Comparing Layered Coding and Multiple Description Coding with Link Diversity

Jinhui Pan (Jinhui@stanford.edu)
Computer Science Department
Changhua He (changhua@stanford.edu)
Electrical Engineering Department
Stanford University

Abstract

For video server, providing optimal service to highly diversified users is challenging because different users have different link conditions and different requirement of quality. Considering the low price of storage, we propose to store multiple video streams in different bit rate, different coding strategies and different formats for the same video sequences. Server can choose a certain video stream to transmit according to link condition. Multiple paths are used to increase the available bandwidth and resist single link loss. In this paper, we compare two coding approaches for multiple paths: Multiple Description Coding (MD) and Layered Coding (LC). Results show that MD is suitable for symmetrical links while LC is good in asymmetrical links with one of them in low loss rate to transmit base layer. Simulation results also support that if server can adapt its bit rate and coding strategies to diversified link condition and feedback, as much as 2dB improvement in video quality can be achieved.

1.Introduction

For video server, providing optimal service to highly diversified users is challenging because different users have different link conditions and different requirement of quality. In order to provide satisfying service, transcoding approaches are studies to transfer one video format into another one or one bit rate into some other one ([1][2]). However, transcoding may suffer from its complexity and hard to be real time. Thus it adds extra delay in the real time video streaming and much more expense in the video server. Considering the lower and lower price of storage, we propose to store multiple video streams for one single video sequence in hierarchal bit rates, different coding approaches and different formats. Besides diversified bit rates, the server can also utilize multiple links to the same user to increase the video quality and resist single link loss. Multiple path transport has been proposed to for wired network to increased available bandwidth by using multiple links. This scheme can also improve the reliability for transmitted data.

Multiple Description Coding (MD) and Layered Coding (LC) have been two efficient coding approaches to multiple paths transmission ([3][4]). Instead of simply separating one-layer video stream onto multiple paths, MD and LC provide much better error resilience than one-layer coding over multiple paths. For one-layer coding, packet loss may cause error propagation that sharply degrades the quality of reconstructed video. However, the different descriptions of MD are complementary. Packet loss in one description can be recovered from other descriptions as long as the same parts of the multiple descriptions are not lost simultaneously. For LC, different protection can be adopted for the base layer and the enhancement layer. In LC, base layer is very important that we can use Forward Error Coding (FEC) or Automatic Retransmission reQuest (ARQ) to support reliable transmission of base layer.

Generally speaking, MD and LC are suitable for different link condition. For MD, multiple description play the same important role and generally with same bit rate. So symmetric links with similar bandwidth and loss rate are more suitable for MD. For LC, however, the base layer is much more important than the enhancement layer: error in base layer will propagate to the enhancement layer. In addition, the base layer generates higher bit rate than the enhancement layer. Thus, in LC, a link with a higher bit rate and lower loss rate should be assigned to the base layer while the enhancement layer can be transmitted on link with a lower bit rate and a higher loss rate.

There are some research studies comparing MD and LC ([5][6][7]). R. Singh et al. present a comparative study of MD and LC in a wide range of scenarios of network condition, such as network without feedback, with long RTTs or low latency requirement ([5]). The simulation results show that MD outperforms LC over a wide range. The study is complementarily considering many different scenarios. But the codec used in the study is actually a JPEG based one, which is accurate only for I frame. A. Reibman et al. conduct a study of MD and LC over an Enhanced General Packet Radio Services (EGPRS) wireless network ([6]). Using a Cumulative Distribution Function (CDF) instead of traditional average PSNR as the criteria for assessing the quality of video streaming, the authors conclude that when two links are available, splitting one-layer video onto the two links gain better performance than both MD and LC. The results also show that MD may be advantageous when the loss rate is sufficiently high. Y. Wang et al. compare the performance of MD and LC over general wireless networks ([7]). Simulation results demonstrate that LC+ARQ is the best strategy; the second one is MD, which is around 1db better than LC without ARQ.

In this paper, we compare MD and LC in different bit rates. Simulations with controlled loss rate and real network setting are presented. Results show that MD is suitable for symmetrical links while LC is good in links with one of them in low loss rate to transmit base layer. Simulation results also show that if server can adapt its bit rate and coding strategies to diversified link condition and feedback, as much as 2dB improvement in video quality can be achieved.

2.Methodology

In this paper, we compare the Multiple Description coding (MD), Layered Coding (LC) and Single-layer Coding (SC). MD and LC provide better error resilience than SC when the loss rate of the link is high. However, SC can achieve higher coding efficiency without overhead, thus providing a better quality when the loss rate of the link is low. In the simulation, we assume that two links between the client and the server are available. The two links are of different characteristics. They vary from symmetric links to links with very different loss rate. From the simulation, we derive the characteristic of the links that are suitable for transmission of MD, LC coded video stream separately.

For MD, we adopt the MD which divides the source video sequence into odd and even frames and compress them separately. With the same bit rate, MD results are lower in quality than SC because MD does not utilize the redundancy of the video sequence as much as SC.

For LC, we use the SNR scalable coding in H.263+([8]). Each frame will generate one base layer and one enhancement layer. The base layer is quantized with a coarse quantization parameter while the enhancement layer plus the base layer can provide a finer video quality of the video. The two layers are transmitted through two links, with base layer through the one with less loss rate or with retransmission; and enhancement layer through the link with higher packet loss rate.

We simulate using two kinds of scenarios. First, we compare MD and LC with controlled loss rate. Under this situation, loss rate is specified by ns2 loss model. No background traffic is considered in order to keep the loss rate predictable. However, in the second scenario, packet loss is caused by congestion and the rate of loss is not predictable. Background traffic is added and playback delay is taken into consideration.

3.Simulation & Results

Standard video sequence foreman.qcif with size 176*144 and format 4:2:0 is used in the simulation. For all coding strategies, bit rate is around 100kbps and 500kbps.

The bit stream of each frame is coded into three packets. For lost packet, the corresponding packet from the last available frame is copied for error concealment. Average PSNR over 10 runs of simulation is used as the criteria of the decoded video sequences.

The detailed simulation topology and scenario are described in Appendix C. We use M/Pareto process to simulate the long-range-depended traffic in the networks, as in ([9])

3.1. Comparison of MD and LC (Different loss rates without background traffic)

In this simulation, we control the loss rate of links with ns2 loss model. No background traffic is added and no delay is concerned. For LC, we always use link with lower loss rate to transmit base layer. So the loss rate can be read as (loss rate for base layer, loss rate for enhanced layer) for LC and (loss rate of description1, loss rate of description2) for MD. Results are shown in Figure1 and Figure 2.

From the results, we can see that with similar loss rate, LC out performs MD because MD decrease the correlation of the frames. In LC, the loss rate of base layer will degrade the video quality much more than loss of enhanced layer. For LC, it is obvious that asymmetrical link with base layer in link with loss rate is better than symmetrical. For example, in LC100k, PSNR is 28.96 for (1%, 5%) and 28.86 for (3%, 3%).

