Comparison of Multiple Description Coding and Layered Coding based on Network Simulations

Raghavendra Singh, Antonio Ortega, Lionel Perret and Wenqing Jiang,

Integrated Media Systems Center
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-2564
E-mail:{rasingh,ortega,lperret wqjiang}@sipi.usc.edu

ABSTRACT

Layered coding has been proposed as a method of "quality adaptation" for the Internet's best effort service model. The disadvantage of layered coding is that if the base layer packets are lost, the enhancement layers are rendered useless. To achieve error free transmission of the base layer, ARQ could be used, but this limits the performance of LC, due to the strict timing constraints of real time transmission. In this paper we compare Multiple Description Coding, an alternative scalable scheme, without retransmission to Layered Coding with retransmission for a wide range of scenarios. These scenarios include network with no feedback support, networks with long RTTs (WANs) and applications with low latency requirements.

1. INTRODUCTION

It has been widely recognized[1] that with the Internet's best effort service model, the quality of service provided by the network changes over time and it is up to the applications to adaptively operate over a wide range of quality of service. One popular method for "quality adaptation" is Layered Coding (LC), where the base layer is sent first and then enhancement layers are added if the bandwidth is available. Other methods for adjusting quality of the pre-encoded data are either more complex, e.g. transcoding, or require higher storage space, e.g. keeping several versions of the data at different qualities.

An example of Layered Coding (LC) is Receiver driven Layered Multicast (RLM) developed by McCanne et al[2] for video conferencing over Multicast networks. Here LC is used to cope with receiver heterogeneity, as the receiver individually adapts its reception rate by adjusting the number of layers that it receives. Another example is the recently proposed protocol for real-time streams over the Internet, Rate Adaptation Protocol (RAP),[3] which is an end-to-end TCP friendly protocol, employing Layered Coding for quality adaptation at the sender.

A disadvantage of layered coding schemes is that if, due to channel noise, the base layer is not received correctly, then the enhancement layers are rendered useless because of their dependence on the base layer. Increasing congestion on the Internet means that this scenario could happen often, at least as long as there is no priority-based infrastructure to deliver the base layer error free. Given this lack of guarantee on quality of service on today's Internet, reliable protocols based on ARQ, e.g. TCP, or error recovery techniques, e.g. Forward Error Correction codes, are used. In ARQ the delay associated with retransmission, in order to get the base layer across error free, could be prohibitive, especially for real-time applications with low latency requirements. Forward Error Correction codes are often used but their disadvantage is that they do not provide graceful degradation in case of severe channel noise.

A new approach to reliable transmission, Multiple Description Coding (MDC),[4] has become popular for real time applications as it provides for graceful degradation and does not require retransmission. In MDC two or more descriptions of the source are sent over different channels to the receiver, as shown in Fig.1. While initial applications of MDC considered physically separate channels, channels such as Internet can also be modeled in the same way. The underlying assumption is that after appropriate interleaving, one can think of virtual channels, one per description, each suffering from independent and random losses.

If only one channel is received, the decoder can reconstruct the signal to a minimum level of distortion. However, if both channels are received, information from one channel augments information from the other and it is possible to achieve a lower level of distortion than with a single channel. Thus, MDC is **robust** due to the redundancy of the multiple descriptions of the same source and it is **scalable** as each correctly received description improves the decoder performance. Also MDC **does not require prioritized** transmission, as each description is independently decodable.

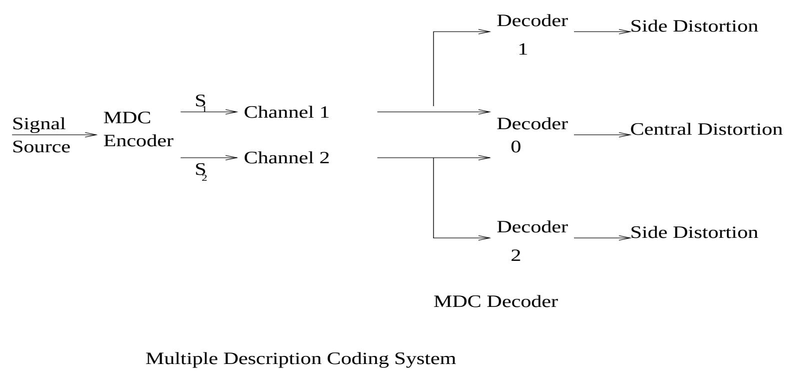

Figure 1. The MDC system diagram.

This paper compares the Layered Coding approach to Multiple Description Coding for real-time transmission, by simulating different protocols and network topologies, using the network simulator *ns*.[5] Though there has been a lot of research on coding techniques for MDC, this paper is the first that actually simulates network conditions to show situations where MDC is useful. Reibman et al[6] have analyzed and compared LC to MDC, for Binary Symmetric Channels and Random Erasure Channels. Unequal error protection (UEP) is provided for LC, using FEC. The main difference between our work and theirs is that we base our results on simulations of a delay constrained network environment, where different qualities of service is not available and transmission robustness is achieved through retransmission, rather than UEP.

