to the fact that the modeled loss rate is binomial, whereas the actual distribution of the packets is not exactly binomial. However, the results obtained for design loss rate of 0.20 also yields an average PSNR quite a bit larger than that obtained by routing (with or without replication).

To provide a fuller picture, we re-plot the average PSNR obtained along with standard deviation in Fig. 5(a). We observe that the standard deviation of the average PSNR value per run decreases as we increase the design loss rate. When the design loss rate is low, the chances of being unable to channel decode a GOP is high and there is a lot of variation in PSNR of the received video across simulations. As the design loss rate gets higher, MD-FEC provides more protection for the data and so less number of packets are required to decode an entire GOP, and hence almost always no source data is lost. Therefore, at higher design loss rates, the standard deviation is low. This is applicable to all the three transmission schemes. In Fig. 5(b), we show the variation of average PSNR for the "best" loss rate for each of the three schemes, across the GOPs. The dotted lines indicate the expected PSNR given by the MD-FEC algorithm for a single link with the same bandwidth and loss rate equal to the corresponding "best" loss rate. We notice that the average PSNR obtained is very close to the expected PSNR given by the design.

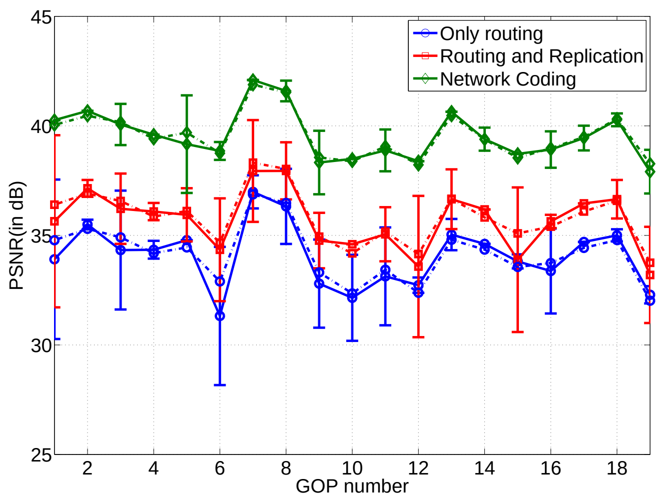
We also tested the *Mobile* sequence (CIF, 30 fps, 16 frames per GOP, 300 frames) using the three schemes for a link loss rate of 0.20. We plot the average PSNR per run at the receivers, averaged over both the receivers, in Fig. 6(a). Again, we notice that the "best" loss rates obtained for the three schemes are similar to those obtained for the *Foreman* sequence. The average PSNR for the "best" loss rate for network coding is around 3 to 3.5 dB more than routing with replication, and around 5 dB more than routing without replication (after taking into account the encoding vector overhead). Fig. 6(b) shows the variation of the average PSNR of the received *Mobile* sequence at the "best" loss rate for the three schemes across different GOPs.

Fig. 7(a) and Fig. 8(a) show the average PSNR obtained at the receivers when the link loss rate is 0.10 and 0.01, respectively. We observe that improvement in the PSNR of the network coding scheme in comparison to routing (with replication) is around 2 to 3 dB, lower than that observed for a loss rate of 0.20. At a loss rate of 0.10, the effective loss rate seen by the receivers decreases in all the three schemes. However, the decrease is more for the routing schemes (with or without replication) when compared to the network coding scheme. Moreover, the PSNR - source rate curve is concave and this means that the difference in the PSNR for the same increase in source rate is greater for lower source rates than for higher source rates. The "best" loss rate for network coding was obtained to be around 0.12 (close to the expected 0.10). For routing (with replication), the "best" loss rate was obtained to be 0.25 (close to the expected 0.24) and for routing (without replication), the "best" loss rate was obtained to be around 0.18 to 0.20 (close to the expected value of 0.19). Fig. 7(b) shows the variation of the average PSNR per run for each GOP at the best loss rate for the three methods at link loss rate of 0.10.