For MD, it works better in symmetrical links than asymmetrical links. For example, in MD100k, PSNR of (3%, 3%) is better than (1%, 10%) and (1%, 15%).

From Figure1 and Figure2 we can conclude that MD performs better in symmetrical links and loss rate of base layer degrade the video quality more than loss in enhanced layer. But we can not show the advantage of MD compared with LC in our simulation. Using a better designed MD may change the results ([10][11]).

Figure1: Performance of LC and MD(bit rate: 100kbps)

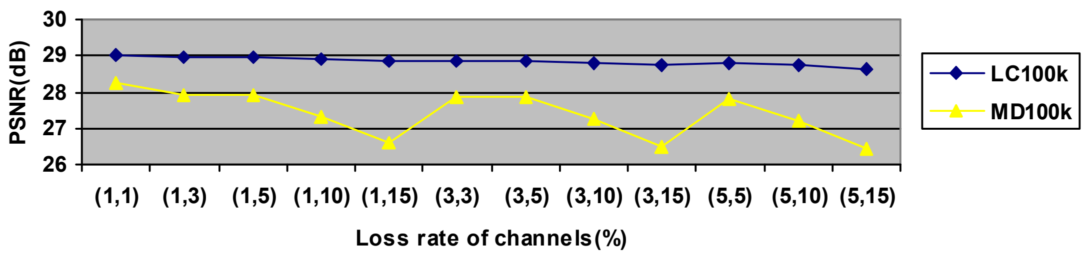
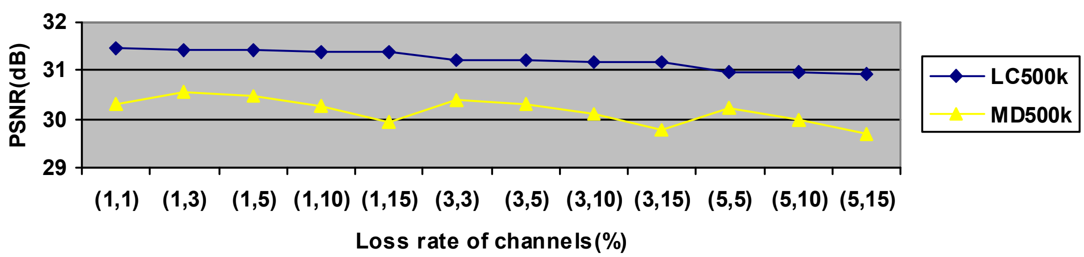


Figure2: Performance of LC and MD(bit rate: 500kbps)



3.2. Comparison of different bit rate of MD and LC (Different loss rates without background traffic)

Network simulation settings are the same as above. However, this simulation compares the video quality of different bit rate under different loss rate. Results are shown in and Figure3 and Figure4. The results illustrates that high bit rate video streams out perform low bit rate streams when loss rates for both are low. However, with the increasing of loss rate, high bit rate video stream is much worse than low bit rate video stream in low loss rate. To achieve best performance, bit rate must be chosen according to the load and available capacity of the links

3.3. Delay consideration with Background traffic

The link capacity is 1Mbps, link delay is 50ms, link loss rate is 3%. Background traffic is 5000 CBR flows, the flows arrival is Poisson and the flow size is Pareto distributed with the shape 1.1 and mean 10 packets. The packets received with delay larger than 120ms will be considered as dropped packets.

Figure3: Comparason of 100kbps and 450kbps (LC)

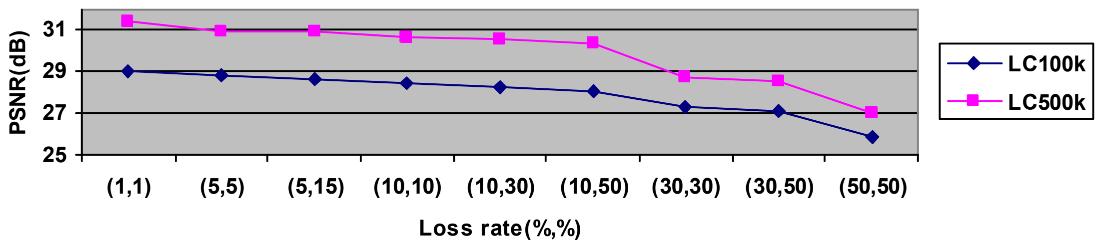
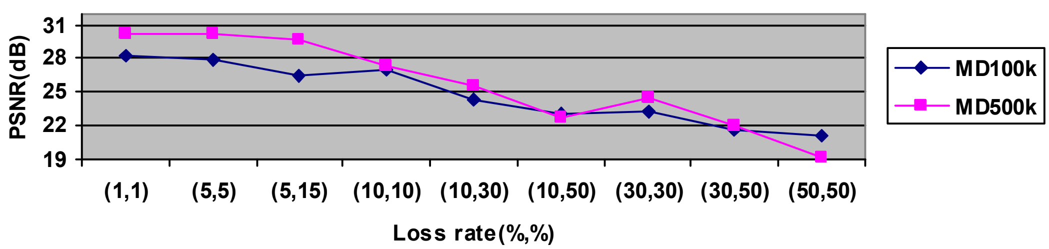


Figure4: Comparason of 100kbps and 450kbps (MD)



.

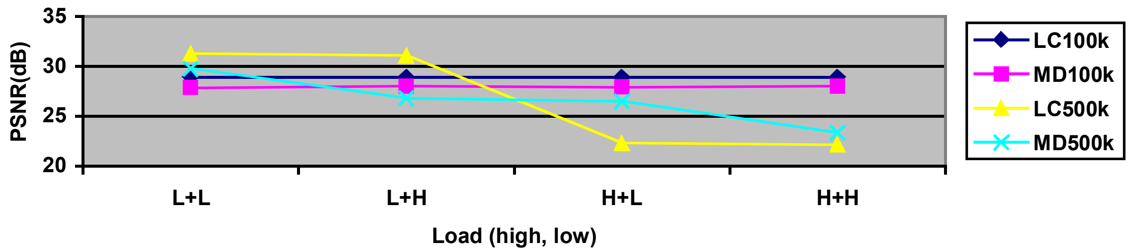
For Link 1, the background traffic has average load 0.2 and data rate 0.2 which corresponds to a low load situation, while for link 2 the average load is 0.8 and the data rate is 0.05, which corresponds to a high load situation.

Results are shown in Table1 and Figure5. “L” means low load and “H” represents high load. As we can see in the table, when link load is low, loss rate for both 100kbps stream and 500kbps stream are both very low, thus bit rate with 500kbps is much better than 100kbps. When the load is very high, loss rate for 500kbps increased greatly (>50%), the quality of video sequence degrades to unacceptable zone. This simulation, different from controlling loss rate of the video traffic, simulates the real network condition and obtains the similar results as experiment of controlling loss rate.