There are many parameters which affect real-time transmission. Some are network dependent, e.g., feedback support, RTT of the link, while others are requirements of the end application, e.g. latency. The network might not support feedback or feedback may be expensive, e.g in point to multipoint transmission (broadcast/multicast) retransmission requests can cause NAK implosion. The transmission delay is another factor which is network dependent, the RTT of the link between sender and receiver is the dominating factor in determining the delay, especially for WANs. The delay that can be absorbed in a real-time transfer is dependent upon the application, e.g. video playback from a server can support high latency, while video conferencing requires low latency/delay.

In this paper, we vary the parameters mentioned above (feedback, RTT, target latency) and show that MDC outperforms LC over a broad range of scenarios. We establish that if no feedback is used, performance of LC, although better than MDC for error free conditions, rapidly deteriorates in the face of packet loss. This is especially true if the base layer (BL) packets are lost, thus reliable transmission of BL is necessary for LC to perform well. To achieve reliability we propose to use ARQ, i.e., BL is transmitted by TCP, while the enhancement layer (EL) is sent using UDP. We compare this to MDC, where both descriptions are sent using UDP. In an experiment, RTT of the network is varied and the results show that LC with ARQ, performs better than MDC with no ARQ, only for short RTTs. Even for short RTT, another experiment shows that, if the latency requirement of the application is very low, MDC outperforms LC.

Papadopoulos et al,[9] were the first to propose that ARQ, with some modifications, could be used for continuous media transmission. Another reason for popularity of ARQ, is that it can be totally reliable, given sufficient time for transmission, while this not true of FEC. FEC also suffers from the disadvantage that it has to be applied assuming the worst case channel scenario, and in a heterogeneous network this could be wasteful. Thus, many of the current reliable techniques use ARQ either by itself or as an hybrid with FEC. Our retransmission results (short RTT, high latency), match those reported by Papadopoulos et al, the novelty of the paper is that that an alternative scalable scheme namely, MDC, has been shown to perform reliably without requiring retransmission.

Our results do not match those obtained by Reibman et al, who indicated that MDC was preferred only for very high erasure and bit error probabilities. We attribute this discrepancy to the fact that Reibman et al used FEC for reliable transmission, while we use ARQ for reliability. Also their assumption of a packet size of 47 bytes is unrealistic, overhead cost of transmitting these packets would be quite large.

Section 2 of the paper reviews previous work in reliable real-time transmission. Section 3 formulates the problem and introduces the system architecture. Section 4 discusses the experiments in detail. We conclude with some observations in Section 5.

2. PREVIOUS WORK

This section starts by describing selective retransmission, an ARQ scheme which is used in many reliable real-time protocols. In particular, we review various approaches for reliable multicast, because ARQ presents some additional difficulties in multicast environment. Thus the need for a scalable scheme, that does not require ARQ, is even more obvious for multicast situations. SR has also been proposed as the error control scheme for reliable unicast transmission alongside RAP.

2.1. Selective Retransmission

Traditional ARQ techniques are based on the Go-Back-N (GBN) approach, where the sender buffers all packets that it has transmitted, until their respective ACK is received. If a NACK is received, or if there is a timeout, the packet is retransmitted. These properties make protocols like TCP totally reliable (given sufficient time for retransmission). As the delay associated with GBN is high, an alternative retransmission scheme called selective retransmission (SR) has been developed by Papadopoulos et al,[9] and the idea has been explored by other researchers (e.g. Limited retransmission for real-time layered media by Podolsky et al[10]).

In SR, the loss is detected by a gap in sequence number, which packets carry in their headers, and a NACK is send to the sender. If the sender estimates that the packet can be received by the receiver before its playback time, it retransmits the packet. Also, the sender buffer is updated at fixed intervals and if the requested packet is not in the buffer it cannot be considered for retransmission. Thus SR has a lower delay than GBN, but it does not guarantee total reliability.

SR was used by Papadopoulos et al, to show that retransmission is feasible for continuous media applications provided the RTT of the link is small, (e.g, LANs ad MANs), and especially if receiver buffering is used.

One of the drawbacks of SR schemes is the complexity in estimation of RTT for multicast situations, especially when using only NACKs.[11,18] RTT estimation is important because the decision to retransmit and the update of server buffer are dependent upon the transmission delay and the receiver state. Also, there could be a constraint on the receiver buffer size, either due to the latency requirements of the application or the available memory at the receiver, which would limit the time available for retransmission.

2.2. Reliable Multicast

Many reliable schemes have been proposed based on FEC, ARQ or hybrid of FEC and ARQ, details can be found on the reliable multicast website.[8]

In multicast networks, feedback leads to a number of additional issues.[20] Solutions for some of these individual problem exist, but no comprehensive system has yet been proposed which solves all the problems. For example Pejhan et al,[11] propose sending retransmission requests to the sender, who upon receiving a request immediately multicasts the repair packet. The advantage of this scheme is quick recovery time, essential for real time data, but the main disadvantage is that in a large multicast group there will be a NAK implosion at the sender.

To overcome this implosion, local recovery (all or some of the receivers can retransmit packets), with randomized timers, has been proposed in Scalable Reliable Multicast (SRM).[12] Any of the receivers in the group can retransmit the packet, but to avoid an implosion of requests or repair packets on the network, random timers are used. These add to the delay in retransmitting packets and make the approach less attractive for real-time transmission. Another reliable multicast scheme is the Real Multicast Transport Protocol (RMTP),[13] which has been used along with SR, in Layered Video Multicast with Retransmission,[19] a reliable layered coding scheme. To avoid randomized timers designated receivers (DR) are used to assist the sender in processing ACKs and in retransmitting data, but finding "good" designated receivers adaptively is still an open problem.