As we go to a loss rate of 0.01 (can be considered to be very close to lossless links), we see that the curves are closer, and the main contribution to the higher performance of network coding is the higher max-flow capacity. The network coding scheme has an average PSNR of 1 dB more than routing with replication (after taking into account the overhead of encoding vectors). The effective loss rate for network coding, routing (with replication), and routing-only methods was obtained to be 0.02, 0.03, and 0.02 (again close to the expected values), respectively. The effect of additional descriptions (linear combinations) at the intermediate nodes does not have much of an effect on the performance, because not many packets are lost at this loss rate. Fig. 8(b) shows the variation of the average PSNR for each GOP at the best design loss rate for link loss rate of 0.01 for the three methods. We would like to mention here that Sarshar and Wu[8] described a method of transmitting multiple description coded video in lossless links. The butterfly network at link loss rate of 0.01 would be quite close to a lossless network. The solution provided by Sarshar and Wu (which involves routing with replication) would yield an average PSNR of around 40.3 dB, which would still be below our solution by around 0.5 dB. Moreover, when we implemented network coding using fixed codes (as in Fig. 1(b)), average PSNR obtained



(a)



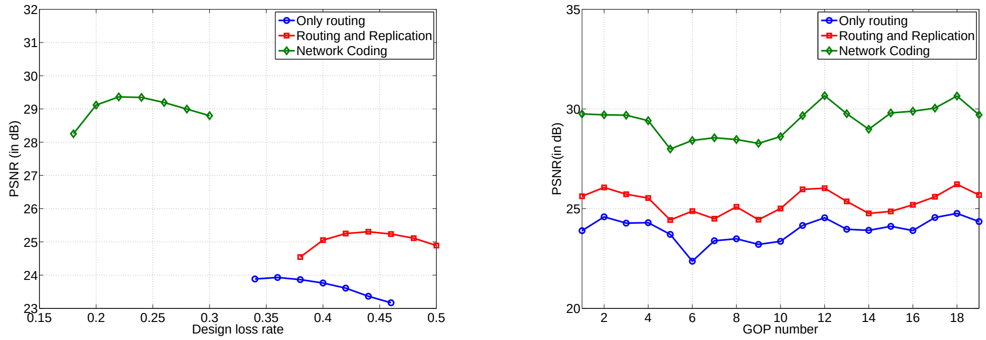
(b)

Figure 5. Results for the *Foreman* sequence for link loss rate of 0.20 in the butterfly network. (a) shows variation of avg. PSNR, and its standard deviation, with modeled loss rate, and (b) shows variation of avg. PSNR with GOP (with standard deviation) for the "best" loss rate for the three schemes. The solid lines indicate the observed PSNR, and the dotted lines indicate the expected PSNR according to the MD-FEC design.

was 41.9 dB. For large networks, finding fixed random codes for all the intermediate nodes requires a lot of computation, and hence a randomized approach (as PNC) is favoured.

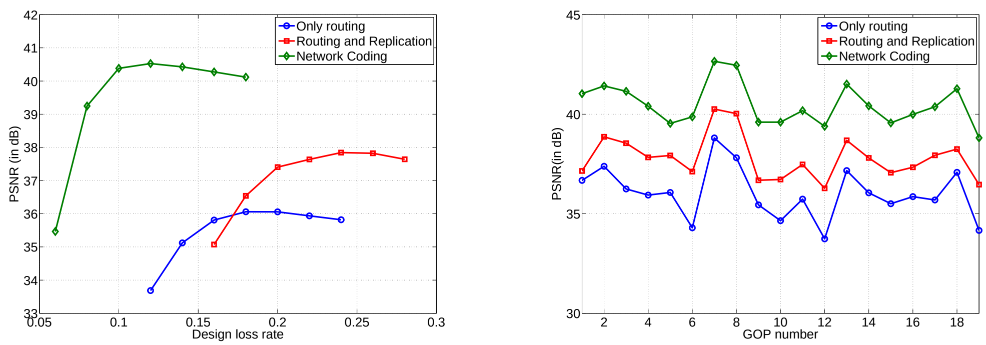
5. CONCLUSIONS

In this paper, we show that there is significant advantage in using practical network coding and multiple description codes for video multicast in lossy packet networks. We observe that compared to traditional routing methods, network coding provides significant improvement in the PSNR of video at the receivers. The improvement is two-fold - one due to increase in the effective bandwidth seen by the receivers, and second, due to the reduction in effective loss rate seen by the receivers (due to the additional linear combinations created at intermediate nodes). The improvement in the PSNR obtained at the receiver using network coding in comparison to