		L+L	L+H	H+L	H+H
100k	LC	28.89	28.88	28.89	28.89
	ULC	28.76	28.76	28.77	28.76
	MD	27.82	27.98	27.90	28.00
500k	LC	31.26	31.07	22.29	22.15
	ULC	31.28	31.28	22.10	22.10
	MD	29.72	26.78	26.46	23.34

Table 1

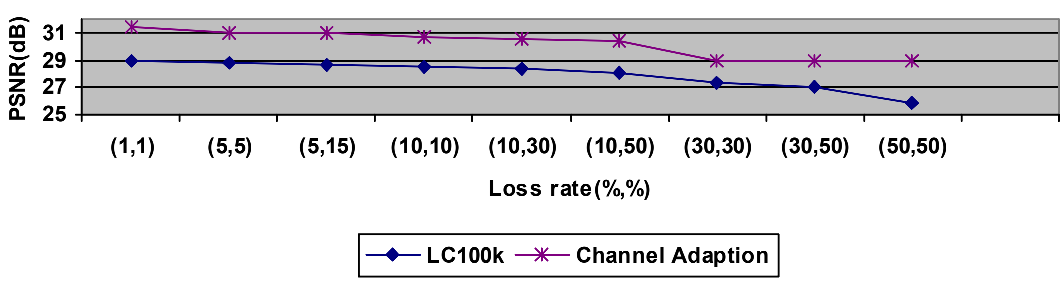
Figure5:LC and MD in real network simulation



3.4 Adapt bit rate to link condition

Based on the above simulations and results, we show the result of adapting bit rate according to link condition. Figure6 shows that if sever store two streams with bit rate 500kbps and bit rate 100kbps and chose video stream according to link feedback, it can obtain as much as 2dB improvement compared to single 100kbps stream.

Figure6: Comparason of 100kbps and 450kbps (LC)



4.Conclusion and future work

In this study, we propose to store streams for one single video sequence in the video server to satisfy different requirement from diversified users. Unlike transcoding approach, this proposal is simple, fast and economical. Compared complex transcoding algorithm and expensive transcoding server, this hierarchical storage proposal utilize cheap storage and achieve diversified service. To support our proposal, we compare different coding strategies (MD and LC) in different bit rates. For MD, simulation results show symmetrical links are much more suitable for MD. While for LC, loss in base layer affects the quality of video much more than loss in enhancement layer. However, we can not show the advantage of MD over LC in our simulation. This is relevant to the characteristics of MD. MD can resist loss in a single link, while it decrease correlation among frames thus reduces the coding efficiency. Adopting a more efficient MD or allowing a little higher bit rate for MD compared to LC may change the results.

We also compare performance of LC and MD with different bit rates. As demonstrated in the simulation results, in order to achieve optimal performance, video streams with diversified bit rate should be chosen according the link capacity, link load and delay. All these factors change the loss rate of received video sequences. So the video server should be able to choose streams of different bit rates adapting to loss rate feedback. Both simulation with controlled loss rate and simulation with background traffic and delay constraints support our conclusion.

Based on the comparison, when multiple links are available, the video server can choose the suitable stream to deliver, which will improve the performance of the system. Assume for one specific video file, the server stores several encoded versions with Layered Coding and Multiple Description Coding with different data rate level. Simply the server can decide to transmit which stream to achieve the best performance according to the link quality, which indicates the current load and the link loss rate, and the link difference. In the real networks, it is easy for the server to know the average link error rate and the average queue length*,* thus the link quality can be calculated

The future work includes two parts. First is to do a more through comparison of more video format and bit rate. For example, besides low bit rate H.263 video, the video server should also support video in other formats, such as MPEG-2 and MPEG-4 and, bit rate around 20 Mbps. The second part is to find the formula to calculate the link quality and to find the optimum data rate levels, which will be a tradeoff between the performance and the storage space.

Acknowledgement

Special acknowledgement goes to Jiang Zhu and Sang-eun Han for helpful discussions and useful information. Also acknowledgements go to Alexander Fishk and Professor Tobagi for motivating and defining the scope and depth of the project.

References

1. S.J. Wee, J. Apostolopoulos and N. Feamster, "*Field-to_Frame Transcoding with Temporal and Spatial Down Sampling,",* IEEE ICIP'99
2. N. Bjork and C. Christopoulous, "*Transcoder architectures for video coding*," IEEE ICIP'98
3.J. Apostolopoulos, W.Tan, S. Wee and G.W. Wornell, " *Modeling Path Diversity for Multiple Description Video Communication*" , IEEE Intern Conf. Acoustics, Speech and Signal Processing, 2002
4. Lallet, A. Dolbear, C. Hughes, J. Hobson, P, "*Review of scalable video strategies for distributed video applications* ", Distributed Imaging (Ref. No. 1999/109), IEE European Workshop, 1999 Page(s): 2/1 -2/7
5.R. Singh, A Ortega, L.Perret and W.Jiang, "*Comparison of Multiple Description Coding and Layered Coding Based on Network Simulation*", VCIP 2000
6. A. Reibman, Y.Wang, X. Qiu, and Z. Jiang, "*Transmission of Multiple Description and Layered Coding Video over EPGRS Wireless Network*", IEEE Inter. Conf. Image Processing, 2000
7. Y. Wang, S. Panwar, S. Lin, and S. Mao, "*Wireless Video Transport Using Path Diversity: Multiple Description vs. Layered Coding* ",Image Processing. 2002. Proceedings. 2002 International Conference on , Volume: 1 , 2002

8. ITU-T Recommendation H.263, "*Video Coding for Low Bitrate Communication*," 1998
9. T. D. Neame, M. Zukerman and R. G. Addie, "*Application of the M/Pareto Process to Modeling BroadBand Traffic Streams*", Proc. of ICON'99, pp53-58, Brisbane, Queensland, Australia, 28 September, 1999.
10. Y. Wang and S Lin, "Error resilient video coding using multiple description motion compensation," *IEEE Trans. Circuits and Systems for Video Technology, to appear*
11. A. Riebman, H. Jafarkani, Y. Wang , M. Orchard and R. Puri, "Multiple description coding for video using motion compensated prediction", *Proc. ICIP 99, Oct. 1999*
12. University of British Columbia, http://spmg.ece.ubc.edu , H.263+ Public-Domain Code

Appendix A   Simulation System

The simulation system is shown in Fig A-1.