Another reliable layered coding scheme, which does not use retransmission, but is related to our MDC approach was proposed by Turletti et al.[18] In this work a polyphase transform is used for producing independent layers for audio coding. Each layer contains one polyphase decomposition of the original sequence. If only one layer is subscribed to, then the other polyphase components are estimated by interpolation. Each additional layer improves the signal quality. The polyphase components are transmitted using RTP, and robustness to packet loss is achieved by sending two polyphase components in the first layer. Their experiments, done over the actual MBONE network, are meant to show the effectiveness of congestion control using a layered approach. The scheme is similar to ours in that each polyphase component can be thought of as a description, but they dont send any redundant information. Another important difference between their work and the work in this paper is that they have not compared their scheme to a traditional layered approach in a lossy packet environment.

3. SIMULATION ARCHITECTURE

The above discussion has highlighted some of the problems that arise in SR schemes, in particular when these are used in a multicast environment. We propose to solve these problems by using redundant source coding (MDC), so as to make retransmission unnecessary to achieve reliable transmission. Using network simulations, we find different conditions under which MDC, performs better than LC.

3.1. Source Coding

We have implemented a MDC scheme based on a balanced MDC approach, proposed by Jiang et al.[15] In our experiments, each frame of the real-time stream is transmitted in JPEG format. To generate the multiple descriptions, the frames are DCT transformed and the transformed coefficients are quantized using two quantizers, high resolution (HR) and low resolution (LR). The quantizers differ in the step size, Qp_{hr}, Qp_{lr}, used in the standard JPEG quantizer. The quantized coefficients in each block are then polyphase transformed along their zig-zag index, as shown in Fig.2. The underlined coefficients are the odd polyphase components of a transformed block. The odd coefficients of HR and the even coefficients of LR form Description 1 (D1). Description 2 (D2) contains even HR coefficients and odd LR coefficients.

Each description is packetized into different packets and the packets are transmitted with interleaving to make the probability of loosing both the packets, low. We have investigated different interleaving techniques and in the simulations, alternate 16 D1 packets with 16 D2 packets. If both the packets are received correctly, odd HR and even HR coefficients are picked up from their respective descriptions and the image is JPEG decoded to HR. If only one packet is received, then the reconstruction algorithm uses the single description in the packet. For example if description 1 is received, a preprocessor would multiply, the entropy decoded, LR even coefficients by Qp_{hr}/Qp_{lr} and feed the coefficients into a standard JPEG decoder. The decoding would be to a quality which is in between the qualities achievable for LR and HR.

The layers are generated by using a progressive JPEG coder, shown in Fig.3. The base layer and the enhancement later are sent in different packets, if a base layer packet is lost, the low frequency coefficients are assumed to be zero and the high frequency coefficient are decoded from the enhancement layer packet. If both packets are lost, all coefficients are assumed to be zero; this is also true in the MDC case.

An example of Lenna coded with our LC and MDC scheme is shown in Figs. 4 & 5, respectively. As can be seen, the visual and MSE distortion of either description is acceptable, while if the base layer is not received in LC, the enhancement layer is visually useless.

3.2. Network Simulation

We simulate the network by using *ns*, with the topology shown in Fig.6. Transmitter T_0 and receiver R_0 are used to send and receive data, while the other transmitters and receivers are used for background traffic. In the table below, the parameters for the bottleneck link and the background sources are listed.

As shown in Fig.6, 10 sources are initialized to generate background traffic in our simulations. 8 of these sources are TCP sources, 5 of which carry a variable amount of data, while 3 carry a fixed amount of data. 2 sources are CBR, all the sources start and stop randomly and the side delay for these sources are also randomly generated.

In our experiments we assume that each frame, of size 512x512, is packetized into 64 packets containing 4096 pixels each. Thus one packet contains a description of four 8x8 blocks. The frame is coded according to the LC and MDC coding schemes described above. The total bit-rate of LC and MDC is always 1.8 bpp, but we have four different schemes, parameters of which are given below. For schemes 1 and 4, each packet of each layer or description has a size of 500 bytes. In schemes 2 and 3 each description is sent using 500 bytes packets but packets for BL and EL are 600 and 400 bytes respectively, because of the higher bitrate of the BL. In schemes 3 & 4, HR is coded at a higher quality than schemes 1 & 2, but total bit rate is the same for all cases.