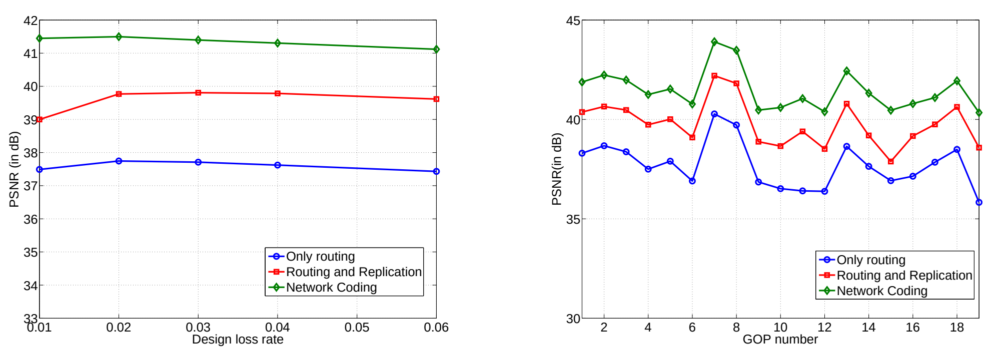
(a) (b)

Figure 6. Results for the *Mobile* sequence for link loss rate of 0.20 in the butterfly network. (a) shows variation of avg. PSNR with modeled loss rate, and (b) shows variation of avg. PSNR with GOP for the “best” loss rate for the three schemes.



(a) (b)

Figure 7. Results for the *Foreman* sequence for link loss rate of 0.10 in the butterfly network. (a) shows variation of avg. PSNR with modeled loss rate, and (b) shows variation of avg. PSNR with GOP for the “best” loss rate for the three schemes.



(a) (b)

Figure 8. Results for the *Foreman* sequence for link loss rate of 0.01 in the butterfly network. (a) shows variation of avg. PSNR with modeled loss rate, and (b) shows variation of avg. PSNR with GOP for the “best” loss rate for the three schemes.

routing becomes larger as the loss rate of the links increases. Using random coding, we find that the effective loss rate seen by the end-receivers can be reduced without channel decoding and re-encoding at the intermediate nodes. Network coding also obviates the need for constructing multiple multicast trees, and hence the construction of the system is greatly simplified. We further show that the MD-FEC optimization can be performed in conjunction with network coding in lossy networks, thus maximizing the expected PSNR of the video at the receivers. Further results and comparisons on larger networks are being pursued.

REFERENCES

1. 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.
2. 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.
3. 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.
4. R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory* **46**, pp. 1204–1215, July 2000.
5. 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.
6. 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.
7. 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.
8. N. Sarshar and X. Wu, "Rate-distortion optimized multimedia communication in networks," in *Visual Communications and Image Processing*, January 2008.
9. J. Cannons, T. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity," *IEEE Transactions on Information Theory* **52**, pp. 777–788, March 2006.
10. 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.
11. Y. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications* **22**, pp. 107–120, January 2004.
12. S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory* **49**, pp. 371–381, February 2003.
13. P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proc. of the fifteenth annual ACM symposium on Parallel Algorithms and Architectures*, pp. 286–294, 2003.
14. T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of the IEEE International Symposium on Information Theory*, p. 442, 2003.
15. P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *51st Allerton Conf. Communication, Control and Computing*, October 2003.
16. S. Hsiang and J. Woods, "Embedded video coding using invertible motion compensated 3-d subband/wavelet filter bank," *Signal Processing: Image Communication* **16**, pp. 745–762, May 2001.
17. "The network simulator, ns-2; http://www.isi.edu/nsnam/ns/."
18. "www.cipr.rpi.edu/∼ramasa/vcip09/."