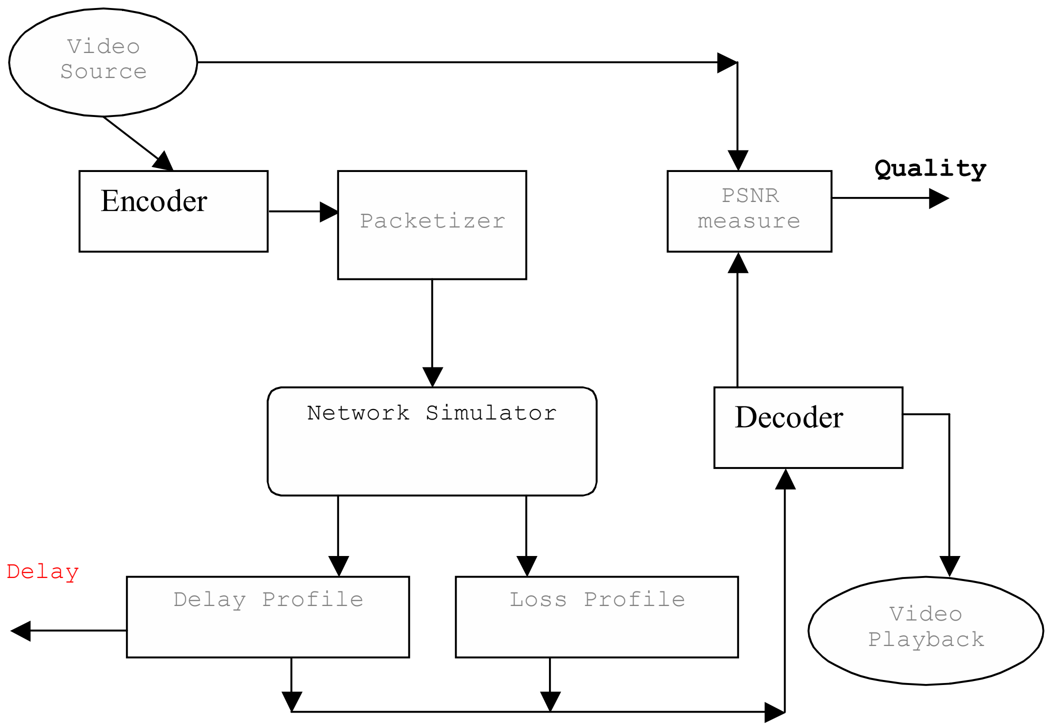


Figure A-1

Appendix B Source Codec and Packetizer

1. H.263([8])

H.263 is ITU-T video coding standard targeted high coding efficiency for low bit rate scenarios. For PSTN, H.263 support bit rate around 10-24 kbps. H.263 has provided a significant advance in coding efficiency over previous technology, and was used to form the efficient backbone of the MPEG-4 design. It was the first standard to demonstrate acceptable video quality over PSTN networks using 28.8 kbits/sec modems and brought video conferencing to the mass consumer market as well as displacing H.261 as the primary video conferencing standard for all bit rates. The H.263+ project, completed in early 1998, enhanced the performance of H.263 and extended the standard to a broader range of applications, including enabling the first highly effective error resilient standard video coding for mobile corrupting links and for Internet packet video. The H.263+ standard incorporates numerous enhancements and new features over the original H.263 design and is even better suited to low data rate and Internet applications.

Based on H.261, H.263 and H.263+ include many new features. H.263 and H.263+ support SNR, spatial and temporal scalability. They allow half-pixel motion vector with motion vector prediction, unrestricted motion estimation, syntax-based arithmetic coding and PB frame mode. Compared to H.261, it can provide 3-4 dB improvement for bit rate under 22 kbps for all ITU test sequences.

2. Encoder/Decoder for SC, MDC and LC
For single-layer coding, we use basic H.263 CBR encoder([12]). For LC, H.263+ SNR scalability encoder is adopted. Base layer and enhancement layer are separated and transmitted separately. For MDC, even frames and odd frames are considered as two descriptions, and they are compressed separately.

In the decoder, the decoder decodes packets which arrive in certain time. Lost packets and packets with large delay are discarded. Error concealment is used to make up the lost and delayed packets. In our simulation, copying from the last available frame is used as error concealment. For example, if GOB 1 to GOB 33 in frame j are lost, the content of GOB1 to GOB 33 in frame j-1 are copied to frame j. For LC decoder, error concealment for base layer is used, but no error concealment for enhancement layer.

3. Packetizer
In order to simulate with ns2, we need a packetizer to packet the bit stream. We use fix GOB number packetizer, which means we put fix number of GOB in each packet no matter how big are these GOBs. This kind of packetizer is better than fix bits packetizer because it is more relevant to source codec. When one packet is lost, it won't have effect across boundary of GOBs. This feature is convenient for error concealment. For our simulation, we packet each frame into 3 packets if they are I,P,B frames. Each frame in enhancement layer is packetized into one packet since enhancement layer in our simulation has far less bit rate than the corresponding base layer.

Timestamp and packet size are included in the trace files by the packetizer for the ns2.

Figure B-1 and FigureB-2 show the components in the video condec of MD (FigureB-1) and LC(FigureB-2).

4. PSNR: the Quality Measurement
Peak SNR for each frame is defined based on the luminance (Y) component:

$$PSNR(dB) = 10\log_{10}\frac{255^2}{E\{(X-\hat{X})^2\}}$$

The PSNR values can also be calculated against chrominance (U, V).

However, PSNR has been used to assess the quality of a video signal transmitted over a digital link. They calculate the PSNR for each frame and finally the average PSNR across the entire sequence. Recently, PSNR has been even used to assess the quality of video sequence transmitted over packet networks and it has been found to correlate surprisingly well with the subjective tests.

The encoder/decoder can calculate the quality measure of the encoded/decoded video as PSNR for Y, U and V for each frame and at end, it also give the average PSNRs across the whole video sequence.

In our simulation results, we calculate PSNR of Y ,U,V component separately. In this study, PSNR of Y component is used as most of the researches.