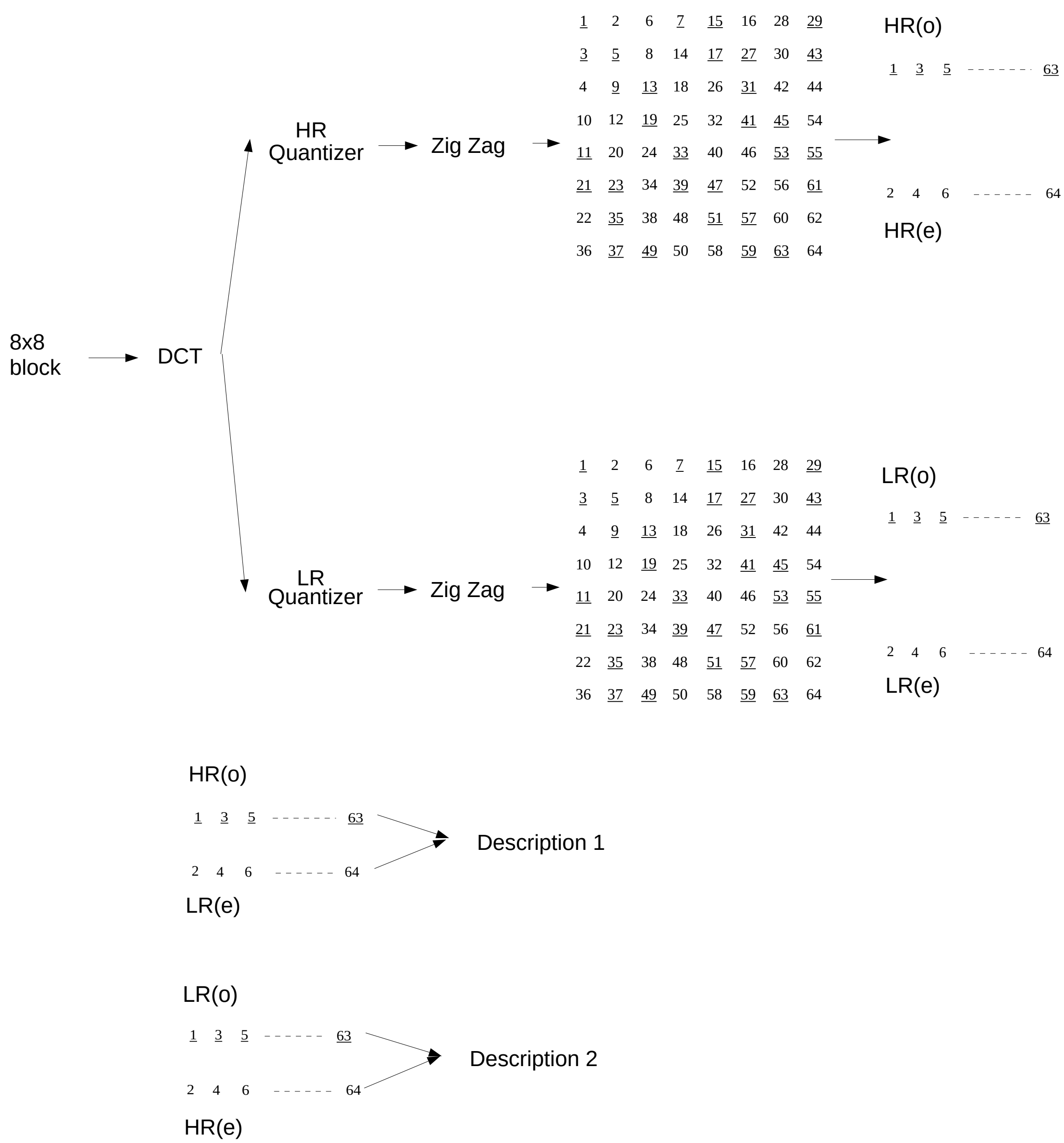

Figure 2. The proposed MDC system based on DCT transform coding. An 8x8 blocks of image data is DCT transformed. It is quantized by the High resolution and Low resolution quantizers and zig-zag indexed for entropy coding. Polyphase transform along the zig-zag index separates the blocks into even and odd (underlined) coefficients for each of the quantizer blocks. The even coefficients of HR are packetized with odd coefficients of LR to form Description 1.

4. EXPERIMENTAL RESULTS

Each experiment has been performed 200 times and the results reported are averaged over all the iterations. Though we are performing experiments with a single frame, the philosophy of the experiment is easily extended to the multiple frame case.

4.1. Experiment 1: All layers and description transmitted via UDP

In this experiment, MDC and LC are sent using UDP with no protection. The first 64 packets are BL/D1 and the last 64 packets are EL/D2. Results, Fig.7, show that for total packet loss of about 1%, or for base packet loss (losses

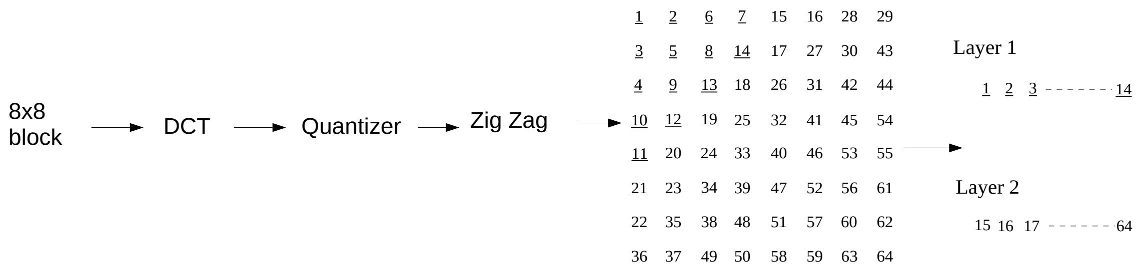

Figure 3. The proposed LC based on Progressive JPEG, the underlined coefficients are the low-frequency coefficients packed in the base layer.