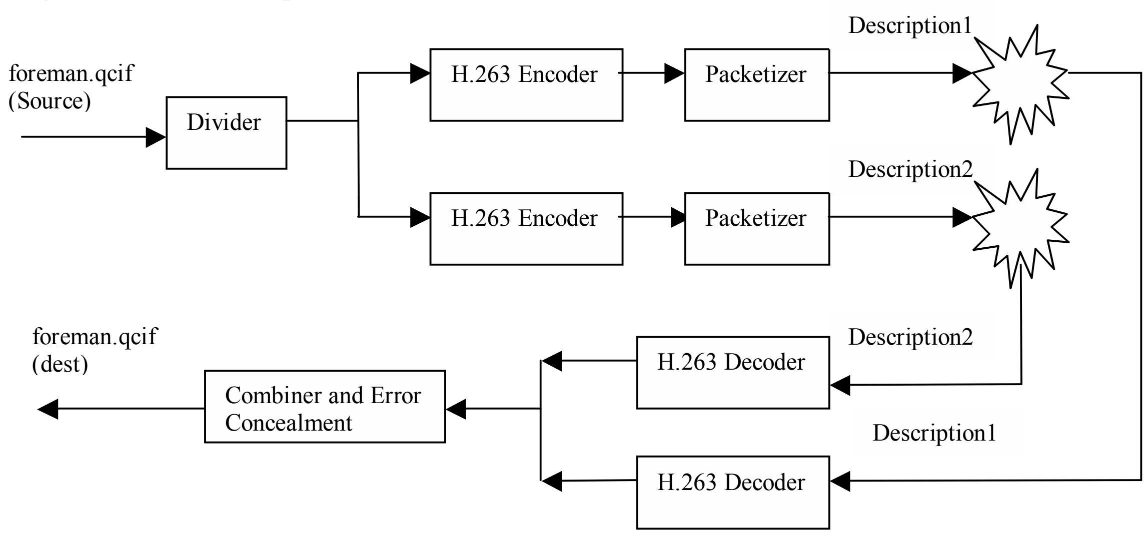
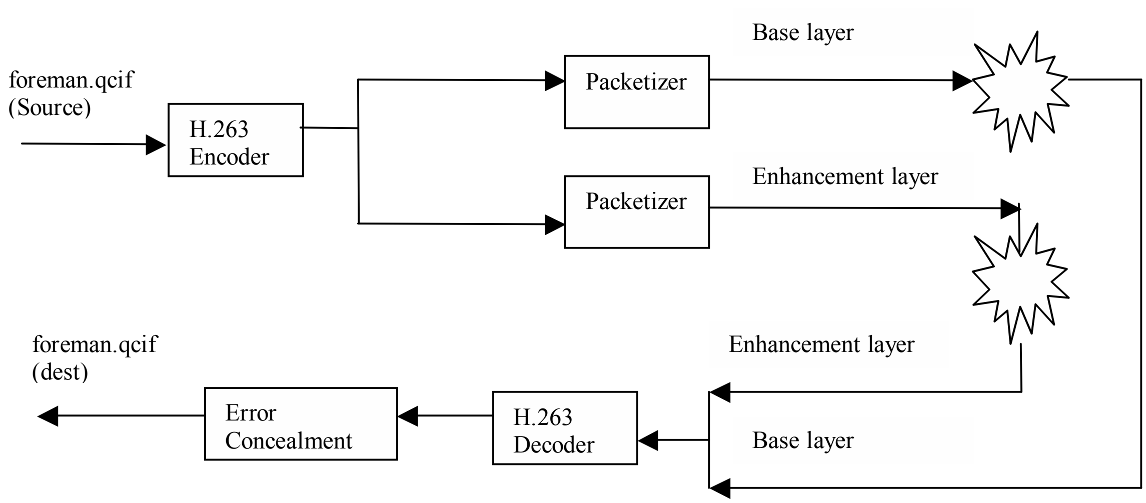


Figure B-1 MD codec



FigureB-2 LC codec

Appendix C Network Simulation

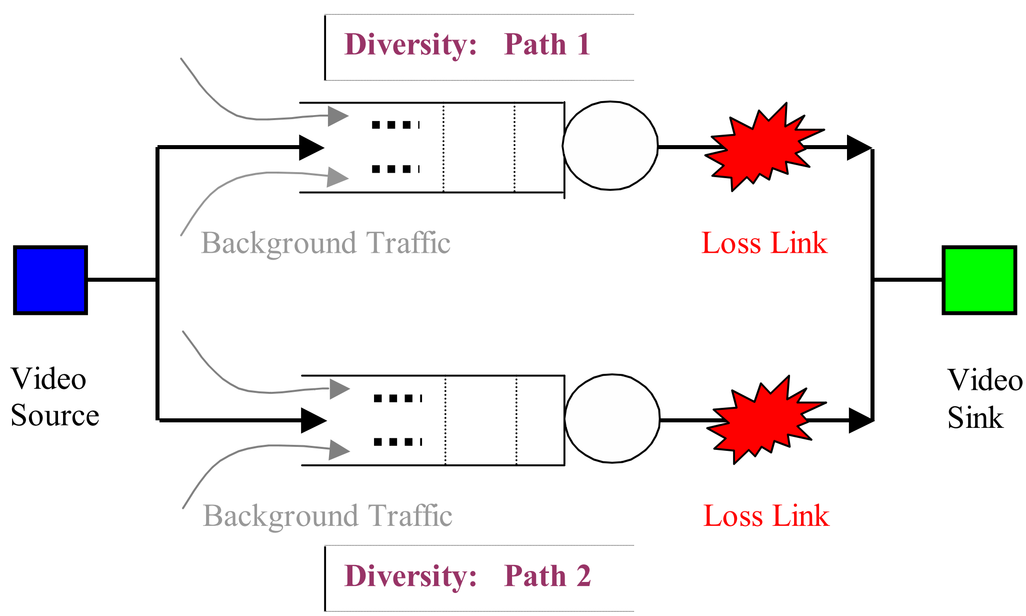
We choose NS2 (network simulator 2) to do the network simulations. That is because it provides fairly good environments and it is open source and free. NS2 was originally developed at the University of California at Berkeley in 1989 and currently in USC/ISI.
We chose ns-allinone-2.26 as our testing version, which can be downloaded from http://www.isi.edu/nsnam/ns/.

1. Simulation Topology

The simulation assumes a video source has some video to transmit to a sink user, and there are multiple, say 2 paths between the server and the user. There may be background traffic and link loss along each path, as shown in the figure. We simulate and calculate the performance under different link loss rate and background traffic situations.

Though the link capacity is very high in the current networks generally, there still exists some low bandwidth links, which will be the bottleneck of the network. Also considering the available bandwidth for each user is not so high, and in order to reduce the simulation time, we choose the link capacity 1Mbps. Together with suitable background traffics, these can simulate the behavior of the video streams over the real networks.

The encoded video streams (for LC the base layer stream and the enhancement layer stream, for MDC the different descriptions) are transmitted along each path separately, the user will combine them and decode to playback. We will do each experiment 10 times to compute the average PSNR for different transmission strategies under different link situations.



FigureC-1. Simulation Topology

In NS2, we can set up the simulation environment through the following steps.

```
#create new network simulator object
set ns [new Simulator]

#create server and user node
set server [$ns node]
set user [$ns node]

#create link between server and user
$ns duplex-link $server $user $link_rate $mean_link_delay DropTail

# set up namtrace
set nf [open $outnam w]
$ns namtrace-all $nf

#set up queue trace
set fo [open $outtrace w]
$ns trace-queue $server $user $fo

$ns at 50 "finish"

proc finish {} {
        global ns fo nf
        $ns flush-trace
        close $fo
        close $nf

        exit 0
}

$ns run
```

2. Video Traffic

Video traffic are obtained from the binary trace file, which is generated by the Packetizer and the format is as follows.

```
. . . .
<inter-arrival time in microsecond> <packet size in bytes>
<inter-arrival time in microsecond> <packet size in bytes>
<inter-arrival time in microsecond> <packet size in bytes>
. . . .
```

Note the traffic trace file should be converted to binary format and the ns2 source code is modified to assure the traffic start from the first packet instead of random position.

```
#creat video UDP source
set videoudp [new Agent/UDP]
$videoudp set fid_ 0
$ns attach-agent $server $videoudp

#Attach video stream trace to UDP
set tfile [new Tracefile]
$tfile filename $intrace
set videotrace [new Application/Traffic/Trace]
$videotrace attach-tracefile $tfile
$videotrace attach-agent $videoudp

# Create NULL UDP sink
set null [new Agent/Null]
$ns attach-agent $user $null

#connect and start the traffic
$ns connect $videoudp $null
$ns at $intrace_start_time "$videotrace start"
$ns at [expr $intrace_start_time + $intrace_duration] "$videotrace stop"
```

3. Simulation Scenario

3.1 Link with different loss rate

In this simulation, a link loss model is inserted to the link between the server and user. There are no background traffic and the link loss rate is set to the desired values (1%, 3%, 5%, 10% and 15%) for comparison.

```
#create loss module
set loss_module [new ErrorModel]
$loss_module set rate_ $lossrate
$loss_module unit pkt

set rng [new RNG]
$rng seed 0
set random_err [new RandomVariable/Uniform]
$random_err use-rng $rng

$loss_module ranvar $random_err
$ns link-lossmodel $loss_module $server $user
```

3.2 Link with fixed loss rate but different background traffics

Here we will investigate the influence of background traffic on the performance. The link loss rate is set to a constant value (3%) and kept unchanged during the simulation, but background traffics are introduced into the system. In order to simulate the general heavy-tailed traffic in real networks, we choose M/Pareto model as in [9]. In such model, flows come according to Poisson process, and each flow has a length complying with Pareto distribution, which is,

$$P(X > x) = \begin{cases} \left(\dfrac{x}{\delta}\right)^{-\gamma} & x \geq \delta \\ 1 & 0 < x < \delta \end{cases}$$

where $\delta > 0$ is the scaling factor, $1 < \gamma < 2$ is the shape factor. Note at this case $E(X) = \dfrac{\delta\gamma}{\gamma - 1}$ and $E(X^2) = \infty$.

In our simulation, the background traffic consists of 5000 CBR flows. The flows arrive at the link according to a Poisson process, in which the arrival rate indicates the average load of the link. Each flow has a size which is Pareto distributed with mean (10 packets, 100 bytes per packet) and shape (1.2), and a constant data rate which will influence the queue length (thus packet delay) along the link.

For Path 1, the background traffic has average load 0.2 and data rate 0.2 which corresponds to a low load situation, while for Path 2 the average load is 0.8 and the data rate is 0.05, which corresponds to a high load situation. The background traffic will start from the beginning and the video traffic will start from 30s because it needs some time for the Pareto distribution to generate large size flows, which causes the desired heavy-tailed traffic.

```
# This will be flow size of background traffic, Pareto distribution
set bgnpkts1 [new RandomVariable/Pareto]
$bgnpkts1 set avg_ $mean_npkts
$bgnpkts1 set shape_ $pareto_shape

# This will be flow interval, exponential distribution (Poisson process)
set bgflow_interval1 [new RandomVariable/Exponential]
$bgflow_interval1 set avg_ $avg_bgflow_interval1

set start_time 0
# set up background connections
for {set j 0} {$j < $numBgFlows} {incr j} {
   #node 0-1
   set udp0($j) [new Agent/UDP]
```

```
    $ns attach-agent $server $udp0($j)

    set cbr0($j) [new Application/Traffic/CBR]
    $cbr0($j) set packetSize_ $pktSize
    $cbr0($j) set rate_ $bgRate1
    $cbr0($j) attach-agent $udp0($j)

    $udp0($j) set fid_ [expr $j + 1]

    $ns connect $udp0($j) $null

    # number of packets follow Pareto distribution
    # actually number of packets is floor of $npkts+1
    set flow_npkts [$bgnpkts1 value]

    set start_time [expr $start_time + [$bgflow_interval1 value]]
    if ($j==0) {
          set start_time 0
    }

    set end_time [expr $start_time + $flow_npkts * $bgInterval1]

    $ns at $start_time "$cbr0($j) start"
    $ns at $end_time "$cbr0($j) stop"

}
```

4. Simulation Output

4.1 NS2 output trace file

As we set trace queue file in above scripts, NS2 will generate out.tr file to record the events happened in the link, and out.nam to feed into nam to animate the network dynamics.

An example of the trace file is shown as follows.

```
. . . .
+ 49.994405 0 1 cbr 100 ------- 4842 0.4841 1.0 122 48959
- 49.994405 0 1 cbr 100 ------- 4842 0.4841 1.0 122 48959
+ 49.994581 0 1 cbr 100 ------- 3011 0.3010 1.0 1280 48960
- 49.995205 0 1 cbr 100 ------- 3011 0.3010 1.0 1280 48960
r 49.997205 0 1 cbr 100 ------- 4842 0.4841 1.0 119 48941
+ 49.997538 0 1 cbr 100 ------- 4981 0.4980 1.0 31 48961
```

- 49.997538 0 1 cbr 100 ------- 4981 0.4980 1.0 31 48961
r 49.998005 0 1 cbr 100 ------- 3011 0.3010 1.0 1277 48942
+ 49.998153 0 1 cbr 100 ------- 4139 0.4138 1.0 560 48962
- 49.998338 0 1 cbr 100 ------- 4139 0.4138 1.0 560 48962
d 49.998338 0 1 cbr 100 ------- 4139 0.4138 1.0 560 48962
. . . .

First column is Event type, "r" means "receive", "d" means "drop", "+" means "enqueue", and "-" means "dequeue". The second column is the time each events occur. The sixth column is the packet size and the eighth column is flow ID. For the details of all the fields please refer to the NS2 manual from http://www.isi.edu/nsnam/ns/.

4.2 Statistical file of packet-dropped pattern

We have a program to analyze the out.tr file of each simulation. Generally the program will record the dropped packets and calculate the delay for received packets, the packet received with the delay exceeds some threshold value will also be considered as a dropped packet. Finally it generates a packet statistical file, which is in the format of follows.

. . . .
[Packet Number] [Packet Size] [Received time in microseconds] [flag of drop/received]
[Packet Number] [Packet Size] [Received time in microseconds] [flag of drop/received]
[Packet Number] [Packet Size] [Received time in microseconds] [flag of drop/received]
. . . .

Based on this file, we can calculate PSNR and feed the received packets to the decoder for playback.

Appendix D Full Results and Analysis

1.Coding detail

In this study, H.263+ package is basic codec for both MDC and LC. Video sequence forman.qcif with frame size 176*144 and frame rate 30HZ is used for the experiment. All simulations are based on 100kbps and 500kbps video stream. First 20 frames in foreman.qcif are simulated and PSNR is calculated as average of 20 frames. Each result is obtained as the average of 10 runs. In this study, we use PSNR of Y component as most studies. Quantization parameters are tuned to control the expected bit rate of bit stream. Table D-2 shows parameters used in our simulations and the achieved bit rates.