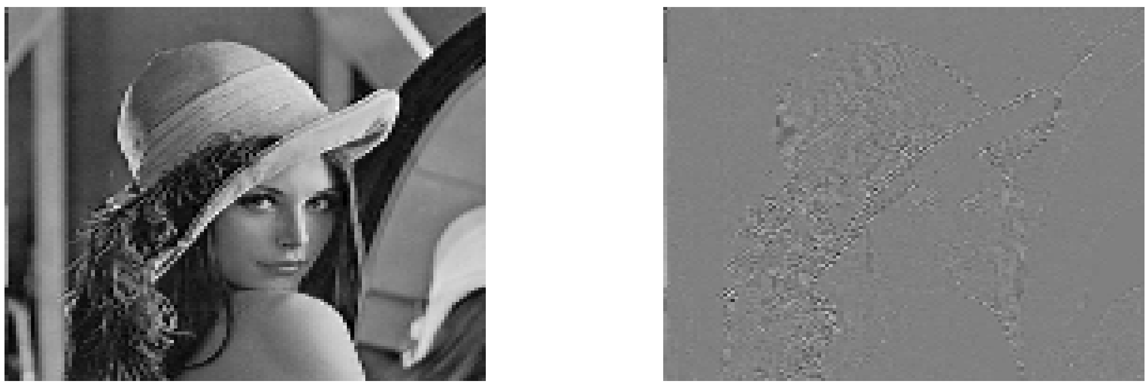

Figure 4. Base Layer MSE =59.06, Enhancement Layer MSE =3589.1, Total Bpp =1.8, Total MSE =8.8506, MSE is w.r.t original image

in the first 64 packets) rate of 0.15%, MDC outperforms LC. The need for reliable transmission of BL is immediately obvious from this experiment.

4.2. Base Layer sent using TCP, Enhancement Layer and All Descriptions sent using UDP

As explained in the introduction we to use ARQ to provide reliable transmission. Thus BL is transmitted using TCP and EL/D1/D2 are transmitted using UDP. In real-time streams, data has to be played back continuously at fixed time intervals, e.g., for jitter free television broadcast, 30 frames must be displayed in a second. This implies that there are strict timing constraints on the delivery of data. To simulate these constraints we define a timeout T at the receiver. If a frame starts transmitting at T_{start}, the frame is played at $T_{start} + T$ with the packets that have been received between T_{start} and $T_{start} + T$. T is dependent upon the receiver playback rate and the receiver buffer size and reflects the needs of the application (e.g in a video conferencing application T will be smaller than for video playback from server).

Parameter	Value
Bottleneck Link Delay	20ms
Side Bandwidth	100 Mbps
Bottleneck B/W	5 Mbps
Bottleneck Queue	50 packets
Bkgnd Packet Size	500 bytes
Bkgnd CBR Rate	0.6 Mbps
Bkgnd TCP Window Size	40 packets

Table 1. Simulation Parameters

Figure 5. Description 1 MSE = 22.16, Description 2 MSE= 22.01, Total Bpp= 1.79, HR MSE= 14.39, MSE is w.r.t. original image

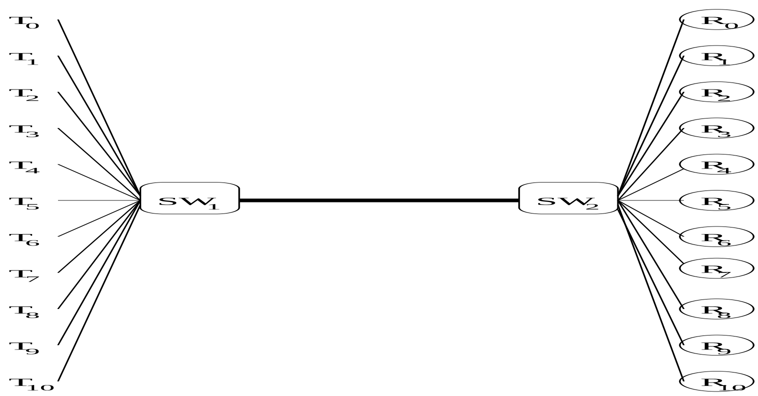

Figure 6. Network Topology, T_i are the transmitter, R_i are the receivers. The link between switches $SW1$ and $SW2$ is the bottleneck link and $SW1$ is the bottleneck point.

Ideally we would like to receive both the layers (and descriptions) before the time $T_{start} + T$. To ensure that BL is received reliably, the sender sends BL first. If there are packets dropped in BL, the sender retransmits them and it has up to $T_{start} + T$ to get BL across. If BL gets across before $T_{start} + T$, the remaining time is used to send EL using UDP.

Using the same topology and the same background sources (we record the start/stop times of the background sources used in LC simulation) we also send MDC with UDP. At end of the T sec. duration, we play back both LC and MDC with the packets that have been received and compare the MSE of the reconstructed frame.

Scheme	Parameters	BL	EL	LC	HR	LR	D1/D2
1.	Bpp	0.88	0.91	1.81	1.17	0.53	0.89
	Mse	59.06	3584.2	8.85	14.39	29.25	22.16
2.	Bpp	1.03	0.89	1.81	1.17	0.53	0.89
	Mse	52.56	3589.2	8.85	14.39	29.25	22.16
3.	Bpp	1.03	0.89	1.81	1.31	0.3	0.89
	Mse	52.56	3589.2	8.85	12.80	136.25	90.35
4.	Bpp	0.88	0.91	1.81	1.31	0.3	0.89
	Mse	59.06	3584.2	8.85	12.80	136.25	90.35

Table 2. Packetization Scheme

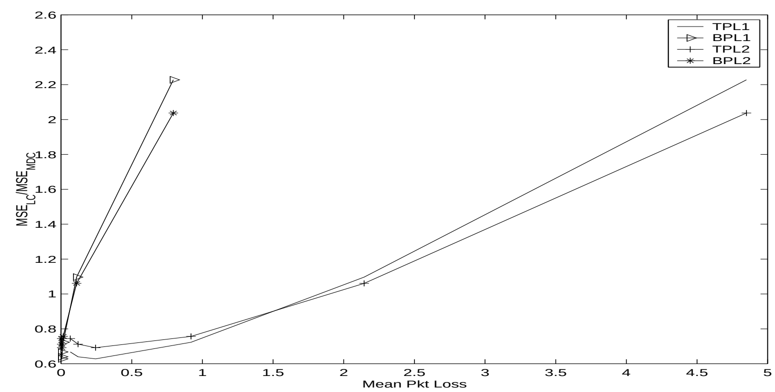

Figure 7. Experiment 1: MDC and LC sent by UDP, TPL stands for total packet loss. BPL is the packet loss for the base layer (first 64 packets), TPL1/BPL1 are results using Packetization scheme1, TPL2/BPL2 are for scheme 4. For very low losses (0.15%) in the base layer MDC does better than LC.

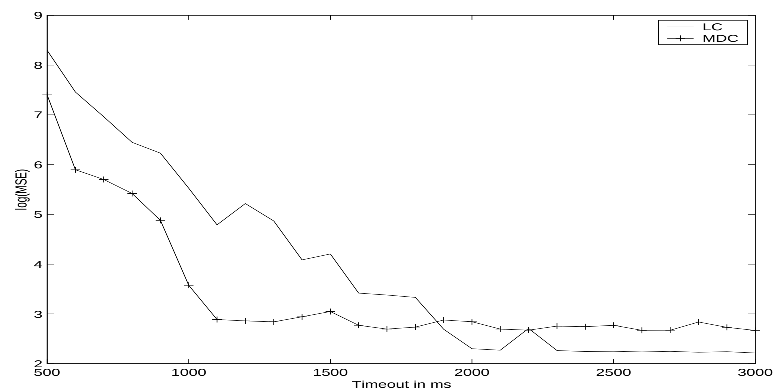

Figure 8. Experimet 2: BL sent using TCP, after all BL packets received, if time is leftover from Timeout, EL is sent using UDP. MDC is sent using UDP. The RTT has been optimized, so as to ensure that all packets are received in 1sec. if there is no congestion. After adding congestion, the results show that MDC can achieve good performance close to 1sec. but LC requires 1.6 sec to achieve similar performance.

4.2.1. Experiment 2: Performance changes under different latency requirements

Given WindowSize of TCP and the BufferSize, we have set a side delay for our network, such that if there is no congestion all 128 packets of LC/MDC would arrive in 1sec. Then we add congestion and vary T. The results are shown in Fig.8. MDC achieves acceptable performance ($MSE < 50$) very close to $T = 1sec$ but LC can achieve this performance only at $T > 1.6sec$. This shows that if an application has a low latency requirement, MDC would be a better alternative to LC.

4.2.2. Experiment 3: Sensitivity to variations in RTT

In the experiment above, given the bottleneck link and TCP parameters, we optimized the RTT of the network so as to ensure that if no congestion occurs, a frame (with both layer/descriptions) can be transmitted in 1 sec. This optimization is obviously not possible in actual conditions. In this experiments we simulate a more realistic environment where we vary the RTT of the network (delay between T_0 R_0 in Fig.6), while setting the timeout to be 1sec. If there was no congestion, both MDC and LC would get 128 packets across, for the RTTs that have been used for the experiment. The results in Fig.9, show that for longer RTT, because of retransmission delay, LC performs worse than MDC. Also, if BL is enhanced the performance of LC degrades (LC2). Note, the delay shown on the x-axis of the figure is the delay of the link between $SW1$-T_0 which is equal to delay of $SW2$-R_0.

For the above experiment we used the default parameters of TCP connection. We have done another experiment where we have started TCP at steady state and used a window size of 32 packets. The results are shown in Fig.10.

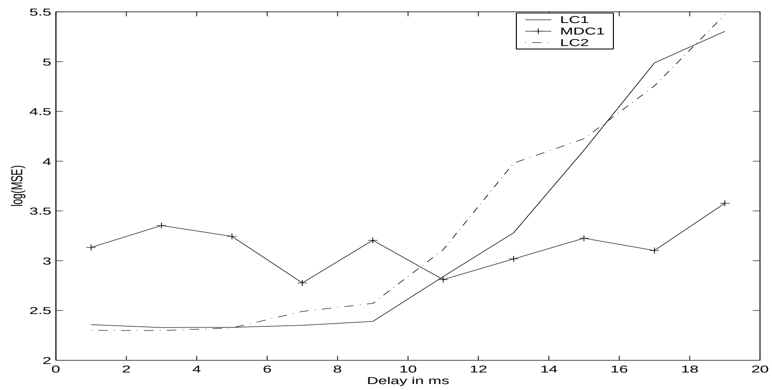

Figure 9. Experiment 3: RTT between sender and receiver is varied, LC1, MDC1 is coded at Scheme 1. LC2 is packetized with Scheme 2. If there was no congestion LC would perform better than MDC for all RTT, if there is congestion than MDC performs better for long RTTs

The only change is that MDC performs better than LC at a higher RTT than in Fig.9. The conclusion from these experiments is that MDC will perform better than LC over a wider range of network RTTs, i.e. for both LANs and WANs. In Fig.10 we also show the performance of a single description (SDC) to establish the scalability of MDC.