		MD		LC	
		Odd frames	Even frames	Base layer	Enhance layer
100k	Q	31	31	18	14
	Bit rate(kbps)	52.99	52.43	55.39	47.99
500k	Q	7	7	5	4
	Bit rate(kbps)	236.94	239.45	312.63	212.62

Table D-1

2. Comparative results for MDC and LC (Link with different loss rates).
In this simulation, we control the loss rate of links with ns2 loss model. No background traffic is added and no delay is concerned. For LC, and ULC, we always use link with lower loss rate to transmit base layer. So the loss rate can be read as (loss rate for base layer, loss rate for enhanced layer) for LC and (loss rate of description1, loss rate of description2) for MDC. Results are shown in Table D-2 and FigureD-3 and Figure D-4.

From TableD-2, we can see with similar loss rate, LC out performs MDC because MDC decrease the correlation of the frames. In LC, the loss rate of base layer will degrade the video quality much more than loss of enhanced layer. For LC, it is obvious that asymmetrical link with base layer in link with loss rate is better than symmetrical. For example, in LC100k, PSNR is 28.96 for (1%, 5%) and 28.86 for (3%,3%).

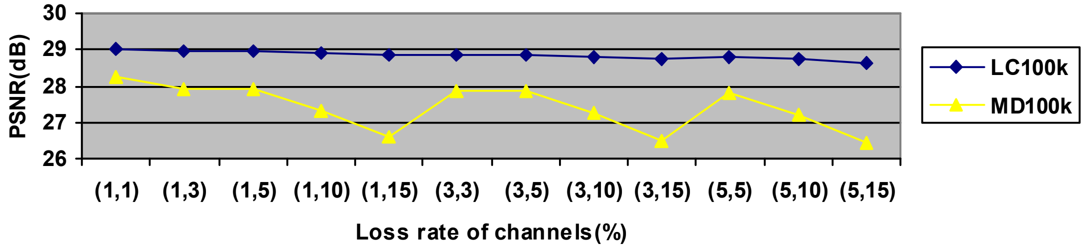
For MDC, it works better in symmetrical links than asymmetrical links. For example, in MD100k, PSNR of (3%, 3%) is better than (1%, 10%) and (1%, 15%).

From FigureD-3 and FigureD-4 we can conclude that MD performs better in symmetrical links and loss rate of base layer degrade the video quality more than loss in enhanced layer. But we can not show the advantage of MD compared with LC in our simulation. Using a better designed MD may change the results ([9][10]).

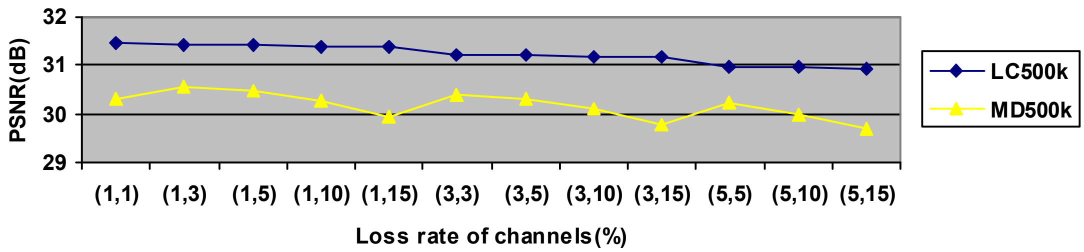
	LC 100k	MD 100k	LC 500k	MD 500k
(1%,1%)	28.99	28.24	31.45	30.71
(1%, 3%)	28.96	27.94	31.44	30.56
(1%,5%)	28.96	27.90	31.42	30.50
(1%,10%)	28.92	27.31	31.40	30.26
(1%,15%)	28.86	26.58	31.40	29.96
(3%,3%)	28.86	27.89	31.22	30.39
(3%,5%)	28.86	27.85	31.21	30.33
(3%,10%)	28.82	27.27	31.19	30.09
(3%, 15%)	28.75	26.50	31.18	29.79
(5%, 5%)	28.79	27.82	30.99	30.24
(5%, 10%)	28.75	27.22	30.96	30.00
(5%, 15%)	28.68	26.46	30.94	29.70

Table D-2

FigureD-3: Performance of LC and MD(bit rate: 100kbps)



FigureD-4: Performance of LC and MD(bit rate: 500kbps)



3. Comparative results for different bit rates(Link with different loss rates)

This simulation compares the video quality of different bit rate under different loss rate. Results are shown in Table D-5 and FigureD-6 and FigureD-7. The results illustrates that high bit rate video streams out perform low bit rate streams when loss rates for both are low. However, with the increasing of loss rate, high bit rate video stream is much worse than low bit rate video stream in low loss rate. To achieve best performance, bit rate must be chosen according to the load and available capacity of the links.

	LC100k	LC500k	MD100k	MD500k
(1%,1%)	28.99	31.45	28.24	30.17
(5%,5%)	28.79	30.99	27.82	30.24
(5%,15%)	28.68	30.94	26.46	29.70
(10%,10%)	28.48	30.68	27.07	27.32
(10%,30%)	28.30	30.56	24.26	25.60
(10%, 50%)	28.08	30.40	23.14	22.80
(30%, 30%)	27.32	28.75	23.22	24.51
(30%, 50%)	27.09	28.56	21.71	22.02
(50%,50%)	25.87	27.00	21.08	19.24

Table D-5

Figure D-6: Comparason of 100kbps and 450kbps (LC)

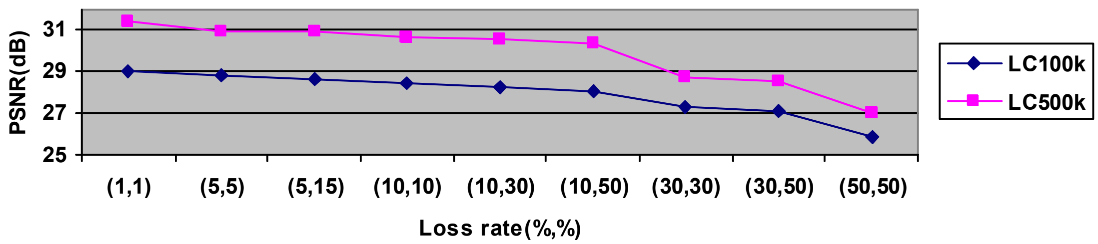
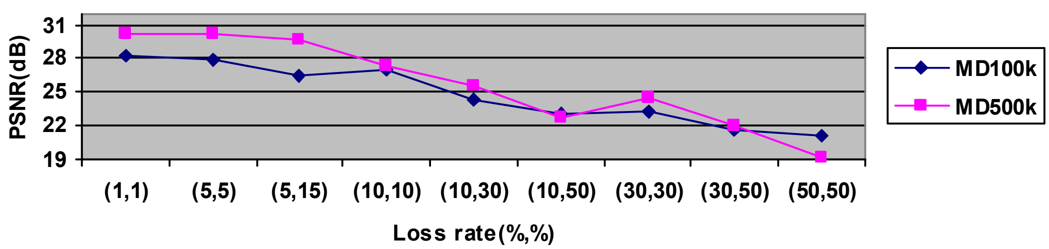