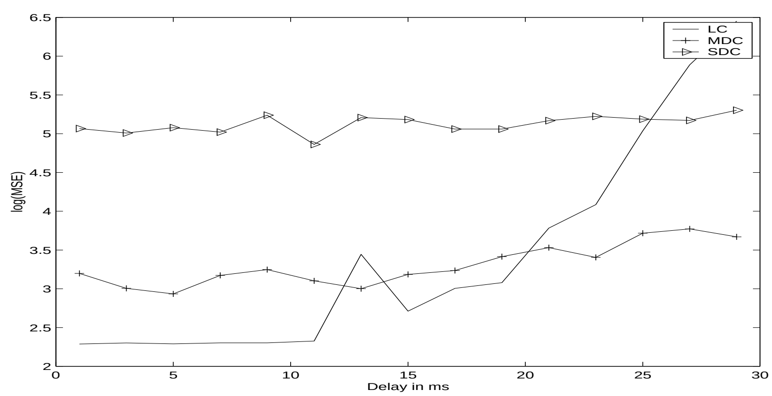

Figure 10. Experiment 3: TCP is started at steady state, with the WindowSize set at 32 packets. SDC is the MSE with only one description

4.3. Experiment 4: Comparing BL robustness via MDC or via TCP

In the above experiments we are comparing the robustness of two scalable schemes at the same bitrate, one in which layers are dependent (LC), other in which they are independent (MDC). But MDC could also be used as a means of robust transmission of the BL, replacing ARQ. Thus in this experiment we multiple description code BL, i.e. the polyphase of low frequency coefficients of the BL are put into two descriptions and a low resolution quantized version of these coefficients is added to the descriptions. This is exactly the same operation as in Fig.2, except that the 8x8 block is from the base layer, not from the original image. These multiple descriptions of the base layer are sent across using UDP and compared to reliable transmission of BL using TCP. The network has been optimized to ensure that 128 packets of BL-MDC and 64 packets of BL-TCP are received in 1sec. Then the timeout is varied, the results are shown in Fig.11. Again MDC is better for shorter latency applications. This experiment shows that instead of using TCP for BL we could use MDC to provide robustness.

4.4. Experiment 5: MDC with 3 descriptions

This experiment is along the lines of the reliable layered multicast protocol proposed by Turletti.[18] Here we generate 3 independent descriptions using 3 polyphase and 3 quantizers. The bit rate of the LC scheme and the MDC scheme

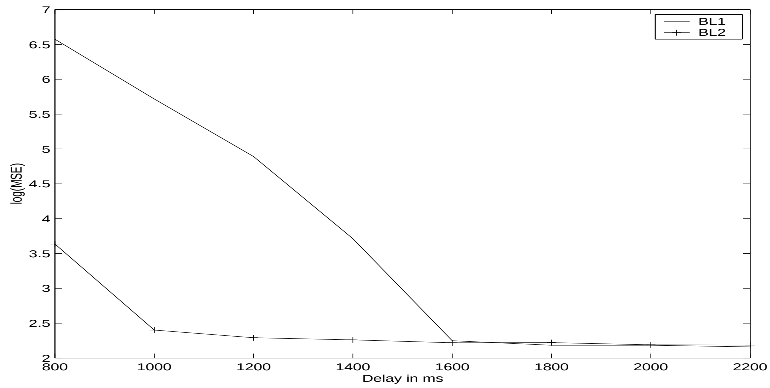

Figure 11. Experiment 4: BL1:- BL sent using TCP. BL2:- BL coded into two descriptions and sent using UDP. Each packet of BL1 is 600 bytes, each packet of BL2 is 400 bytes, there are 64 packets in BL1 and 128 packets in BL2. Redundancy in BL2 is coded at 0.19 bpp, while BL1 is coded at 1.17 bpp

is the same, BL is sent by TCP while EL and the descriptions are sent using UDP. 2 descriptions could be send to the first multicast group to achieve reliability while the other description is sent to a different group, which a receiver will subscribe to if it has the available bandwidth. In Fig.12, results with 2 descriptions (SDC) and with all three descriptions are shown. Note that with each added description, the robustness to loss also improves.