Figure D-7: Comparason of 100kbps and 450kbps (MD)



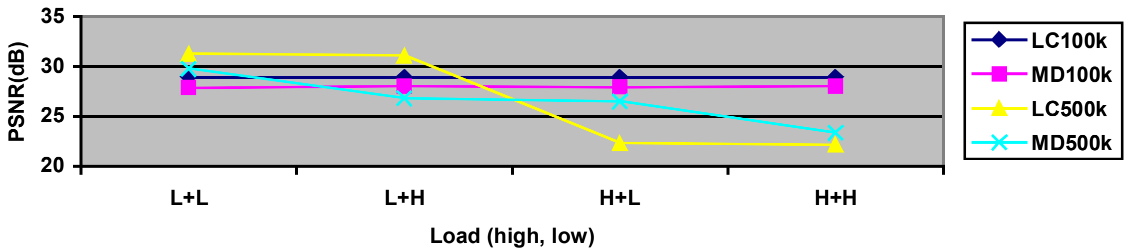
4.Complete results for MDC and LC (Link with fixed loss rate but different background traffics)

Besides controlling loss rate, we also simulate with real network scenarios. Please refer to Appendix C, part3.1 for simulation details. Results are shown in TableD-5 and FigureD-6. "L" means low load and "H" represents high load. As we can see in the table, when link load is low, loss rate for both 100kbps stream and 500kbps stream are both very low, thus bit rate with 500kbps is much better than 100kbps. When the load is very high, loss rate for 500kbps increased greatly(>50%), the quality of video sequence degrades to unacceptable zone. This simulation, different from controlling loss rate of the video traffic, simulates the real network condition and obtains the similar results as experiment of controlling loss rate.

		L+L	L+H	H+L	H+H
100k	LC	28.89	28.88	28.89	28.89
	ULC	28.76	28.76	28.77	28.76
	MD	27.82	27.98	27.90	28.00
500k	LC	31.26	31.07	22.29	22.15
	ULC	31.28	31.28	22.10	22.10
	MD	29.72	26.78	26.46	23.34

Table D-7

FigureD-8:LC and MD in real network simulation



3.4 Adapt bit rate to link condition

Based on the above simulations and results, we show the result of adapting bit rate according to link condition. FigureD-9 shows that if sever store two streams with bit rate 500kbps and bit rate 100kbps and chose video stream according to link feedback, it can obtain as much as 2dB improvement compared to single 100kbps stream. FigureD-10 presents that storing two streams with link adapting choose also outperform only storing 500kbps stream, especially in very crowded links.

FigureD-9: Comparason of 100kbps and 450kbps (LC)

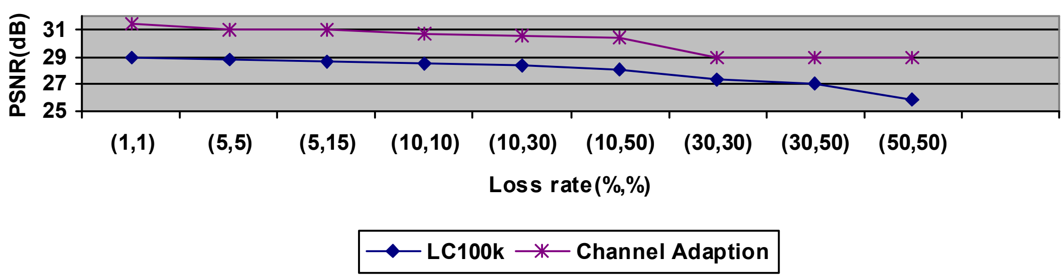
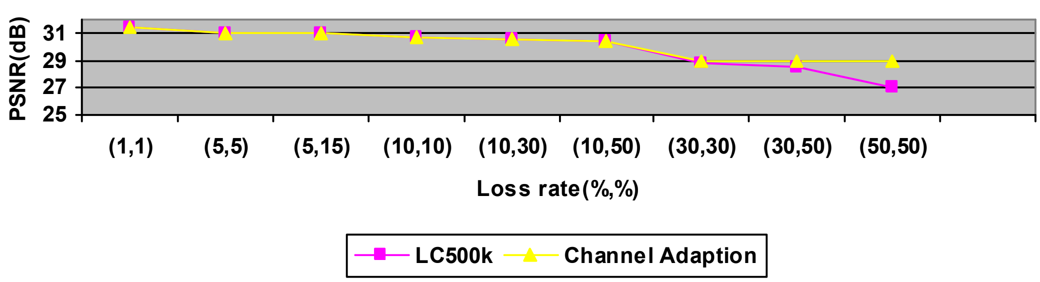


Figure D-10: Comparason of 100kbps and 450kbps (LC)



Appendix E Conclusion and Future Work

In this study, we compare different coding strategies (MD and LC) and different bit rate. For MD, simulation results shows symmetrical links are much more suitable for MD. While for LC, loss in base layer affects the quality of video much more than loss in enhancement layer. However, we can not show the advantage of MD over LC in our simulation. This is relevant to the characteristics of MD. MD can resist loss in a single link, while it decrease correlation among frames thus reduces the coding efficiency. Adopting a more efficient MD or allowing a little higher bit rate for MDC compared to LC may change the results.

We also compare performance of LC and MD with different bit rate. As demonstrated in the simulation results, in order to achieve optimal performance, different video streams with diversified bit rate should be chosen according the link capacity, link load and delay. All these factors change the loss rate of received video sequences. So the video server should be able to choose different bit rate streams adapting to loss rate feedback. Both simulation with controlled loss rate and simulation with background traffic and delay constraints support our conclusion.

Based on the comparison, when multiple links are available, the video server can choose the suitable stream to deliver, which will improve the performance of the system. Assume for one specific video file, the server stores several encoded versions with Layered Coding and Multiple Description Coding with different data rate level L_k. Simply the server can decide to transmit which stream to achieve the best performance according to the link quality, which indicates the current load and the link loss rate, and the link difference. In the real networks, it is easy for the server to know the average link error rate ε and the average queue length *l,* thus the link quality can be calculated as $A_i = f(\varepsilon_i, l_i)(i = 1,2)$. On the other hand, the link difference can be obtained simply by $D = |A_1 - A_2|$. The algorithm is described as follows.

if { D > some threshold value} then
    choose Layered coding files
    if { A_i *lies in* L_k *} then*
        send LC stream file corresponding to data rate L_k
    end
else
    choose Multiple Description files
    if { A_i *lies in* L_k *} then*
        send MDC stream file corresponding to data rate L_k
    end
end

The future work includes two parts. First is to do a more through comparison of more video format and bit rate. For example, besides low bit rate H.263 video, the video server should also support video in high quality in MPEG-2, MPEG-4 format and bit rate around 20 Mbps. The second part is to find the formula to calculate the link quality and to find the optimum data rate levels, which will be a tradeoff between the performance and the storage space.