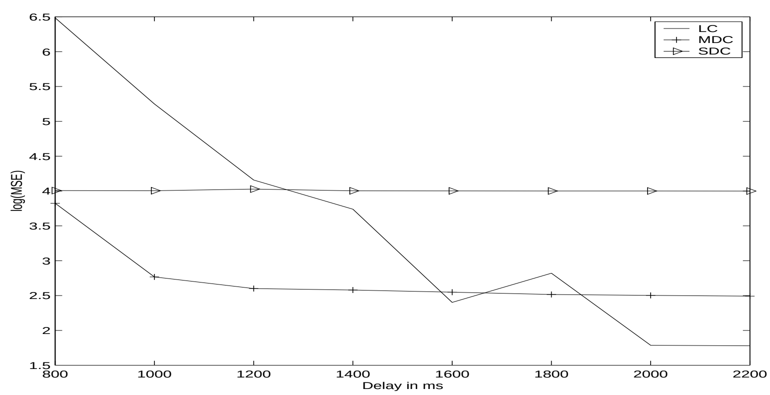

Figure 12. LC is BL sent using TCP, EL sent using MDC. In MDC scheme 3 polyphase transform generate 3 descriptions. MDC is the result when all three are being sent, SDC is the result when only 2 are being sent.

5. CONCLUSION

This paper shows that, as was to be expected, LC needs reliable transmission of its base layer for it to perform well. If reliable transmission is performed using ARQ, besides the special problems of multicast, there are general constraints which would limit its use for real-time transmission. Though the results will change with the use of SR instead of TCP, we believe that these constraints still hold true. MDC on the other hand is a scalable, reliable transmission scheme which does not require retransmission, thus its performance will be better over a wider range of network scenarios.

Inspite of the higher cost of sending redundant data, MDC in a way is already being used over the Internet. A scheme for reliable audio, Reliable Audio Tool (RAT),[22] uses a method of adding redundancy which is similar to the proposed MDC approach. In their scheme, error control is performed by sending a low quality redundant information of the data after some time lag. Also there are commercial vendors, e.g. Real Audio, who store data at different resolutions. If these resolutions are packetized together, then they can generate descriptions described in our MDC scheme.

REFERENCES

1. R. Rejaie, M. Handley, D. Estrin, "Quality Adaptation for Congestion Controlled Video Playback over the Internet", in *Proc. of ACM SIGCOMM*, 1999.
2. S. McCanne, M. Vetterli, V. Jacobson, "Low Complexity Video Coding for Receiver driven Layered Multicast", *IEEE Journals on Selected Areas in Communications* . pp. 983-1001, Aug. 1997.
3. R. Rejaie, M. Handley, D. Estrin, "RAP: An End to end Rate based Congestion Control Mechanism for Real-Time Streams in the Internet", in *Proc. of INFOCOM*, 1999.
4. A.A. El Gamal, T.M. Cover, "Achievable rates for multiple descriptions", *IEEE Trans. Inform. Theory*, vol. 28, pp. 851-857, Nov. 1982.
5. VINNT Project, *http://netweb.usc.edu/vint*.
6. A.R. Reibman, H. Jafarkhani, M.T. Orchard, Y. Wang, "Performance of Multiple Description Coders on a Real Channel", in *Proc. of ICIP*, 1998.
7. D. Clark and D. Tennenhouse, "Architectural Considerations for a New Generation of Protocols", in *Proc. of ACM SIGCOMM*, 1990.
8. Reliable Multicast links, *http://research.ivv.nasa.gov/RMP/links.html*
9. C. Papadopoulos, G.M. Parulkar, "Retransmission Based Error Control for Continuous Media Applications", in *Proc. of NOSSDAV*, 1996.
10. M. Podolsky, M. Vetterli, S. McCanne, "Limited Retransmission of Real-Time Layered Multimedia", in *Proc. of MMSP*, 1998.
11. S. Pejhan, M. Schwartz, D. Anastassiou, "Error Control Using Retransmission Schemes in Multicast Transport Protocols for Real Time Media", in *IEEE/ACM Transactions on Networking*, Vol. 4, pp. 413-427, June 1996.
12. S. Floyd, V. Jacobson. C-G. Liu, S. McCanne, L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", in *IEEE/ACM Transactions om Networking*, Dec. 1997.
13. J.C. Lin, S. Paul, "RMTP: A Reliable Multicast Transport Protocol", in *Proc. of INFOCOM*, 1996.
14. X.R. Xu, A.C. Myers, H. Zhang, R. Yavatkar, "Resilient Multicast Support for Continous Media Applications", in *Proc. of NOSSDAV*, 1997.
15. W. Jiang, A. Ortega,"Multiple Description Coding via Polyphase Transform and Selective Quantization", *Proc. of VCIP* 1999.
16. R. Singh, A. Ortega,"Erasure Recovery in Predictive Coding Environments using Multiple Description Coding", *Proc. of MMSP*, 1999.
17. V.A. Vaishampayan, S. John, "Interframe Balanced Multiple Description Video Compression", in *Proc. of PVW*, 1999.
18. T. Turletti, S.F. Parisis, J-C. Bolot, "Experiments with a Layered Transmission Scheme over the Internet", in *INRIA Research Report No 3296*, Nov. 97.
19. X. Liu, S. Paul, P. Pancha, M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes", in Proc. of NOSSDAV, 1997.
20. Hoffman, Nonnenmacher, Rosenberg, Schulzrinne, "A Taxonomy of feedback for Multicast, in *IETF Internet drafts, draft-hnrs-rmt-avt-feedback-00.txt* , June 1999.
21. H. Schulzrinne, S. Casner, R. Frederick. V. Jacobson, "RTP: A Transport Protocol for Real Time Applications", in *RFC 1889*, Jan. 1996.
22. V. Hardman, M.A. Sasse, M. Handley, A. Watson, "Reliable Audio for use over the Internet", in *Proc. of INET*, 1995.